Service Composition Formal Techniques in Service Oriented Architecture: A Survey

Amjad Hudaib[1], **Katrina Sundus**[2], **Sandi N. Fakhouri**[3]

[1, 2, 3]King Abdullah II School for Information Technology, University of Jordan, Amman, Jordan

Abstract: ***Service-Oriented Architecture (SOA) is a special kind of software design where services are provided to the other components by application components, via a network communication protocols. Service-oriented architecture can be implemented with Web services. SOA simplify a loosely coupled structure in a heterogeneous structure manner that enables the contribution of different business enterprises to develop the web service composition approaches together. The web service composition is an operation of combining several different web services together to accomplish certain business process that could be from different enterprises. In this paper, we presented different service composition existing formal techniques in service oriented architecture to classify them according to automated based web service composition techniques, algebra based web service composition techniques and other techniques with the verification and validation of their behavior in a survey manner.***

Keywords: Service Oriented Architecture, Service Composition, SOA, Formal Service Composition Techniques, Process Algebra Service Composition and Automated Service Composition

1. Introduction

Service-oriented computing is an emerging and promising distributed computational model. It is a contribution to the advanced type of software system, Service based system which is composed of many existing services and delivered as a value-added service [1]. Many enterprises such as Google, Amazon and Netflix, show a growing interest in using service composition to accomplish their deliver business applications. With the increase of web services having similar functionality, Quality of Service (QoS) and other non functional requirements become an important indicator of selecting the best services when immediate the service composition [2]. The development of the internet technology has led to creation and the application of SOA; Software systems which based on the application functions in the form of program services which access the internet or intranet environment not depending on the location in the global computer network. The principles of existing SOA do not take into consideration the functions performed by the services from the features of the main problem model or task to be solved, the tasks are optimal selection of the services from a variety of available and their dynamic adaptation to the specific needs, which can lead to adoption of the effective solutions for the organizing information processes [3]. Moreover, SOA services are composed of grouping services together, each service represent the basic part of service composition provided by service providers with functional features or non functional features Known as Quality of Services (QoS) utilized by clarifying the identical functionality of the service and its mission, in each service part there is a collection of determined services form the service composition and the service quality. The features of service composition can be changed by modifying services. The best service collection must be selected by Software engineering team to implement the system non-functional and functional requirements [4]. SOA automation optimaizational problem is a challenging field which could be solved by finding some efficient methods by software engineers to solve the desired requests, like using the search-based software engineering (SBSE); a method to perform the optimization techniques of the meta-heuristic search-based. Also, using genetic algorithm on heuristic search to perform and solve the optimization problem [5]. Service composition represent transferring user functional and non functional requirement to execution plan of service composition as Service Level Agreement (SLA) to detect the services that could achieve those requirements. Service Level Agreement can be a structure scenario or an optimal execution plan. General companies or individual systems direction is to get the help from external sources that could perform the tasks to build specific requirements for automated service composition. a lot of service composition approaches have been presented; the utilization of AI planning techniques was firstly introduced that leads to a two-staged composition. After that, different approaches were presented like workflow-based approach and template-based approach. Lately, the service composition approaches in the obvious semantic match emergence service composition development and implementing of Quality of Service (QoS) requirements [6].

Composition of services can be either manual or automatic service composition as shown in Figure 1. In the manual service composition the developer based on the incremental number of the web services to accomplish a specific application which causes a higher delay and a greater cost.

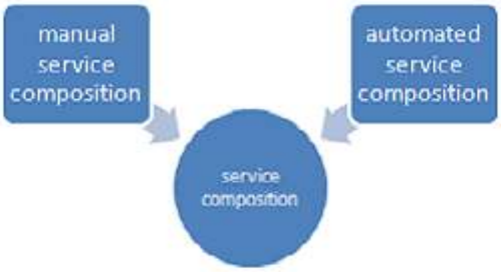

Figure 1: Service Composition Types [6].

It is not easy to detect the suitable web services according to the complexity of services. As a result, the decision has been taken to use the computing system; which is more beneficial approach with the privileges of lower costs than using manual service composition which could decrease the risk. Automatic composition of web services process the user who makes the request to which service that have the objective characterization or determine a specific entries output, the process is continue until the workflow of its procedures have been created. A lot of probably solutions may be preceded but the best solution will be taken to achieve the functional and non functional requirements that desired [6]. This paper is structured as follows: Section 2 introduces the related terminology. Section 3contains the existing related surveys. Section 4 illustrated the classification of the formal techniques in the service composition. Section 5 introduced the comparison in deferent tables. Finally, we presented the conclusion of our survey.

2. Related Terminology

In this section we will illustrate the Service Oriented Architecture (SOA), web service composition, dynamic service composition and finally the service composition framework.

2.1 Service-Oriented Architecture (SOA)

Service-oriented architecture (SOA) is an approach of the software development that based on the uses of the services with standard interfaces, the principles of reusing IT functional elements, eliminating duplication of functionality in software, unifying typical operating processes, ensuring the translation of the company's operating model to centralized processes and a functional organization based on an industrial integration platform [7]. Components of the program can be distributed to different nodes in the network communication model; each node is independent, loosely related construction in the complex distributed software systems. SOA has been proven itself for building large enterprise software applications. A number of developers and integrators that offer tools and solutions based on like the WebSphere platform, BEA Aqua logic, Windows Communication Foundation, SAP Net Weaver, JVP Jupiter [8]. Shrivastava et al. [9] they focuses their work on recognizing the faulty services and their replacement depending on QoS constraints instead of modifying the whole composition, reconfiguration of the faulty region which will reduce the overheads of computations and service distractions in service delivery. The problem solved by using two approaches. First approach is generating an optional path of services from its predecessor services to the completion of service process. The second approach is establishing the alternative service path of the same service process from begging until the end. Casati et al. [10] proposed a service composition based on the QoS consideration that supported the management specification and performance of E-services, sampled as processes and operated by a service process engine. Each node has its own rules for service selection, means some QoS parameters are correlated with each node. Zhou et al. [11] proposed a QoS aware service composition approach which expands the option limits for service selection by allowing services of various granularities to be available for the selection. This approach is based on mixed integer linear programming concepts and optimized the user defined objectives to meet QoS constraints. They based on optimizing the selection approach by maximizing and minimizing the linear objective function; constrained in nature. Zhang et al. [12] proposed an approach for identifying the faulty service within a service composition. They also proposed the approach of diagnosing and recovering the fault through reconfiguration using Dependency Matrix. This approach is a polynomial time algorithm that based on the concept of Predicate on Probe (PoP). This dependency matrix is built on the basis of process workflow structures and without any historical knowledge of prior executions. The traditional Quality of Service (QoS) analysis model cannot be directly applied on the big data service composition which could lead to other problems according to the huge amount of heterogeneous data and applications that evolved in this structure. Dong Li et al. [13] proposed an expanded edition of QoS-based analysis model that uses the linear regression model with a set weight of QoS using AHP analysis that improved the service selection algorithm based on backtracking method and validate its effectiveness.

