WEB SERVICES COMPOSITION METHODS AND TECHNIQUES: A REVIEW

Narges Hesami Rostami and Esmaeil Kheirkhah and Mehrdad Jalali

[1]Department of Computer Engineering, MashhadBranch, Islamic Azad University,Mashhad, Iran

ABSTRACT

Web Services are modular, self-describing, self-contained and loosely coupled applications that can be published, located, and invoked across the web. With the increasing number of web services available on the web, the need for web services composition is becoming more and more important. Nowadays, for answering complex needs of users, the construction of new web services based on existing ones is required. This problem is known as web services composition. However, it is one of big challenge problems of recent years in a distributed and dynamic environment. The various approaches in field of web service compositions proposed by the researchers. In this paper we present a review of existing approaches for web service composition and compare them among each other with respect to some key requirements. We hope this paper helps researchers to focus on their efforts and to deliver lasting solutions in this field.

KEYWORDS

Semantic Web Service, Semantic Similarity, Web Services Composition, Composition Techniques.

1. INTRODUCTION

Service Oriented Architectures (SOA) and Web Services are presented in the mainstream scientific and industrial for many years. SOA is an architectural paradigm and interactions or patterns between them for components of a system. In this architecture, service is a contractually defined behavior that can be implemented and provided by a component for by using another component [1].

Web service is a [2]. software component that takes the input data and produces the output data. Web services are loosely coupling that allows developers to create, generate and compose them at runtime, interfaces that describe a collection of operations that are network-accessible through standardized web protocols and its features are described by using a standard eXtensible Markup Language (XML)-based language. However, Web Services are syntactically usually and described with standards such as Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Universal Description Discovery and Integration(UDDI)) [3]. The description of web service consists of the technical parameters, constraints and policies that define the terms to invoking service[1].. A web service is defined as a four-tuple WS= (name, des, In, Out), name represents service name and is used as a unique identifier; des represents the

DOI : 10.5121/ijcseit.2013.3603

description of service; In represents the input parameters set of service; and Out represents the output parameters set of service. The SOAP is a protocol to exchanging structured information in a decentralized and distributed environment. WSDL is a XML format for describing the web services and only describes the syntactic interface of web services that alone cannot be used for automatic composition of web services. Thus, semantic standard protocols such as WSDL-S (Web Service Description Language-Semantic), WSMO (Web Service Modeling Ontology), OWL-S (Ontology Web Language - Service) and SAWSDL (Semantic Annotations for Web Service Description Language) have been developed for the automatic web service composition and also UDDI is a virtual registry that exposes information about Web services[4].

In many cases, atomic web service isn’t sufficient to achieving complex needs of the user. Therefore, web services composition is appropriate solution to finding an optimal composition of web services to satisfy various user requests using their syntactic and/or semantic features[5][6][7].

The rest of this paper is organized as follows: Section II describes preliminary knowledge and background for following discussion. Section III we provide a review of some approaches and current methods and then we categorized all these approaches with respect to some service composition requirements. Finally, Section IV sums up the conclusion.

2. PRELIMINARY CONCEPTS

This section presents preliminary concepts and background of web service composition.

2.A: Ontology

Ontology is a shared conceptualization of the world. Ontologies provide a common understanding of a particular domain and also provide a set of well-founded constructs to building meaningful higher level knowledge for specifying the semantics of terminology systems. In a particular domain, ontology represents richer language for providing more complex constraints on types of resources and their properties and usually It expressed with logic-based language, so that meaningful distinctions can be made among the classes, properties and relations[8].

2.B: Semantic Web Architecture

The Semantic Web identifies a set of technologies, tools and standards that support the vision of the web associated with meaning form the basic building blocks of an infrastructure. It provides a process level description for the web service that in addition to functional information have models the preconditions and post conditions of the process that can be inferred the evolution of the domain logically. The Semantic Web architecture is composed of a series of standards organized into a certain structure that is an expression of their interrelationships[8][9].The Semantic Web enables machines for interpreting, combining and using from data on the Web. The basis the Semantic Web is computer-understandable descriptions of resources and is used to adding semantics to Web service technology (e.g. OWL-S, WSDL-S or WSMO etc.). To make sense of all data and services, semantic web is made on the foundations of logic and knowledge representation to help computers for finding the right information[10]and also introduces framework for semantic description of web services and related aspects[11].

Semantic web services are software components that provide dynamic service discovery, composition and invocation of web services and facilitate automated handling of web services for the users. Semantic web and web services are synergistic: the semantic web transforms web into a repository of computer readable data, while web services provide tools for the automatic using of that data. Thus, concept of Semantic Web Service (SWS) has been established [12][13].

