Indian Journal of Science and Technology, *Vol 8(33), DOI: 10.17485/ijst/2015/v8i33/72733, December 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Novel Rubric and Feature-based Appraisaland Comparison Framework for the Evaluationof Semantic Web Services Composition Approaches

Samira Ghayekhloo* **and Zeki Bayram**

Department of Computer Engineering, Eastern Mediterranean University, Famagusta, Northern Cyprus, Mersin 10, Turkey; sghayekhloo@gmail.com, zeki.bayram@emu.edu.tr

Abstract

Background/Objectives: This article proposes a novel methodology based on rubrics and feature-based schemes for the appraisal and comparison of approaches to semantic Web services composition and discovery. **Methods/Statistical analysis:** In order to evaluate the Semantic Web services composition and discovery approaches we created a new framework called RFSWS. This framework is the combination of traditional feature-based evaluation schemes and newly developed analytic rubrics tables. Five recently introduced prominent Semantic Web services composition approaches were identified, explained, and then evaluated/compared using RFSWS. **Findings:** In this work we determined aspects of Semantic Web services composition and discovery processes that can be evaluated as performance criteria in rubric tables. This is a novel application of rubrics, which have traditionally been used for grading student performance by teachers. We created a novel framework called RFSWS consisting of rubric tables and a feature-based evaluation scheme for the evaluation and comparison of Semantic Web service discovery and composition approaches, and applied it in the evaluation of five recently introduced prominent Semantic Web services composition approaches. Considering the shortcomings of existing Semantic Web services composition approaches that were discovered through this evaluation, we proposed an idealized dynamic Semantic Web service discovery and composition method, a yardstick by which all future Semantic Web services composition approaches can be evaluated. **Application/ Improvements:** Our novel RFSWS framework can be applied in the comprehensive and systematic evaluation/comparison of Semantic Web services discovery and composition approaches.

Keywords: Composition, Discovery, Evaluation, Feature, Rubric, Semantic Web Service

1. Introduction

Two common types of WSs are SOAP-based Web services, which use Web Service Description Language (WSDL) for their description, and RESTful Web services, which conform to the REST architectural principle[1]. From the information technology viewpoint, WSs are loosely coupled, platform independent, and are accessible through programming over the internet.

Existing proposals regarding WSC have been surveyed and the results presented in several works in the last decade. These surveys have some important shortcomings, such as not stating clearly what requirements need to be met for an approach to successfully solve the problems of SWSC, and lacking in the level of detail and precision with which they present and compare approaches to WSC.

Such shortcomings prompted us to develop for the first time analytic rubrics, as well as a feature-based comparison scheme, for evaluating and comparing of SWSC approaches. We called our method Rubric and Feature-based appraisal and comparison framework for Semantic Web Services (RFSWS). RFSWS includes some of the key composition issues and requirements that were identified based on a comprehensive study of the literature on SWSC methods. We then proceeded to actually evaluate five prominent approaches for SWSC introduced since 2011 in order to determine their relative strengths and weaknesses, and finally come up with an idealized

* *Author for correspondence*

approach for SWSC. Our work is novel not only in its development and use of special rubrics (which have been traditionally used by teachers in grading the students' performance, Wolf and Stevens[2]) for the evaluation of SWSC approaches, but also in the level of detail in which the evaluation has been made, and in the specification of an idealized SWSC approach which can be the yardstick with which future SWSC approaches can be compared to. Furthermore, it distinguishes itself from other recent surveys on WSC approaches[3,4,5] in that it surveys and compares the state-of-the-art SWSC approaches.

The rest of paper is organized as follows. Section 2 gives background information on SWSs and WSC. In section 3 we present our RFSWS framework for the comparison of SWSC approaches, which involves specially developed rubrics and a feature-based evaluation scheme. A brief overview of the approaches surveyed in this paper is given in section 4, and the actual evaluation and comparison of the approaches are carried out in 5. In section 6 we discuss the positive and negative of the SWSC approaches. Section 7 presents our notion of what features and characteristics an ideal SWSC approach should aspects possess. Finally section 8 is the conclusion.

2. Semantic Web Services Composition

In recent years, due to the increasing the number of WSs and complexity of users' demands, traditional WSs have not been able to answer complex user requests adequately. In many instances, the user's request cannot be answered by just one service, and several services must be combined to produce the required result. This job must be done manually if traditional WSs are used. SWSs automate process of WS discovery, selection and composition through Semantic annotation of WSs and expressive definition of user desires in the form of goals. SWS description languages based on Web Service Modelling Ontology (WSMO)[3], as well as Web Ontology Language for Services (OWL-S)[4], for example, are rich enough to allow such a definition of goals and WSs. Intelligent software agents can then carry out these activities on behalf of human users, minimizing the need for human intervention in the process[5]. One of the important building blocks of SWSs is ontology. Ontology is defined as a formal, explicit specification of a shared conceptualization[6]. In the context of SWSs, ontology provides a common vocabulary to denote the types in the form of classes or concepts, properties and interrelationships of concepts in a specific domain.