2.2 Web Services Composition

A Web service is a network technology that provides a program communication between different applications based on Web standards. Web-services allow you to access from one application to another and simultaneously perform certain functions. The application solution could be provided by web services or from the consumer web services published by other providers. The Web service has a programming interface, represented in the WSDL (Web Services Description Language) format as an example; a language for describing and accessing Web services, based on the XML language, designed for unified presentation of external web interfaces [6]. . FEKIH et al. proposed a new approach based on the local consistency build up methods to enhance the skyline approach and to reach the user requirements. The Harmony Search algorithm adopted to catch a near-optimal composition. The experimental evaluation is reduced the size of classic Skyline in a reasonable time computation [14]. Also, Elli Rapti et al. adopted the artificial potential fields that can be mapped to the service composition problem in Internet of Thing networks to achieve global equilibrium through local interactions [15]. Yuqian Lu and Xun Xu examined the knowledge-based service composition and adaptive resource planning in a dynamic cloud environment by developing a systematic and easy to be implement service composition mechanism .They used system utilizes distributed knowledge for intelligent service composition and adaptive resource planning [16]. Kamath et al. recommend a framework that employs a base up methodology to constructing semantic Web services starting with existing service descriptions on the Web, Subsequently decreasing those duration of the time Furthermore cosset included in substantial scale manual annotation [17]. In the Dynamic Service Composition mechanism where the application services in the runtime are connected with many data services, the state of being interconnected is deployed dynamically based on the characteristics of database management; for effective data accessing many cloud databases runtime is being provision by connectivity of the data service dynamic and application service [18]. Wang et al. [18] proposed a novel pattern built element service framework that merge outlined pattern

What's more design implementation methodologies to prepare composited services. The pattern may be formalized in such uncommon chart similitude matching issues. Furthermore, the algorithm may be outlined that could figure those similitude services creation results. Zhang et al. [19] recommend an algorithm MR-IDPSO (MapReduce according to progressed discrete particle swarm Optimization), which makes utilization of those moved forward discrete particle swarm optimization (PSO) for those Map Reduce should solved extensive scale dynamic service composition. Analyses indicate that our algorithm outperforms the parallel genetic algorithm and the result of effective vast scale dynamic service composition. Kamath et al. [20] recommend a framework that employs the base up methodology to construct the semantic Web services, dynamically starting with existing service descriptions on the web, decreasing the time duration. Furthermore, the cost that included in substantial scale manual annotation. Parsia et al. [21] depicted actualized framework which performs translation, employ a developed SHOP2 usage on arrange for What's more over the translated domain; cetera executes those coming about compositions. Furthermore building that framework, present approach, the organizer constantly executes yield handling activities concerning illustration its compositions. Wang et al. [22] proposed a novel technique in the view of SHOP2, which domain is independent a state based ahead HTN organizer. The technique employments a weighted vector will mirror the choice makers execution for Different targets. Also, anytime requester algorithm will enhance the capacity by Figure that best result want to from the composition. Resource and service are two ideal ideas. The suppliers deal with their services internally and uncover services to clients. The service, once bought eventually the customer perusing the client. Furthermore, it guaranteed toward providers; by providing an agreement between these gatherings. Those suppliers would be able to answer for mapping those services with configurations (a supplier might delay to setup until the client begins to utilize them on attain resources) [23]. Ponnekanti et al. [24] Portrayed SWORD, a toolset that permits the developers with a fast creating existing web benefits to figure it out as a new composite web services. SWORD gives a particular data-point in the self evident expressiveness complexity efficiency tradeoff that exists in the web service composition.

2.3 Service Composition Framework

Services that could be offered online are defined as a loosely coupled and reusable software component which encapsulates discrete functionality [25]. According to the maximization in the potential SOA solution by identifying the business closely connected services framework; which is being designed based on the work requirements and the service contract that the services solution must accomplish. The Service composition framework contains the following components: translator, evaluator, process generator, and execution engine and service repository. And have two main participants' service requester and service provider, service requester firsthand participants who suggest the service in order to be used through the framework which provide external specification to the translator [26]. The service requesters expand service provider's services and data. The translator is a connecter between service provider and process generator to make the language understandable for process generator; by translating external specification language that used by service requester to the language that used by process generator. The execution engine executes the process and plan of data from the evaluator and sends the output to the service requester. For every request translator sends internal specification to process generator .For each request, the process generator produce a plan which control the existing service repository services in order to accomplish the request. If there are many plans the evaluator evaluates all plans and give the leading service for processing [27]. Ghazal et al. [28] generated a Services Composition Model (SCM) that provides a general solution for the services composition problem by inquiring the requirements of the new service using the requirements of the already existing service. Also, produced a Services Composition Language (SCL) as a simple text-based language which allows the user to define the requirements of his request, the inserted request will then be analyzed using Parsing Algorithm to obtain the name of the requested services, after that the Service Composition Algorithm will be executed all the steps of the composition process and return the result of the composition to the user. The web service composition Workflow techniques classified into orchestration and choreography composition types. In orchestration model, the engaged web services are under control of a single central process (another web service). In choreography model, it does not depend on a central point, all web services should participate. Moreover, in the choreography has to know exactly when to become active and with whom to interoperate [29].

3. Existing Service Composition Serveys

Syu et al. [28] reviewed an automatic web service composition surveys. Their work focuses on already published surveys in AWSC, they their paper according to service combination-centric, service selection-centric, combination and selection hybrid, service discovery-centric and workflow description-centric. Kalamegam [30] summarized different existing approaches that used Colored Petri Nets model to design verification of the web service composition. Glaucia et al. [31] presented a structured survey that shows many service composition approaches models, class of formalism and the purpose of their formalization. Beek et al. [32] compared between different approaches according to a set of characteristics (such as, trust, security and performance) which is a variety of concrete proposals from the formal methods community that have been emerged in order to verify the correctness of the web service composition; based on state action models or process models. Bartalos et al. [33] discussed the related issues to the semantics of services, which is important for automatic Web service composition. Also, suggested the problem formalization contributing to the formal definition of the pre-/post-conditions, with possibility to have value restrictions, and their relation to the semantics of services. Moreover, they provided an overview of several existing approaches dealing with the problem of Web service composition and discussed the achievements flow in the field and depict some open research areas. Rao et al. [34] gave an over view of the progress in automatic Web services composition. First, they proposed a model for Web services composition process consisting from five-steps; the service composition model consists of the service presentation, translation, process generation, evaluation and execution. Each step of these needs different languages, platforms and methods. Then, they summarized some research efforts in the automatic Web service composition both from the workflow and AI planning research community sides.