2 .C: Semantic Annotation

The term “annotation” can denote both the process of annotating and the result of that process. An annotation attaches some of the data to some of other data. It establishes a relation between the annotated data and the annotating data within some of contexts[14]. Semantic annotations is to describing the meaning of certain parts of web information and the meaning of message elements employed increasingly by web services [8]

2.D: Reasoner

Inference engine is called Reasoner. Reasoner is a software application that obtains new facts or associations from existing information. The use of Reasoner in semantic web, allows applications to inquire why a particular conclusion has been reached; this means that semantic applications provide proof of their conclusions. Nowadays, there are many inference engines i.e. Jena

2.E: Matching

The matching is a correspondence operation between two concepts according to similarity features that is between them [4].The key of web service selection is service matching. In the traditional web services matching, service description is based on syntax that has low recall and precision. But, with the emergence of the Semantic Web, matching algorithms is based on ontology concept similarity and based on Quality of Service (QoS) optimization mechanism is regard as the secondary selection strategy to filtrate the candidate services [15].

2.F: Similarity Measure

Similarity is a measure that quantifies the dependency (independency) between two terms or two concepts. The similarity measure can represent similarity between two documents, two queries or one document and one query and also is a function that computes the degree of similarity between a pair of text objects. Two ways to measuring similarity follows:

- **Syntactic Similarity Measure**

  The syntactic similarity feature measures the unifiability of a candidate’s parse tree with the question’s parse tree and it uses syntactic restrictions as well as lexical measures to compute the unifiability of critical syntactic participants in the parse trees [16].

- **Semantic Similarity Measure**

Semantic similarity is the similarity of the action in the candidate sentence to the action asked in the question and is measured Similarity on main verbs [16]. The semantic similarity measuring techniques can be classified into three classes as follows: The first and normal way measures the semantic similarity by using ontology or taxonomy (e.g. WordNet) to calculate the distance. The second class of techniques uses training corpora and information content (IC) to estimate the semantic similarity and relatedness between two concepts. In this class, the techniques use machine learning, statistical-based, rule-based or other corpus-based approaches. The third class simply includes the techniques that employ a combination form the first two classes [17].

2.G: Description Languages of Web Services

A web service usually has one or more operations. The input, output, precondition and effects (IOPE) are generally used to describe web service operation. Moreover, the syntactic name and the semantic annotation of Web service operation indicate its functionality[18].Thus, languages such as UDDI, WSDL, SOAP, WSDL-S, OWL-S and so on are define standard ways for service discovery, description and invocation (message passing). The description of web services is divided into two groups:

- **Syntactic Description**

The main goal of syntactic descriptions of web services such as the Web Service Description Language is to describe interfaces of web services. The web services at the syntactical level are insufficient for creating meaningful descriptions of web services[19].WSDL does not support from the specifications of various constraints, management statements, classes of service, Service Level Agreement (SLAs) and other contracts and protocols between web services. Thus, the concept of Semantic Web Service (SWS) has been established[20][21].

- **Semantic Description**

Service descriptions are provided in a semantic framework that is the combination of ontologies, and also they are used to describing software artefacts. One example of a language that facilitates capability-driven description of services is OWL-S and other semantic description languages there are such as WSDL-S, WSMO, SAWSDL and etc.

OWL-S is made based on Web Ontology Language and consists of different Ontologies for describing Web service. Because OWL-S uses from ontologies to describe web services, web services and their behavior are become machine interpretable and thus tasks such as discovery and composition can be automated. OWL-S provide using of three different Ontologies: a Service Profile that states what the web service does, a Service Model that describe how the Web service performs the tasks and a Service Grounding that describes how to access the web service[20].

WSMO is another initiative to develop specifications for SWSs and is a framework for describing Web services. It consists of four top-entities: *Ontology*, *Web Service*, *Goal* and

Mediator; *Ontology* provides the terminology that is used by other WSMO elements. *Web Service* describes the capabilities, interfaces and internal working of web services. *Goal* represents user desires and *Mediator* provides bridges between different *Ontologies*. WSMO uses from a specific designed language that is called Web Service Modeling Language (WSML) and. Also it contains a grounding feature to link concepts with WSDL data types that can be achieved automatic invocation [20].

As well as, SAWSDL is a set of extensions for WSDL that provides a standard description format for Web services. Indeed, SAWS-DL extends WSDL with pointers to semantics that are crucial for achieving automation. The major technologies for Web services are SOAP and WSDL[10].Semantic web services combine the semantic web technology with web service technology, thus it enable automated discovery, selection, composition and execution of web services[21]. As a result, WSDL solves the need of interoperability in a technical manner, but is lacks adding semantic information.

2.H: Quality of Services (QoS)

Quality of Service (QoS) defines the non-functional requirements of service such as response time, price, availability etc. QoS properties are divided into two sub-categories measurable (throughput, response time, and latency etc.) and non-measureable (reputation and security etc.). Considering QoS aspects is important when deciding services to include in a service composition schema[22].

III. Web Service Composition: Current Methods

III. A: Web Service Composition

Web service composition involves the combination of a number of existing web services to produce a more complex and useful service. The composition of web services is a topic that attracts the interest of researchers. It offers complex problems process ability even with simple existing web services while cooperating with each other. Web service composition is an important technology of SOA that is in a complex and distributed environment and still there are many potential problems[23]. One of main targets of Web service composition is reusing existing web services and composing them into a process. Such programs enable user to manually specify a composition of programs to perform a task, but it is already beyond the human capability to deal with the whole process manually. Despite all efforts, the web service composition still is a highly complex task and generally, the complexity comes from the following sources[2]:

- First, the number of web services available on the Web is increasing dramatically during the recent years and can expect to have a huge repository of web services for searching.
- Second, Web services can be created and updated on the fly, thus the composition systems needs to detect the updating at runtime.
- Third, Web services can be developed by different organizations that use with different concept models for description of the web services. However, not exist unique language to defining and evaluation the Web services.

The variety of composition techniques can be classified according to two approaches. The first is syntactic composition based on syntactic description and other is semantic composition based on semantic description. We will see in this section several approaches for web service composition based on syntactic and web service composition based on semantic and we will discuss their limitations with regards to its requirements. The overall, the composition of web services can be done in a static or dynamic way that in the following is described. Web service composition contains three methods are: Manual/Static Composition, *Automatic/Dynamic Composition*, *Semi-automatic/dynamic or static Composition*. *Manual/Static Composition* is at syntactic groups. *Automatic/Dynamic Composition* and *Semi-automatic/dynamic/static Composition* are at semantic groups. The service composition consists of four different models[3]: *workflow-based*, *artificial intelligence (AI) planning-based*, *semantic- based*, and *graph-based*.

- **Manual/Static Composition**

The static means that the requester should build an abstract process model before beginning the composition; the abstract process model includes a set of tasks and their data dependency. Each task contains a query clause that is used to search the real atomic web service to fulfill the task [2]. Two possible approaches are currently investigated for the static web service composition. The first approach, referred to *web services orchestration;* this approach combines available web services with adding a central coordinator that is responsible for invoking and combining the single sub-activities. The second approach, referred to *Web services choreography* that does not assume the exploitation of a central coordinator, but it defines complex tasks via the definitions of the conversation that should be undertaken by each participant [9]. In static composition, the aggregation of the services is done at design time and composition is performed manually means that each web service is executed one by one in order to achieve the desired goal/requirement [24]. This type of composition is not is not flexible. There are many proposed manual web service composition techniques. These manual techniques are usually used for designing business processes in workflow management systems. Also different languages (e.g. BPEL4WS) are proposed for specifying composition. But as it may seem, these techniques are just usable by software developers, not by end users. In other words, manual composition of web services needs some programming experiences [25].

To compose services by labor force, traditionally there are two distinct design approaches: *top-down* and *bottom-up*. *Bottom-up* is that, at first, the potential partner WSs is identified (they are concrete executable services) and then them is connected with specific process logic. Another one, *top-down* design, is entirely different. It starts from specifying business process (workflow) consisted of abstract non-executable activities and subsequently, choosing a fittest concrete service for each activity. As a fatal drawback, manual composition relatively demands for much higher cost. Therefore, currently a large proportion of research efforts are dedicated to automations instead of costly and time-consumed manual composition that is trying to thoroughly eliminate human intervention. The research concerns of automatic composition are various technical aspects regarding how to automatically and efficiently generate composite services that exactly meet the expectations of requesters[26]. Today, a lot of techniques are proposed for manual service composition (e.g. workflow management systems). But creating composite services manually is hard and time consuming task for the user.

Dong and YU et al [27] proposes static framework and a method to model the workflow of web service composition based on BPEL using High-level Petri Nets(HPNs). The corresponding HPN is constructed by analyzing the structure of web service composition based on BPEL. The relationship between BPEL conceptions and HPNs is specified in four levels that are inter-service, intra-service, inter-activity and intra-activity. The verification process of composition of WSs is using HPN. After translation, the equivalent HPN of the web service composition based on BPEL can be verified based on existing tools.

Hamadi and Benatallah[28] propose a Petri net-based algebra for web service composition. This proposed approach is a static approach for web service composition. Any web service expressed by using the algebra constructs that can be translated into a Petri net representation. The Petri net represents the behavior of a service and contains one input place for absorbing information contains and so one output place is for emitting information. The algebra allows the creation of new value-added web services using existing ones as building blocks. They use from a formal model for verification of properties and the detection of inconsistencies both within and between web services.