4. Service Composition Formal Techniques

There are many different languages have been proposed to allocate service compositions (like, WS-BPEL [35], WS-CDL [36]), the design and execution of using these languages are still lake of formal semantics, which not support the correct validation and sometimes lead to error-prone. This could be fixed by applying Formal Description Techniques (FDTs) to specify service compositions. Sometimes, possible to use these Formal Description Techniques with a combination of the languages mentioned earlier. FDTs are particularly useful in the design of safety-critical systems and whenever failures can have severe consequences that not desired in such systems. The main benefit of applying FDTs is the possibility of verifying, validating and refining services compositions. In this section, we will divide the service composition techniques into Formal Techniques, Semi-Formal Techniques, each of them will be explained briefly and illustrated the related techniques. There are a lot of languages have been proposed to allocate service compositions the design and execution. By using these languages there are still lake of formal semantics and capabilities, which not support the correct validation and sometimes lead to error-prone. The Unified Modeling Language UML is an interesting and reasonably compelling basis for embedded system design methodology. UML has a shortage of several key capabilities. The embedded systems usually not implemented from the scratch, they are compositions of subsystems. Both functional and architectural descriptions of the system could be based on the composition of the subsystems. With this design strategy, the conformation must be equally on the validity of the composition as on the correctness of the constituent parts [37]. Where, the formal methods provide a strict mathematical means to ensure the large software systems to be achieved the desired needs. All formal methods discussed to allow automatic composition, hierarchical service compositions and scalability. Most these methods ensure the validation and correctness of the service composition [32]. We classified the Formal Description Techniques into three main classes of formalism illustrated in Automata, Process Algebra and Petri Nets.

4.1 Automata Based Service Composition Techniques

Automata are commonly used model for the formal specification of the systems. They are consisting of a set of states, actions, transitions between states and an initial state. Each transaction can be denoted by a label from one state to another. Moreover, Automata have been applied as the basics for defining other formal description techniques that gives information about the system execution behavior [38]. Many service composition techniques used automata techniques. In other words, automated service composition algorithms are one form of service integration. Given a set of service descriptions, an initial and goal states, the composition algorithm tries to orchestrate a subset of the services in order to achieve the goal state. During orchestration, constraints imposed by services that are part of this composition, such as HATEOAS, must be respected. Verborgh et al. [39] proposed a composition and execution algorithm specifically for Representational State Transfer (REST) (services supporting the HATEOAS constraint. In addition, recent SOAP (Simple Object Access Protocol) composition algorithms have achieved high performance marks. On the other hand, Publish/Subscribe services are treated in complete isolation with techniques such as CEP (Complex Event Processing). The service composition algorithms are quit offering structure from claiming service integrative provided for a situated from claiming services descriptions. Furthermore, the algorithm creation tries to coordinate a subset of the services in place will keep in the objective state. Jingjing et al. [40] proposed the Timed Automata for Web Service Composition TAC model and implemented the web service composition automation engine that could receive the web service interface description language (WSIL) and automatically generating TAC model to perform a composite service. Also, a new model to verify composition web service using Enhanced Stacked Automata Model (ESAM) has been proposed in [29]. The composited algorithm based on automata. The correctness properties have been evaluated based on safetyness, liveness, dead transition, deadlock, and reachability. The web services are composed by using Business Process Execution Language for Web Service (BPEL4WS) and then converted into ESAM (combination of Muller Automata (MA) and Push Down Automata (PDA) and it is turned into Promela language, an input language to the Simple ProMeLa Interpreter (SPIN) tool. The model is proved using SPIN tool and the experiments shows better results in term of finding dead transition and deadlock. Moreover, shehu PhD thesis [41] in automata area of service composition which contributes the prediction problem by presenting a novel in learning automata-based non-negative matrix factorization algorithm (LANMF) to estimate end-to-end network latency of the composition service in the cloud environment. LANMF encodes every web service node as an automaton which permits it to estimate its coordinator in such way that prediction of the error is minimized. Atampore et al. [42] proposed a formal framework, Supervisor Aware Service Composition Architecture (SASCA) that models the Web service composition based on the Supervisory Control Theory (SCT) of Discrete-Event Systems (DES). The novelty in their approach is the application of control theory to service-oriented computing and the incorporation of run-time input into the supervisor generation process. Bhat et al. [43] proposed another semantic language for organizer of network system services and constructing created benefits individually on the automated manner. Those semantic language may be extensible permitting to a totally about organize services that will be displayed. The novel algorithm exhibited in the organizer will fill in with situated documentation that should discover different made services which could be a chance to be developed with domains past system benefits. Zhu et al. [44] characterize the idea of the multi functional and create a model of comprehensive service composition. Since A large portion existing composition thinking strategies might just handle a single functionality, they augment them also to create new algorithm to automated comprehensive service composition. Aggarwal et al. [45] present demand Web Service Composition device in METEOR-S, which permits those transform designers on Web benefits with a conceptual process, In view of business and procedure imperatives What's more produce an executable procedure. Approach may be to decrease a great part of the Service Composition issue will a demand fulfillment issue. It makes utilization a multi-phase methodology to demand dissection. This worth of effort might have been done similarly as and only the METEOR-S framework, which plans on backing the complete lifecycle from claiming semantic Web forms. Kalamegam et al. [7] Recommend Web service composition as a large portion of full grown with regard to it out quickly evolving prerequisites from claiming benefits of the business to service-oriented results. Web service composition coordination what is more on web service choreography would those types for compositions. The formal routines help to weigh whether the service composition matches those fancied determination alternately that a principle property of the composition is disregarded. Asmethal et al. [46] recommended Web service framework is used to give services in an ideal way. In the suggested structural engineering five levels are utilized. Benefits need aid made independently with those help of a toolkit thus those seniors could select their service effectively. Web service Adapters need aid utilized for conveying for a few Web services. It may be necessary for build the composite benefits manually. This will be particularly vital on account of composite service formation may be not so much a one-time exertion. Kamath et al. [47] proposed adaption of different service composition language. The devices recognized for the assessment exhibited here are WSDL2OWL-S [48] Mind swap OWL-S API [49] and OWL-S [50] editorial manager. Web services need aid presently the practically common method for accomplishing service introduction to endeavor frameworks again those Web. Ontology’s similar to OWL-S and WSMO bring been characterized Likewise and only those semantic Web activity. Siddiqui et al. [51] proposed a two phase approach called template based composition to find a compromise between complexity and flexibility. They intermediate approach called template based composition (TFC) which utilizes the advantages of aforementioned selection and composition techniques to reduce the complexity and provide the flexibility. Moreover, they occupied the methodology in two phases in the time deployment during the design phase and the run time phase. Also, they considered the non functional requirements such as the security and the QoS. Song et al. [52] proposed a composition system AFLOW based on OWL-S; a combination of AI planning and workflow technology. They discussed the implementation of the AI planning subsystem in the AFLOW by supporting a higher dynamic automation of composition. Also, they described how the AI planning system (SHOP2) can be combined with OWL-S Web service type descriptions. Bhandari et al. [53] Suggested An intermediary based structure to secure the web service composition Eventually perusing presenting An broker’s layer between benefits of the business methodology Also web benefits. The secure creations about web benefits will a chance to be performed utilizing transposition procedure dependent upon star topology and Intalio business transform architect. Those recommended framework will be illustrated with the instance of distinctive running in distinctive puts done secure way through the web. Those suggested method will assistance to decrease the UDDI load. Ordonez et al. [54] Proposed an automated planning based framework for automated convergent composition based in Hierarchical Task Networks (HTN); a mechanism to include the user context information into the HTN planning; and a mechanism to tag the convergent services using HTN descriptors. They consider in the designing HAUTO framework to be easy to use, can be integrating with a high performance platform used in telecommunications and the reaction to exceptions and fault tolerant; planning systems adopted in dynamic environments that must be able to react to potential failures during execution of the services. Personalization: users may express their preferences in any request. Typical cases include minimizing the time of the execution and cost. Gabrel et al. [55] proposes QoS-aware syntactic service composition issue which need have been the liable from claiming various investigations. Those well-known benchmark WSC, permits experimenting methodologies to such issue utilizing manufactured benefits. In this article, we review that that hypothetical unpredictability of the tests from claiming needs aid different: the service creation issue displayed to WSC-08 (without QoS) will be NP-hard, same time for those issue displayed On WSC-09 (QoS-aware service composition), and they recommend a polynomial-time algorithm. Paik et al. [56] Suggested adjustable transaction What's more QoS-aware service Choice approach under five client custom settings In light of hereditary calculation (GA) on deliver over worries. QoS-awareness encouraged Eventually Tom's perusing multi-objective QoS criteria. Also, utilizes the multi variant streamlining. We led to careful evaluation, Also it reveals to suggested technique viably Also effectively scope the worldwide ideal of the generally determination criteria. Raik et al. [57] presented ASTRO a comprehensive framework that could be used to define and support highly adaptive context-aware service-based business processes and applied to a real-world scenario from the logistics domain. In their approach, they made an adaptation to the related tasks, from detecting the adaptation problem to until finding solutions and applying them to the running process instance which, performed automatically at run time with minimal execution cost of the processes. KASSMI et al. [58] proposed development of the security requirements over the existing framework called Discovery and Visual Interactive Web Service Engine (DIVISE). DIVISE main objective is generating the best match between the Web Services Composition and the relative BPEL code according to user’s queries that handles the functional requirements. Moreover, handling of web services’ Non Functional Requirements during the composition, and the verification of generated composite service according to the user’s specified properties done by utilized model checking tool UPPAAL. Parsia et al. [21] designed a DAML-S Process Model to support the application of AI planning techniques to the automated composition of Web services. SHOP2 is a Hierarchical Task Network (HTN) planner appropriate for working with the Process Model. They proved the consistency between the semantics of SHOP2 and the situation calculus semantics of the Process Model. Also, they implemented a system that sounds to have completely plans over sets of DAML-S descriptions using a SHOP2 planner, after that, it executes the resulting plans over the Web. Sell et al. [59] presented an instrument that takes under record WSMO description that will be a user-guided, intelligent media composition approach whereby Web services need aid ran across Also recommended of the clients as stated by the composition connection. The produced composition is arranged in IRS-III by our java API to dataflow coordination. Pathak et al. [60] proposed research introduces a new incremental approach to service composition, MoSCoE (Modeling Web Service Composition and Execution), based on the three steps of abstraction, composition and refinement. Abstraction refers to the high-level description of the service desired (goal) by the user, which drives the identification of an appropriate composition strategy. In the event that such a composition is not realizable, MoSCoE guides the user through successive refinements of the specification towards a realizable goal service that meets the user requirements.

4.2 Process algebra base service composition

A Process Algebra can be defined as a mathematical framework for the high-level characterization of interactions, communications and synchronizations among a collection of transactions that concurrently executed. Also, process algebra can be used to detect unwanted properties and derived desirable properties of a system specification. Furthermore, it could provide establishment whether two processes have equivalent behaviors or have similarities. Some well-known process algebras are Calculus of Communicating Systems (CCS), Calculus of Sequential Processes (CSP) and Language of Temporal Ordered Systems (LOTOS) [61]. Process algebras can be applied in model concurrent and distributed interactive systems. Compensating Customer Service Provider (cCSP) is a language defined to the business transactions model within the framework of CSP process algebra. It has the mean of modeling faults within a transaction as compensations. However, CPS language lacks the automated tool that supports the verification of the service composition. Wherever, the Finite state Process (FSP) supports the verification of the services composition. The cCSP is a combination of both benefits of each CSP and FSP, in the service composition [62]. Mateescu et al. presented a technique for generating adaptors by given the description of reused entities' conversations and an abstract specification of the way mismatch can be settled. The approach is based on an encoding into the LOTOS process algebra; by taking the advantages of the CADP toolbox for LOTOS to verify the correctness of the contract [63]. Fujii et al. [64] Displays a semantics-based context-aware element service composition schema that composes the provision through joining together conveyed parts in a light semantics segments; more contexts for clients. The suggested schema comprised for the part service model with semantics (CoSMoS), part Runtime surroundings (CoRE). Furthermore, the semantic chart based service composition (SeGSeC). The recommended framework gets client starting with user-specified standards by means of Taking in. Additionally, schema should adjust the progressive situations.

Petri nets are a graphical and mathematical modeling tool used in many systems. It is very helpful tool to specifying information processing systems [65]. Petri Nets used for describing systems that are characterized as being concurrent, asynchronous, distributed, parallel and nondeterministic. It belongs to a special kind of directed and bipartite graph consisting of a set of places, transitions, directed arcs and an initial marking. The places may contain a discrete number of marks which represent the state of the modeled system. Petri Nets can be used to analyze system behavior according to the properties of the system such as reachability, liveness and boundness. Colored Petri Net and Hierarchical Petri Net are examples of Petri Nets expansion [66]. In the proposed a meaningful object-oriented Petri net (G-NET) based on algebra that managed complex composition in Web services. The efficient technique for Web services composition used the standard internet protocols such as Universal Description Discovery and Integration (UDDI), Web Service Description Language (WSDL) and Simple Object Access Protocol (SOAP) in order to make the service available. Moreover, the specification and prototyping of complex Web services are allowed to model G-Nets. The formal semantics that used in the composition operators are defined, by means of G-Nets. This provides a rigorous approach to verify properties and detect inconsistencies between Web services [67]. Nie et al. [68] proposed a model based on the colored Petri net (CPN) that provides a semantic support to the complex web service composition combined with a closed composing algebra, that is defined to attain a framework which enables declarative composition of web services. Fan et al. [69] proposed a new service composition technique based on the template mechanism in the scalable and flexible network framework and implemented a system framework for the development of network services that can be deployed and used in the service composition in an efficient manner to ensure the non functional requirements such as the security and QoS aspects.