- **Automatic/Dynamic Composition**

The dynamic composition creates process model and selects atomic services automatically and requires the requester to specify several constraints including the dependency of atomic, the user’s preference and so on[2] Since, Manual web service composition is time-consuming and hard task, the automatic or computer aided (semiautomatic) composition of web services is a recent trend. Also the need for automatic composition of web services with the increasing number of web services available on the internet is become more and more important[29]. The automation[2]means that either the method can generate the process model automatically or the method can locate the correct services if an abstract process model. For automatic web service composition, Semantic web is proposed. Ontology is used to give well-defined meaning to the semantic web [25]. Semantic web service composition [13] is a feature that improves the flexibility of the system. An automatic web service composer should compose ‘‘right” services in a composition according to the user’s specification. The ability of automatic composition of web service for creating a new composite web service is one of the key features for the future of the semantic web. Furthermore, composite web services [30] are dynamic that their components can be automatically selected at run-time based on specific requests. Mostly automatic composition techniques are either interface based or functionality based. In interface based composition, inputs and outputs through interfaces of users obtains composite web services and after composition the desired results are achieved. The drawback of it this is that functionality is not guaranteed, whereas in functionality based composition, user provides the formula that explains logic with interface information. Most of these methods are based on Artificial Intelligence (AI) planning [24]. There are many problems in artificial intelligence that are very similar to the automatic web service composition: artificial intelligence planning, automatic software generation, automatic workflow generation, logical deduction, etc [25]. In automatic composition of web service agents are used to select a web service that may be composed of multiple web services, but from user’s viewpoint, it is considered as a atomic service [31]. There are some successful methods for automatic web service composition that the following mentioned:

Talantikite et al[4] present a model automatic for Web Services Discovery and its Composition. In order to understandable descriptions, Semantic Annotation is used for web service Discovery and composition. The proposed approach uses from an inter-connected network of semantic Web services describing in OWL-S using the similarity measure between concepts like pellet before any submitted request. Their proposed approach gives several composition types: serial, dependent parallel and independent parallel. The Semantic Network is explored in backward chaining and depth-first in a single pass. At the end, are obtained several composition plans that satisfy the request and only one optimal composition plan using QoS is returned to the requester.

Paikari and Livani et al [13] proposed an automatic framework for composition of semantic web services in P2P network and they use from an algorithm that is based on a phased algorithm that the composition is done step to step. This algorithm matches the output of semantic web services of the previous phase with a new one whose inputs using Matchmaking. The framework is modeled by Multi-agent System Engineering methodology which is a famous agent oriented methodology and a top-down approach. This approach is consist four agents: UI Provider , Service Finder, Service Provider and Composer. For describing web services has been used from OWL. The composition process is performed through several steps. At each step composer sends its request for proper next web service to a service finder.

Shanfeng and Xinhuai et al [29] presented a mechanism to classify web services into different types based on function automatically and then they design a web services composition system based on service classification and AI Planning that consists of two main parts: service management sub-system and service provision sub-system. The service management part is based on the service classification management mechanism and service provision part is to meet the need of users' request by AI Planning. Also they focused on the classification of Web services. In this process, they compare web service instance with existing web service types by computing their similarity with comparing their semantic descriptions and then they designed a service management system that is a part of their service composition system. Finally according to user's request as input, using AI Planning Engine have generated a suitable composition plan for meet user's request.

Van and Zhijian et al[21]a new dynamic composition algorithm of semantic web service based on QoS ontology is proposed. Whereof, QoS ontology is an essential component because it provides non-functionality aspects of service. For this reason, they are present a hierarchical QoS ontology QoSHOnt which is consists of three layers: upper, middle and lower. Their algorithm chooses the best service according to the quality of service.

YAN and XUE et al [32] was asses sed web services ontology and Ant Colony (AC) algorithm. They proposed a method of composition of semantic web services that is based on AC algorithm. Then, they generate a graph of input and output of semantic web services by using this method. Then the composition of web services is transformed into finding a satisfying path in the graph. They focus on the quality and efficiency of composition of web services that is based on users' requests in the field of dynamic composition of web services and OWL-S is used for description of web services and their relationships. This paper proposes an optimal AC algorithm for composition of web services to ensure that the best composition of web services can be gotten in less time.

- **Semi-automatic/dynamic or static Composition**
  Semi-automatic composition is the same automatic composition, but with the difference that according to the different conditions, it chooses different processes to accomplish goals. In this processes, the fixed processes and implementation of processes can be variability reflects the semi-automatic feature [32].

Wang and Guttula et al [18] presented a semi-automatic approach for web service composition that including both data mediation and service suggestion algorithms. This paper seeks to aid users trying to compose web services into a process by providing service suggestions. A graph IODAG (Input Output Directed Acyclic Graph) is defined to formalize an input/output schema of a Web service operation. Three data mediation algorithms leaf-based, structure-based and path-based are developed to address data heterogeneities in process design. For adding semantic description into web services have used from Semantic Annotation for WSDL and XML Schema. This approach utilized various types of annotations and QoS. Finally they have developed a data mediation approach that tries to find automatically the optimal mappings between outputs and inputs. And finally came to the conclusion that path-based algorithm is best data mediation algorithm from other two algorithms.

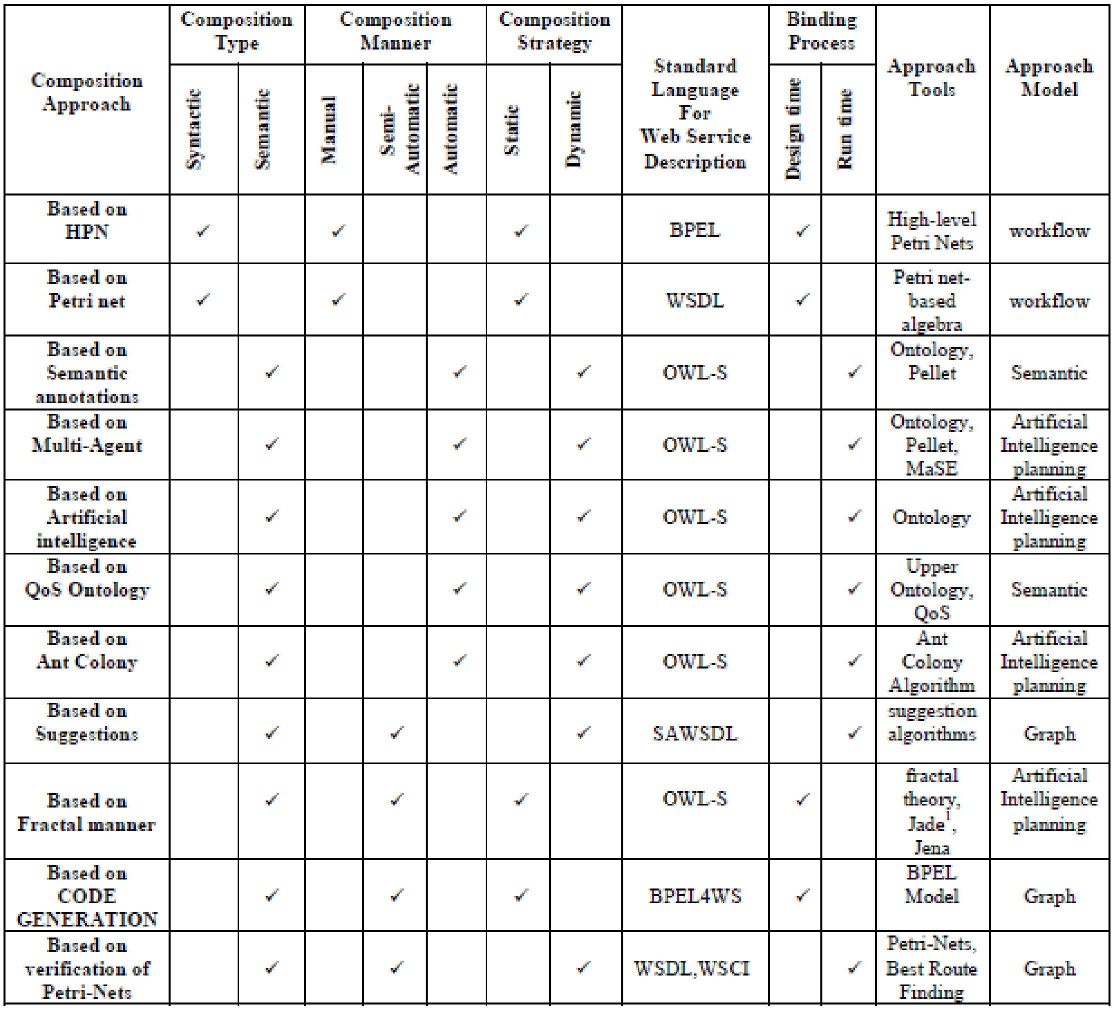
Adrian et al [33] present a semi-automatic approach. In their proposed approach the composition is done in a fractal manner using existing web service chains that can easily be incorporated in new web service chains. All web service chains in this framework are described using Web Service Business Process Execution Language (WS-BPEL) and be used as building blocks to create new service chains. All services are managed by intelligent agents. Besides the intelligent agents, other components of their framework are the annotated web services semantically. In this approach, existing web service chains are combined in a fractal like manner to easily create new and more complex web service chains.

Talib et al [34] have suggested a method that semi-automatically generate static web service composition in BPEL4WS language. Their proposed system captures minimum necessary information from the user and stores them in a relational model and also captures the information required to develop the composition from the composition modeler and WSDL interfaces of the collaborating partners. Finally the transformation algorithm with all captured information is applied to map from relational to BPEL model.