4.3 Other Techniques

Meer et al. [70] they proposed a FUSION service model which provided a formal approach of web service composition and automatic execution. It is uses the abstracting of an optimal execution plan from the requirements and specifications of the user. The verification is based on the foundation of the user satisfaction criteria, and provides appropriate recovery procedures. Also, they presented a Web Services Execution Specification Language (WSESL), a language that developed to describe the execution plans in the context of the FUSION services model. Alves et al. [71] introduced a semantic web service composition (SWSC) approach that deals with web service execution flaws. SWSC enables the automatic adjustment of the composition at run-time. SWSC based on a contingency strategy which encompasses many alternative web service compositions that allows the workflow execution engine to readily choose a composition to execute, keep track of execution progress, and switch; if possible; to a suitable alternative web service composition in case of a web service operation execution flaw. They used the BPEL4WS (Business Process Execution Language for Web Service) Service Translator module translates the contingency strategy tree to a collection of superimposed processes as an orchestration workflow. Peng et al. [72] introduced an adaptive approach, named estimation of distribution algorithm based on Restricted Boltzmann Machine (rEDA) to iteratively explore the solution space of service composition until the optimal composite service is established on the fly. In other words, it maintains in affective way a variety of alternative solutions, by leveraging the inference ability of Restricted Boltzmann Machine to hold the potential solutions. rEDA constructs the probabilistic distribution of composite services in dynamic nature and improves the current exploration strategy with the consideration of the degree of how well a service contributes to the global QoS. Zhao et al. [73] they adopted services composition in wireless sensor networks by reducing the multi-objective and multi-constrained optimization problem, that could be solved by adopting particle swarm optimization (PSO) algorithm and genetic algorithm (GA) by finding an approximate optimal Wireless Sensor Networks (WSNs) services compositions. Yan et al. [74] Recommend a novel Graph-Based Memetic algorithm (GBMA) to solve the QoS-aware WSC issues. GBMA adopts the chart representational suggested by GraphEvol, which will be a standout amongst the state-of-the-symbolization algorithms. Wang et al. [75] designed and implemented a self-adaptive and context-aware service composition system that can be adapted to the execution contexts changing and made the adjustments according to the context events and user-defined personalized policies. They used OWL to model context ontologies and extend the OWL-S service model to support context information. Service consumers can submit their requests and get context aware services. Their system can composite services according to the service consumers' requests, execute services, monitor execution contexts and adjust its action when the contexts changed. Kona et al. [76] Introduced framework utilizing USDL (Universal Service-Semantics depiction Language), a language for formally describing those semantics for web-services. USDL is dependent upon the Web metaphysics language (OWL) and employs Word Net Similarly as a basic foundation for Comprehension the significance about benefits.

5. Comparison

Table 1 summarizes different web service composition techniques that we included in our survey. This table will illustrate the compression between the different approaches in different perspectives such as follows: the researcher name, composition approach, composition time (the service composition is established during the design time, the running time or through the dynamic binding), composition strategy; automatic, semi automatic or dynamic; the composition correctness ;if the service composition approach achieves the correctness, not achieved or partial correct, and functional F Non Functional NF; functional, non-functional or both requirements that achieved from the service composition approach.

Table 2 gives another view of the comparison between the illustrated approaches in terms of the composition approach name, the researcher, the strength points (advantages of the approach), the weaknesses (the disadvantages of the approach), sources or tools that used in the approach and finally the composition type.

6. Conclusion

Service composition techniques and approaches and the public access number on the different services are growing up. So, we can see the rapid improvement on the service composition technologies to accommodate the rapid growing. In this survey, we presented the existing service composition formal techniques in service oriented architecture to be classified according to automated based web service composition techniques, algebra based web service composition techniques, and others techniques with the verification and validation of their behavior in a survey manner. The comparison between the listed approaches is done in different sight views which illustrated in two tables. Our future work is to add more techniques to our comparison and making the comparison more specific such as comparing different approaches that satisfy in one non-functional requirement.

Table 1: Comparison between Service Composition Approaches.

Researcher	*Composition approaches*	*Composition time/service binding*	*Composition strategy*	*Composition correctness*	*Requirement*
Aggarwal et al. [45]	Meteror-s	Run-time Design-time	Dynamic automatic Semi-automatic	No	Partial (F)
Ponnekanti et al. [24]	SWORD	Dynamic binding Run-time	Semi-automatic	No	F,NF
Siddiqui et al. [51]	Template based composition	Dynamic binding	Automatic	Yes	F,NF
Kamath et al. [20]	Bottom-up approach E-Flow	Run-time Dynamic binding	Dynamic dynamic	No No	Partial F
Parsia et al. [21]	Shop2	dynamic	semantic	No	F
Bhandari et al. [53]	Star WSCop	Dynamic binding	Dynamic Automatic	No	F Partial NF
KASSMI et al. [58]	DIVISE	Run-time Dynamic binding	Automatic	Yes	F, NF
Ordonez et al. [54]	HAUTO	Dynamic binding	Automatic	Yes	F
Pathak et al. [60]	MOSCOE	Design –time Dynamic binding	Semi- automatic	Partial	NF
Fujii et al. [64]	SECSE Cosmos & fujii	Design –time Run-time	Semi- automatic Dynamic	No No	NF F,NF
Gabrel et al. [55]	QoS	Run-time	Automatic	Yes	F,NF

		Design time			
Raik et al. [57]	astro	Run-time	Automatic	No	NF
Jingjing et al. [40]	TAC	Run-time	Automatic	Yes	F, NF
Nagamouttou et al. [29]	ESAM	Run-time	Automatic	Yes	F, NF
Shehu [41]	LANMF	Run-time	Dynamic	Yes	F, NF
Ripon et al. [62]	cCSP	Run-time	Automatic	Yes	F
Chemaa et al. [67]	G-Nets	Dynamic binding	Automatic	Yes	F
VanderMeer et al. [70]	FUSION	Run-time	Dynamic	Yes	F
Atampore et al. [42]	SASCA	Run-time	Automatic	Yes	F
Jhonatan Alves et al. [71]	SWSC	Run-time	Automatic	Yes	F
Shunshun Peng et al. [72]	rEDA	Run-time	Dynamic	Yes	F

Table 2: The Advantages and the Disadvantages of the Web Service Composition Techniques

Composition approaches	*Researcher*	*Strength*	*Weaknesses*	*Sources*	*Composition Type*
AFLOW	Song et al.[52]	Run time composition. Functional validation.	The service should be in the registry to execute. Non-functional requirements not considered.	OWL-S [50]	Orchestration
SHOP2	Parsia et al. [21]	Automatic. Composite web services.	Non-functional requirements not considered.	DAML-S [77]	Orchestration
DIVISE	KASSMI et al. [58]	Security (privacy, integrity and authentication) Run time verification. QoS Correctness Run-time.	Security adaptation overhead.	BPEL [78] UPPAAL [79]	----
HAUTO	Ordonez et al.[54]	Dynamic environment. Minimal execution time and cost. Correctness. Easy to use.	Non-functional requirements not considered.	SAWSDL [80]	----
MOSCOE	Pathak et al.[60]	High performance. Cost consideration.	QoS. Security and other non-functional requirements.	OWL-S [50] WSDL [81] BPEL4WS [82]	Orchestration
Self adaptive and context aware system	Wang et al.[75]	Automatic QoS (availability, cost, time, reliability, satisfaction) Run time. Correctness.	Security	OWL-S [50]	----
QoS	Gabrel et al. [40] Paik et al.[55]	QoS based service composition in terms of cost, reliability, execution time, and throughput. Polynomial time algorithm.	Security	SOAP [83] REST [84]	----
Meteror-s	Aggarwal et al. [45]	Automatic. Ensure correctness and optimality. QoS(time, cost, availability and reliability)	Security	UDDL [85] WSDL [81] BPEL4WS [82] BPWS4J [86]	Orchestration
ASTRO	Raik et al. [57]	Complex service based systems. Real-world applications (synchronization). Dynamic execution environment. Parallel execution of services. Minimal execution cost of the processes.	Security. QoS.	BPEL [78]	Orchestration
SWORD	Asmethal et al. [46] Ponnekanti et al. [24]	Improve the government-citizen communication. Support both automatic and semi-automatic web service composition. Scalability.	QoS Security	UDDL [87] SWORD [24]	----
TAC	Jingjing et al. [40]	Real time applications (synchronization). Minimal cost.	Non-functional requirements not considered.	UPPAAL [88]	----
ESAM	Nagamouttou et al. [29]	Suitable for both deterministic and non-deterministic systems. Avoid the dead transactions and deadlock (safety). Correctness. Reachability. Liveness (process termination).	Security	BPEL4WS [89] SPIN [90]	Orchestration
LANMF	Shehu [41]	Ensures the QoS in dynamic environment.	Security	XML [91]	Orchestration