Chan et al [35] provided a dynamic web service composition with verification of Petri-Nets and their proposed approach is a semi-automatic approach. Their method is based on two standard web service languages that are WSDL and WSCI and composes the Web services with the information provided by these two standards; WSDL describes the entry points for each available service and WSCI (Web Service Choreography Interface) describes the interactions among WSDL operations. After the composition, they verify the web service to be deadlock free with modeling the Web service as a Petri-Net. In this approach, they use from N-version Programming web for composing web services to improve the reliability of the overall system. Then, the composition of the Web service with the information obtained in the composition procedure is performed and produced the Web services composition with the Best Route Finding system (BRF). Finally, Petri-Net is employed to verify the correctness of the composed web service.

3 .B: Summary

According to studied papers, various parameters are effective in the field of web service composition. We classify papers by using these parameters that have been shown in Table 1. One of the most effective parameters in composition of web services is composition type. The composition types are syntactic and semantic that is discussed in section III. In [27], [28] is used from syntactic composition type; in this way the composition is performed manually. But, manual composition is time-consuming and hard task. Therefore in [4], [13], [29], [21], [32] are used from semantic to resolve disadvantages of manual composition that its composition manner is automatic and is not required to user intervention for performing composition. Static and Dynamic composition are two type of strategy that be discussed in section III. In static composition, composition of web services is performed in design time that is required to user intervention, whereas dynamic composition is performed in run time and is not required to user intervention. Other type of composition manner is semi-automatic. In this case, the composition strategy is either static or dynamic. In [18], [33], have been presented a semi-automatic/static composition and in [34] is presented the semi-automatic/dynamic composition. In each of papers is used from tools for composition of web service and standard languages such as wsdl, owl-s and etc for description of web services. The overall, we summarize overall framework in table 1 and so we classify the advantages and disadvantages of each reviewed paper and their characteristics in table 2.



Composition Approach	Composition Type		Composition Manner			Composition Strategy		Standard Language For Web Service Description	Binding Process		Approach Tools	Approach Model
	Syntactic	Semantic	Manual	Semi-Automatic	Automatic	Static	Dynamic		Design time	Run time		
Based on HPN	✓		✓			✓		BPEL	✓		High-level Petri Nets	workflow
Based on Petri net	✓		✓			✓		WSDL	✓		Petri net-based algebra	workflow
Based on Semantic annotations		✓			✓		✓	OWL-S		✓	Ontology, Pellet	Semantic
Based on Multi-Agent		✓			✓		✓	OWL-S		✓	Ontology, Pellet, MaSE	Artificial Intelligence planning
Based on Artificial intelligence		✓			✓		✓	OWL-S		✓	Ontology	Artificial Intelligence planning
Based on QoS Ontology		✓			✓		✓	OWL-S		✓	Upper Ontology, QoS	Semantic
Based on Ant Colony		✓			✓		✓	OWL-S		✓	Ant Colony Algorithm	Artificial Intelligence planning
Based on Suggestions		✓		✓			✓	SAWSDL		✓	suggestion algorithms	Graph
Based on Fractal manner		✓		✓		✓		OWL-S	✓		fractal theory, Jade[1], Jena	Artificial Intelligence planning
Based on CODE GENERATION		✓		✓		✓		BPEL4WS	✓		BPEL Model	Graph
Based on verification of Petri-Nets		✓		✓			✓	WSDL,WSCI		✓	Petri-Nets, Best Route Finding	Graph

Table 1: Characteristics Reviewed Approaches of web service composition

Composition Approach	**Advantages**	**Disadvantages**
Based on HPN	Optimized composition plan	Static and Manual composition, time-consuming task, human intervention, non-automatic
Based on Petri net	Use of algebra for Semantic description web services, verification of properties and the detection of inconsistencies both within and between services, Avoiding deadlock	Static and Manual composition, time consuming task, human intervention, non-automatic, The lack of an optimal combination of plan Checking
Based on Semantic annotations	semantic annotations, dynamic and automatic composition, Lack of user intervention, composition plan several creation according to Quality of Service, Semantic Network creation, Prevent deadlocks using Petri-Net	High Storage space, time-consuming, Performance decreases with increasing number of web services, Increase response time to requests
Based on Multi-Agent	Semantic approach, Flexible, Hide the searching complexity from the user using MaSE	Requires high analysis, Do web service composition as step by step, lack of selection an optimal combination
Based on Artificial intelligence	Automatic and dynamic composition, Easier manage large numbers of web services and retrieve them, Lack of user intervention, efficient and scalable	lack of selection an optimal combination, Ignoring the Quality of Service
Based on QoS Ontology	Automatic and dynamic composition, Consideration Quality of Service, selection the best service according to the quality of service, execution service composition instantaneously without predefined template, Reduction service composite time	the template needs people's intervention, Reduction degree of automation of service composition
Based on Ant Colony	Automatic and dynamic composition, The use of graph structure and , semantic description for web service composition, high successful rate of services composition, ensuring of the quality and efficiency of composition, reduction time for best composition	Lack of Consideration Quality of Service
Based on Suggestions	Use of graph structure, the use of SAWSDL for semantic description, Semi-automatic approach	High storage space, Ignoring the Quality of Service, Performance decreases with increasing number of web services, Lack of fully semantic approach
Based on Fractal manner	Semi-automatic**,** Consideration Quality of Service, the use of agent-based, reduction of the development	Lack of fully semantic approach, complexity of the problem

	time	
Based on CODE GENERATION	Semi-automatic approach, semi-semantic description for web services,	Static composition, Lack of fully semantic approach, Lack of Consideration Quality of Service, lack of selection an optimal combination
Based on verification of Petri-Nets	Prevent deadlocks using Petri-Net, semi-automatic and dynamic composition, selection of an optimal combination plan	Lack of fully semantic approach, Ignoring the Quality of Service