		QoS (response time, availability, reliability and cost). Low latency. Large environment (cloud).		SOAP [92] WSDL [93] UDDL [87]	
cCSP	Ripon et al. [62]	Supports both sequential and parallel composition. Composition verification in parallel to check. Ensures the system correctness synchronization.	Non-functional requirements not considered.	cCSP [94] LTSA [95]	Orchestration and Choreography.
G-Nets	Chemaa et al. [67]	Dynamic behavior of services. Parallel composite services. Increase the reliability. Service availability. Respond in optimal time. Object oriented approach (reusable software, extensible components, encapsulation, etc).	Security	UDDL [96] WSDL [93] SOAP [83]	Orchestration
FUSION	Meer et al.[70]	Develop that describes execution plans. Formal description of the notion of the correctness and optimality.	Non-functional requirements not considered.	WSESL [97]	Choreography
SASCA	Atampore et al. [42]	The correct correctness resulted by construction control synthesis. Have a controller that guarantees the system requirements. Specify services by providing a well defined syntax and semantics for system model.	Non-functional requirements not considered.	WS-BPEL [98]	Orchestration
SWSC	Jhonatan et al. [71]	Run- time Automatic Calculate the min-cost time Support non-deterministic planning. Adaptation on the fly.	Security QoS.	BPEL4WS [99]	Orchestration
rEDA	Peng et al. [72]	Distributed systems. Maintain alternative solutions. Adaptation on the fly. Dynamic nature.	Security	----	----

References

[1] J. Yan, H. Jin, and Y. Yang Q. He, "Quality-aware service selection for service-based systems based on iterative multi attribute combinatorial auction", IEEE Trans. Software Eng., vol. 40, no. 2, pp. 192–215, 2014.

[2] J. Yang, and W. Zhao H. Zheng, "Probabilistic qos aggregations for service composition", TWEB, vol. 10, no. 2, pp. 12:1–12:36, 2016.

[3] Sang Il Kim and Hwa Sung Kim, "Ontology-based Open API Composition Method for Automatic Mash-up Service Generation", 2016.

[4] Yousef Hassouneh, Abdel Salam Sayyad and Nariman Ammar Wafaa Radwan, "a GA-based Optimizer for Optimizing Security and Cost in Service Compositions" 2017.

[5] Wenjian Wang and Hang Xu Lifang Ren, "A Reinforcement Learning Method for Constraint-Satisfied Services Composition", 2017, DOI 10.1109/TSC.2017.2727050.

[6] Alexis Huf, Ivan Salvadori and Frank Siqueira, "Planning and Execution of Heterogeneous Service Compositions", IEEE Symposium on Computers and Communications (ISCC), 2017.

[7] Poonkavithai Kalamegam, "Usage of CPN Models in Web Service Compositions", International Conference on Technical Advancements in Computers and Communications, 2017.

[8] Chitra Babu, R.C. Brinda, Alexis Vallidevi, and Krishnamurthy, "Performance Evaluation of Dynamic Composition & Dynamic Reconfiguration in SOA Applications", International Conference on Computer, Communication, and Signal Processing (ICCCSP-2017), 2017.

[9] Ashish Sharma, Deepika Shrivastava and Saurabh Shrivastava, "An Approach for QoS Based Fault Reconfiguration in Service Oriented Architecture", Procedia Computer Science 46, pp. 766 – 773, 2015.

[10] Ilnicki S, Jin L, Krishnamoorthy V, Shan M and Casati F, "Adaptive and dynamic service composition in eflow", Technical Report, 2000.

[11] Keting Yin, Honghong Jiang, Shuai Zhang and Bo Zhou, "QoS-based Selection of Multi-Granularity Web Services for the Composition", Journal of Software, vol. 6, no. 3, pp. 366 – 373, MARCH 2011.

[12] Xiaoqi Zhang, Yi-Chin Chang, Kwei-Jay Lin and Jing Zhang, "The Implementation of A Dependency Matrix-based QoS Diagnosis Support in SOA Middleware", vol. 12, no. 7-9, July-September 2012.

[13] Jiechu Wu, Zehang Deng, Zelin Chen and Yang Xu Dong Li, "QoS-Based Service Selection Method for Big Data Service Composition", IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), 2017.

[14] Sabri MTIBAA, Sadok BOUAMAMA and Hela FEKIH, "Local-Consistency Web Services Composition Approach Based On Harmony Search", pp. 1102-1111, September 2017.

[15] Anthony Karageorgos, Vassilis C. Gerogiannis and Elli Rapti, "Decentralised Service Composition using Potential Fields in Internet of Things Applications", ScienceDirect, ELSEVIER, pp. 700 – 706, 2015.

[16] Yuqian Lu and Xun X, "A semantic web-based framework for service composition in a cloud

manufacturing environment", Journal of Manufacturing Systems 42, pp. 69–81, 2017.

[17] Sowmya Kamath S and Ananthanarayana V.S., "A Bottom-up Approach Towards Achieving Semantic Web Services", International Conference on Advances in Computing, 2013.

[18] Zhitao Wan and Ping Wang, "A Novel Pattern Based Dynamic Service Composition Framework", IEEE International Conference on Services Computing, 2016.

[19] Zihui Jing, Yiwen Zhang and Yanping Zhang, "MR-IDPSO: A Novel Algorithm for Large-Scale Dynamic Service Composition", vol. 20, no. 6, December 2015.

[20] Sowmya Kamath S and Ananthan arayana V.S., "A Bottom -up Approach Towards Achieving Semantic Web Services", International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013.

[21] Bijan Parsia, Evren Sirin, James Hendler and Dana Nau Dan Wu, "Automating DAML-S Web Services Composition Using SHOP2", International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013.

[22] Hongwei Wang, Chao Qi, Dan Liu, Jian Wang and Zhe Wang Minglei Li, "Handling Multi-objective in an HTN Planner for Emergency Decision-making", Conference Anthology, 2013.

[23] Shu Huang and Ilya Baldin, "Dynamic Market-based Path Service Composition for Software Defined Networks", 2016.

[24] Shankar R. Ponnekanti and Armando Fox, "SWORD: A Developer Toolkit for Web Service Composition", 2002.

[25] Ian Sommerville, "Software Engineering", Tenth Edition ed. United States of America: RR Donnelley, Pearson Education Limited. 2016.

[26] Hongming Cai and Wei Wang Junjie Mao, "A Web of Things Based Device-adaptive Service Composition Framework", International Conference on e-Business Engineering, 2016.

[27] Chengyang wang, Eunsam Kim, Sumi Helal and Choonhwa Lee, "Blueprint Flow: A Declarative Service Composition Framework for Cloud Applications", vol. 5, 2017.

[28] Amjad. Hudaib, Waffa Maitah and Ghadeer Ghazal, "SOFTWARE ARCHITECTURAL PATTERNS FOR SERVICE COMPOSITION", International Journal of Computer Science & Information Technology (IJCSIT), vol. 6, August 2014, No 4.

[29] Ilavarasan Egambaram, Muthumanickam Krishnan, Poonkuzhali Narasingam and Danapaquiame Nagamouttou, "A verification strategy for web services composition using enhanced stacked automata model", 2015.

[30] Poonkavithai Kalamegam, "Usage of CPN Models in Web Service Compositions-A Survey", International Conference on Technical Advancements in Computers and Communications, 2017.

[31] Nelson S. Rosa, Luís Ferreira Pires Glaucia M. and M. Campos, "A Survey of Formalization Approaches to Service Composition", International Conference on Services Computing, 2014.

[32] M. Bucchiarone, A. Gnesi and S. ter Beek, "A survey on service composition approaches: From industrial standards to formal methods", Technical Report 2006-TR-15, 2006.

[33] Peter Bartalos and Maria Bielikova, "Automatic Dynamic Web service composition:a survey and problem formalization", Journal on Computing and Informatics, vol. 30, 2011.

[34] Jinghai Rao and Xiaomeng Su, "A survey of automated web service composition methods", Proceedings of the First international conference on Semantic Web Services and Web Process Composition, pp. 43-54, 2004.

[35] T. Erl, "Service-Oriented Architecture (SOA): Concepts, Technology, and Design", Boston: Prentice Hall, p. 792, 2005.

[36] D. Burdett, G. Ritzinger, T. Fletcher, Y. Lafon, C. Barreto and N. Kavantzas, "Web Services Choreography Description Language Version 1.0.", W3C Candidate Recommendation, November 2005.

[37] Grant Martin, "UML for Embedded Systems Specification and Design: Motivation and Overview", Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition (DATE'02), 2002.

[38] Rajeev Motwani, Jeffrey D., Ullman John. and E. Hopcroft, "Introduction to Automata Theory, Languages, and Computation", 3rd ed.: Boston: Adison- Wesley Publishing Company.

[39] D. Arndt, S. Van Hoecke, J. De Roo, G. Mels,T. Steiner, J. Gabarro and R. Verborgh, "The pragmatic proof: Hypermedia API composition and execution", Theory and Practice of Logic Programming, vol. 17, no. 1, pp. 1–48, 2017.

[40] Zhu Wei, Zhao Xing, Zhu Dongfeng and Hu Jingjing, "Web Service Composition Automation based on Timed Automata", Appl. Math. Inf. Sci. 8, no. 4, pp. 2017-2024, 2014.

[41] Umar Galadima Shehu, "Novel Optimization Schemes for Service Composition in the Cloud using Learning Automata-Based Matrix Factorization", PhD thesis, October 2015.

[42] Juergen Dingel, Karen Rudie and Francis Atampore, "Automated Service Composition Via Supervisory Control Theory", Proceedings of the 13th International Workshop on Discrete Event Systems, 2016.

[43] Robinson Udechukwu, Rudra Dutta, George N., Rouskas and Shireesh Bhat, "On Service Composition Algorithm for Open Marketplaces of Network Services", 2017.

[44] Farokh Bastani, I-Ling Yen, Jicheng Fu, Yuqun Zhang and Wei Zhu, "Automated Holistic Service Composition:Modeling and Composition Reasoning Techniques", 24th International Conference on Web Services, 2017.

[45] Kunal Verma, John Miller, William Milnor and Rohit Aggarwal, "Constraint Driven Web Service Composition in METEOR-S", IEEE Conference Publications, 2015.

[46] R. Asmetha Jeyarani, "E-Government application using service oriented architecture with an integration of SWORD", International Conference on Emerging

Trends in Science, Engineering and Technology (INCOSET), 2012.

[47] Sowmya Kamath S. and Ananthanarayana V., "Towards Semantic Web Services: An empirical evaluation of service ontology generation tools", Annual India Conference (INDICON), 2015.

[48] M. Burstein, D. Mcdermott, S. Mcilraith, D. Martin and et al., "Bringing semantics to web services with owl-s", World Wide Web, vol. 10, no. 3, pp. 243–277, 2007.

[49] E. Sirin and B. Parsia, "The OWL-S Java API", Third International Semantic Web Conference (ISWC2004), 2004.

[50] M. Burstein, J. Hobbs, D. Martin and et al., "OWL-S: Semantic Markup for Web Services, Version 1.1", 2004, avilable at: http://www.daml.org/services/owl-s.

[51] Abbas Siddiqui, Rahamatullah Khondoker and Paul Muller, "Template Based Composition for Requirements Based Network Stacks", 2012.

[52] Zhaoteng Song, Xiangfeng Luo and Xinhuai Tang, "Planning Engine based on SHOP2 for AFLOW", Seventh International Conference on Semantics, Knowledge and Grids, 2011.

[53] Rajat Bhandari and Ugrasen Suman, "Broker Based Secure Web Service Composition Using Star Topology", International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2012.

[54] J. C. Corrales and P. Falcarin A. Ordonez, "Automated composition of convergent services based in HTN planning", Ingeniería e Investigación, vol. 34, no. 1, pp. 66 – 71, Apri 2014.

[55] Maude Manouvrier, Kamil Moreau, Cecile Murat and Virginie Gabrel, "QoS-aware Automatic Syntactic Service Composition problem: complexity and resolution", 2015.

[56] Incheon Paik, Banage T., Kumara T., Akila S. and Siriweera, "QoS and Customizable Transaction-aware Selection for Big Data Analytics on Automatic Service Composition", 14th International Conference on Services Computing, 2017.

[57] Antonio Bucchiarone, Nawaz Khurshid, Annapaola Marconi, Marco Pistore and Heorhi Raik, "ASTRO-CAptEvo: Dynamic Context-aware Adaptation for Service-based Systems", Eighth World Congress on Services, 2012.

[58] Ilyass EL KASSMI and Zahi JARIR, "Security Requirements in Web Service Composition: Formalization, Integration and Verification", 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2016.

[59] Farshad Hakimpour, John Domingue, Enrico Motta, Roberto C., S. Pacheco and Denilson Sell, "Interactive Composition of WSMO-based Semantic Web Services in IRS-III ", 2014.

[60] Samik Basu, Robyn Lutz, Vasant Honavar and Jyotishman Pathak, "MoSCoE: A Framework for Modeling Web Service Composition and Execution", vol. 30, pp. 793–827, 2011.

[61] W. Fokkink, "Introduction to Process Algebra", Amsterdam: Springer- Verlag New York, p. 166, 2000.

[62] F. Sultana, F. Rahman and S. Ripon, "Verification of Service Composition and Compensation by Using Process Algebra", Journal of Advances in Computer Networks, vol. 4, no. 4, December 2016.

[63] Pascal Poizat, Gwen Salaün and Radu Mateescu, "Adaptation of Service Protocols using Process Algebra and On-the-Fly Reduction Techniques", Institute of Electrical and Electronics Engineers, 2012.

[64] Keita Fujii and Tatsuya Suda, "Semantics-based context-aware dynamic service composition", Trans. Auton. Adapt. Syst., vol. 4, 2009.