Table 2: Reviewed Approaches Advantages and Disadvantages

4. CONCLUSION

In this paper we described a comparative study of recent approaches for Web Service composition. We review web service composition models namely, the semantic web service composition model and syntactic web service composition model; the syntactic composition that is based on syntactic description and semantic composition that is based on semantic description. The composition of web Services includes three methods that each of which are belong to corresponding syntactic or semantic groups are Manual/Static Composition, Automatic-/Dynamic Composition, Semi-automatic/dynamic or static Composition. In this paper, we provide an overview of service composition methods and approaches and we discuss the advantages and disadvantages of each investigated paper.

References

[1] Hadi Saboohi, Amineh Amini, and Hassan Abolhassani, "Failure Recovery of Composite Semantic Web Services using Subgraph," in Proceedings of the International Conference on Computer and Communication Engineering 2008, Kuala Lumpur, 2008, pp. 489 - 493.

[2] Jinghai Rao and Xiaomeng Su, A Survey of Automated Web Service Composition Methods.: Springer Berlin Heidelberg, 2005, vol. 3387.

[3] YU Qing-mei, WANG Lan, and HUANG Dong-mei, "Fishery Web Service Composition Method Based on Ontology," Journal of Integrative Agriculture, vol. 11, no. 5, pp. 792–799, May 2012.

[4] Hassina Nacer Talantikite, Djamil Aissani, and Nacer Boudjlida, "Semantic annotations for web services discovery and composition," Computer Standards & Interfaces 31 (2009) 1108–1117, vol. 31, no. 6, pp. 1108-1117, November 2009.

[5] Wonhong Nam, Hyunyoung Kil, and Dongwon Lee, "Type-Aware Web Service Composition Using Boolean Satisfiability Solver," in 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, Washington, DC, 2008, pp. 331 - 334.

[6] Mutaz Beraka, Hassan Mathkour, Sofien Gannouni, and Hussein Hashimi, "Applications of Different Web Service Composition Standards," in 2012 International Conference on Cloud Computing and Service Computing, Shanghai, 2012, pp. 180 - 185.

[7] Pablo Rodriguez Mier, Manuel Mucientes, Manuel Manuel, and Miguel I. Couto, "Composition of web services through genetic programming," Evolutionary Intelligence, vol. 3, no. 3-4, pp. 171-186, December 2010.

[8] Jorge Cardoso, Semantic Web Services: Theory, Tools, and Applications. New York, United States of America, 2007.

[9] Antonio Bucchiarone and Stefania Gnesi, "A Survey on Services Composition Languages and Models," in Bucchiarone, Gnesi, 2006, pp. 56-58.

[10] Jacek Kopecky, Tomas Vitvar, Carine Bournez, and Joel Farrell, "SAWSDL: Semantic Annotations for WSDL and XML Schema," Internet Computing, IEEE, vol. 11, no. 6, pp. 60 - 67, 2007.

[11] Toktam Ghafarian and Mohsen Kahani, "Semantic Web Service Composition Based on Ant Colony Optimization," in Networked Digital Technologies, 2009. NDT '09. First International Conference on, Ostrava, 2009, pp. 171 - 176.

[12] Yasmine Charif and Nicolas Sabouret, "An Overview of Semantic Web Services Composition Approaches," Electronic Notes in Theoretical Computer Science, vol. 146, no. 1, pp. 33-41, January 2006.

[13] Elham Paikari, Emadoddin Livani, Mohammad Moshirpour, Behrouz H Far, and Günther Ruhe, "Multi-Agent System for Semantic Web Service Composition," in Knowledge Science, Engineering and Management.: Springer Berlin Heidelberg, 2011, pp. 305-317.

[14] S Handschuh, "Creating Ontology-based Metadata by Annotation for the Semantic web," Ph.D. thesis, University of Karlsruhe, 2005.

[15] GUO Song, Xinyang China, and Wuhan China, "A New Approach for Web Service Composition Based on Semantic," in Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on, Wuhan, 2010, pp. 1-4.