[65] S. Ripon, Nakul C. Das, Orin Hossain and Mohammad Salah Uddin, "A Comparative Study of Web Service Composition via BPEL and Petri Nets", International Journal of Computer and Electrical Engineering, vol. 6, no. 2, April 2014.

[66] T. Murata, "Petri Nets: Properties, Analysis and Applications", Proceedings of the IEEE, vol. 77, no. nº 4, pp. 541-580, 1989.

[67] Faycal Bachtarzi, Allaoua Chaoui and Sofiane Chemaa, "A high-level Petri net based approach for modeling and composition of web services", International Conference on Computational Science, Procedia Computer Science 9, pp. 469 – 478, 2012.

[68] Houxiang Wang, Xiaopei Jing, Zhihao Xie and Kai Nie, "Web Service Automatic Composition Model Based on Colored Petri Nets", Communications and Network, vol. 5, pp. 101-105, 2013.

[69] Wen Fan and Jun Shen, "A service composition method based on the template mechanism in the Service Scalable Network Framework", 8th International Conference on Computer Science & Education, 2013.

[70] Datta A, Dutta K, Thomas H, Ramamritham K, Navathe and VanderMeer D, "FUSION: A System Allowing Dynamic Web service Composition and Automatic Execution", International Conference on E-Commerce, 2003.

[71] Jerusa Marchi, Renato Fileto, Mario A. R. and Dantas Jhonatan Alves, "Resilient Composition of Web Services through Nondeterministic Planning", Symposium on Computers and Communication (ISCC), 2016.

[72] Hongbing Wang and Qi Yu Shunshun Peng, "Estimation of Distribution with Restricted Boltzmann Machine for Adaptive Service Composition", 24th International Conference on Web Services, 2017.

[73] ZhangBing Zhou, Ke Ning, Yucong Duan, Liang-Jie and Zhang Deng Zhao, "An Energy-Aware Service Composition Mechanism in Service-Oriented Wireless Sensor Networks", International Congress on Internet of Things, 2017.

[74] Yi Mei, Hui Ma, Mengjie Zhang and Longfei Yan, "Evolutionary Web Service Composition: A Graph-based Memetic Algorithm", 2016.

[75] Boxue Wang and Xinhuai Tang, "Designing a Self-adaptive and Context-aware Service Composition System", 2014.

[76] Ajay Bansal, Luke Simon, Ajay Mallya, Gopal Gupta, Thomas D. Hite and Srividya Kona, "USDL: a service-semantics description language for automatic service discovery and composition", International Journal of Web Services Research, vol. 6, 2009.

[77] Burstein M., Hobbs J.R., Lassila O., Martin D., McDermott D., McIlraith S.A., Narayanan S., Paolucci M., Payne T., Sycara K. and Ankolekar A., "The DAML Services Coalition:DAML-S: Web Service Description for the Semantic Web", the First International Semantic Web Conference, 2002.

[78] "Web Services Business Process Execution Language Version 2.0", 2007.

[79] Alexandre David, Kim G. Larsen, John Hakansson, Paul Petterson, Wang Yi, Martijn Hendriks and Gerd Behrmann, "UPPAAL 4.0", the 3rd international conference on the Quantitative Evaluation of Systems, September 2006.

[80] IBMHolger Lausen, DERI Innsbruck and Joel Farrell, "Semantic Annotations for WSDL and XML Schema (SAWSDL) ", W3C Recommendation, 28 Augus 2007.

[81] P. Rajasekaran et al., "Enhancing Web Services Description and Discovery to Facilitate Orchestration", In conjunction with ICWS'2004, 2004.

[82] Andrews et al., "Business Process Execution Language for Web Services Version 1.1", 2003, available at http://www-106.ibm.com/developerworks/webservices /library/ws-bpel/.

[83] D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, D. Winer and D. Box, "Simple object access protocol (soap) 1.1", May 2000, [Online]. Available: http://www.w3.org/TR/2000/NOTE-SOAP-20000508.

[84] Markus Fischer, Dimka Karastoyanova, Frank Leymann, Karolina Vukojevic and Florian Haupt, "Service Composition for REST", 18th International Enterprise Distributed Object Computing Conference, 2014, 1541-7719/14.

[85] "UDDI, Universal Description, Discovery and Integration", 2002, http://www.uddi.org.

[86] Curbera et al., "IBM Business Process Execution Language for Web Services JavaTM Run Time, BPWS4J", http://www.alphaworks.ibm.com/tech/bpws4j.

[87] "UDDI Architecture", 2004, http://www.uddi.org/pubs/uddi-v3.0.2 20041019.htm#_Toc85907967.

[88] Gmez R., "Model-checking Timed Automata with Deadlines with Uppaal", Formal Aspects of Computing, vol. 25, pp. 289-318, 2013.

[89] Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, S. Weerawarana and F. Curbera, "Business Process Execution Language for Web Service (BPEL4WS) 1.0", 2002, Published on the World Wide Web by BEA Corp, IBM Corp and Microsoft Corp.

[90] Holzmann G, "The model checker SPIN", IEEE Trans Softw Eng 1, pp. 279–295, 1997.

[91] Newcomer and Eric, "Understanding Web Services: XML, Wsdl, Soap, and UDDI", In Addison-Wesley Professional, 2002.

[92] M. Hadley, N. Mendelsohn, J. Moreau, H. Nielson, A. Karmarkar, Y. Lafon and M. Gudgin, "Simple Object Access Protocol", 2007, http://www.w3.org/TR/soap12-part1/.

[93] F. Curbera, G. Meredith, S. Weerawarana and E. Christensen, "Web services description language (wsdl) 1.1", March 2001, [Online]. Available:http://www.w3.org/TR/wsdl.

[94] S. H. Ripon, "Process algebraic support for web service composition", SIGSOFT Softw. Eng. Notes, vol. vol.35, no. no. 2, 2, March 2010.

[95] J. Magee, "Behavioral analysis of software architectures using LTSA", in Proc. the 21st international conference on Software engineering(ICSE '99), pp. pp. 634-637, 1999.

[96] M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, S. Weerawarana and F. Curbera, "Unraveling the web services web an introduction to soap, wsdl, and uddi", IEEE INTERNET COMPUTING, pp. 86–93, 2002.

[97] F. Curbera, G. Meredith, S. Weerawarana and E. Christensen, "Web services description language (wsdl) 1.1", March 2001, [Online]. Available:http://www.w3.org/TR/wsdl.

[98] F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, S. Weerawarana and T. Andrews, "Business Process Execution Language for Web Services Version 1.1", May 2003, [Online; accessed 10- June-2014].

[99] Matjaz B. Juric, "Business Process Execution Language for Web Services BPELand BPEL4WS 2nd Edition", 2nd ed., 2006, ISBN:1904811817.

[100] Anshuk Dubey and Sanjukta Pal, "Dynamic Service Composition towards Database Virtualization for Efficient Data Management", 2017.

[101] Yong-Yi Fanjiang, Jong-Yih Kuo, Shang-Pin Ma and Yang Syu, "A Review of the Automatic Web Service Composition Surveys", International Conference on Semantic Computing, 2014.