[16] Abolfazl Keighobadi Lamjiri, Leila Kosseim, and Thiruvengadam Radhakrishnan, "Comparing the Contribution of Syntactic and Semantic Features in Closed versus Open Domain Question Answering," in Semantic Computing, 2007. ICSC 2007. International Conference on, Irvine, CA, 2007, pp. 679 - 685.

[17] Hoa A Nguyen and Hisham Al-Mubaid, "New Ontology-based Semantic Similarity Measure for the Biomedical Domain," in Granular Computing, 2006 IEEE International Conference on, 2006, pp. 623 - 628.

[18] Rui Wang et al., "Web Service Composition using Service Suggestions," in 2011 IEEE World Congress on Services, Washington, DC, 2011, pp. 482 - 489.

[19] Ahtisham Muhammad Aslam, Jun Shen, and Soren Auer, "An Integration Life Cycle for Semantic Web Services Composition," in Proceedings of the 2007 11th International Conference on Computer Supported Cooperative Work in Design, 2007, pp. 490 - 495.

[20] Jordy Sangers, Flavius Frasincar, Frederik Hogenboom, and Vadim Chepegin, "Semantic Web service discovery using natural language processing techniques," Expert Systems with Applications, vol. 40, no. 11, pp. 4660–4671, September 2013.

[21] Hai Van, Wang Zhijian, and Lu Guiming, "A Novel Semantic Web Service Composition Algorithm Based on QoS Ontology," in 2010 International Conference on Computer and Communication Technologies in Agriculture Engineering, vol. 2, Chengdu, 2010, pp. 166 - 168.

[22] Furkh Zeshan and Radziah Mohamad, "Semantic Web Service Composition Approaches: Overview and Limitations," International Journal on New Computer Architectures and Their Applications, pp. 640-651, 2011.

[23] Hu Yan and Wang Hui, "Constraints in Web Services Composition," in IEEE, Dalian, 2008, pp. 1 - 4.

[24] G Vadivelou and Dr E Ilavarasan, "Solution to Dynamic Web Service Composition related to QoS," in Electronics Computer Technology (ICECT), 2011 3rd International Conference on, vol. 5, Kanyakumari, 2011, pp. 351 - 355.

[25] Soheil Hassas Yeganeh, Jafar Habibi, Habib Rostami, and Hassan Abolhassani, "Semantic web service composition testbed," Computers and Electrical Engineering 36 (2010) 805–817, vol. 36, no. 5, pp. 805–817, September 2010.

[26] Yang Syu, Shang-Pin Ma, Jong-Yih Kuo, and Yong-Yi FanJiang, "A Survey on Automated Service Composition Methods and Related Techniques," in 2012 IEEE Ninth International Conference on Services Computing, Honolulu, HI, 2012, pp. 290 - 297.

[27] Wen-Li Dong, Hang YU, and Yu-Bing Zhang, "Testing BPEL-based Web Service Composition Using High-level Petri Nets," in Proceedings of the 10th IEEE International, Hong Kong, 2006, pp. 441 - 444.

[28] Rachid Hamadi and Boualem Benatallah, "A Petri Net-based Model for Web Service Composition," in Australian Computer Society, vol. 17, 2003, pp. 191-200.

[29] Shanfeng Qi, Xinhuai Tang, and Delai Chen, "An Automated Web Services Composition System Based on Service Classification and AI Planning," in 2012 Second International Conference on Cloud and Green Computing, Xiangtan, 2012, pp. 537 - 540.

[30] Guisheng Fan, Huiqun Yu, Liqiong Chen, and Dongmei Liu, "Petri net based techniques for constructing reliable service composition," The Journal of Systems and Software, vol. 86, no. 4, pp. 1089–1106, April 2013.

[31] K Rajeswari, E Jeevitha, Stephy R Graph, and Suresh M Kumar, "Dynamic Web Service Composition through Network Analysis with Backtracking," Undergraduate Academic Research Journal (UARJ), vol. 1, no. 1, pp. 40-45, 2012.

[32] Kai YAN, Gang XUE, and Shao wen YAO, "An Optimization Ant Colony Algorithm for Composition of Semantic Web Services," in 2009 Second Asia-Pacific Conference on Computational Intelligence and Industrial Applications, vol. 2, Wuhan, 2009, pp. 262 - 265.

[33] Cotfas Liviu Adrian, Diosteanu Andreea, and Smeureanu Ion, "Fractal Web Service Composition Framework," in Communications (COMM), 2010 8th International Conference on, Bucharest, 2010, pp. 405 - 408.

[34] Muhammad Adeel Talib and Zongkai Yang, "SEMI-AUTOMATIC CODE GENERATION OF STATIC WEB SERVICES," in Engineering, Sciences and Technology, 2004, pp. 132 - 137.

[35] Pat. P. W. Chan and Michael R. Lyu, "Dynamic Web Service Composition: A New Approach in Building Reliable Web Service," in 22nd International Conference on Advanced Information Networking and Applications, Okinawa, 2008, pp. 20 - 25.