3. A Framework for Comparison and Appraisal of Semantic Web Service Composition Approaches

3.1 Evaluation through Rubrics

3.1.1 Introduction to Rubrics

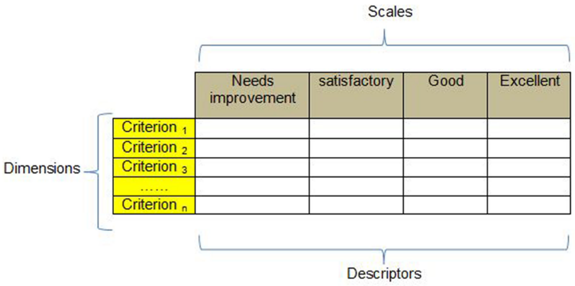
A rubric, in its traditional role, is a scoring and instructional tool used to assess student performance using a task-specific set of criteria, providing informative feedback to the instructor regarding the level of understanding on the part of the students, as well as informing students about the expectations of instructors from their work. To measure student performance a rubric contains the essential criteria for the task and levels of performance (i.e., from poor to excellent) for each criterion. The meaning of each level of performance for each criterion is defined explicitly to permit objective evaluation.

There exist two types of rubric, namely holistic and analytic. In the former, the teacher scores the overall process or product as a whole, without taking into account the component parts singly[7]. In the latter, first the teacher lists all parts of product or process, then considers a score for each part, and at the end sums the individual scores to obtain a total score[11,12].

Rubrics generally contain three components:

- Dimensions
- Rating Scale
- Descriptors

Dimensions are generally referred to as performance criteria, the rating scale as levels of performance, and descriptors as definitions. Figure 1 depicts the general form of a rubric table.



	Needs improvement	satisfactory	Good	Excellent
Criterion $_1$				
Criterion $_2$				
Criterion $_3$				
……				
Criterion $_n$				

Figure 1. Structure of rubric table.

Table 1. Sample of research report rubric table[14]

Criterion	Score	Description
Amount of information	Needs Improvement	All topics not addressed or most questions answered with words or phrases instead of sentences.
	Satisfactory	All topics are addressed, and most questions answered with 1-2 sentences about each.
	Good	All topics are addressed and most questions answered, with at least 3 sentences about each.
	Excellent	All topics are addressed, and all questions answered, with at least 3 sentences about each.
Organization	Needs Improvement	There appears to be little organization of the material.
	Satisfactory	Information is generally organized, but no headings are used.
	Good	Information is organized with headings, but some material under the headings may be out of place.
	Excellent	Information is very well organized with headings that relate clearly to the material.
Quality of Information	Needs Improvement	Information gathered has little or nothing to do with the questions posed.
	Satisfactory	Information gathered provides answers to main questions, but no details and/or examples are given.
	Good	Information gathered provides answers to main questions along with 1-2 supporting details and/or examples.
	Excellent	Information gathered provides answers to the main questions along with several supporting details and/or examples for each.
Sources	Needs Improvement	Some sources for information and graphics are not documented.
	Satisfactory	Sources for information and graphics are documented, but most are not in the correct format.
	Good	Most sources for information and graphics are documented in the designated format.
	Excellent	Sources for information and graphics are documented in the designated format.

3.1.2 Rubrics for the Appraisal of SWSC Approaches

For the first time, we apply the technique of rubric tables outside the area of education, and develop rubrics for the evaluation and comparison of SWSC approaches. We then proceed in section 5 with the actual evaluation of several recent SWSC approaches using our rubrics. Since not all comparable attributes of the approaches are amenable to comparison using rubrics, we also develop a feature-based evaluation scheme, and use that scheme in concert with the rubric to obtain a better picture of the approaches under consideration. Tables 2, 3 contain our analytic rubric table. Our rubric table contains six important criteria needed in composition processes of SWSs, along with their descriptions. These criteria are automation, scalability, adaptivity, dynamicity heterogeneity, workflow pattern. Automation means what is the automation level of WSC approach, scalability shows how many WSs can the system deal with, adaptivity identify to what degree is the system flexible for modifying its behaviours in volatile environments and responding to significant changes at execution time, dynamicity means to what degree of dynamism can the approach combine WSs for user's request at runtime, heterogeneity means to what degree can the WSC approach deal with heterogeneity of WSs, and workflow pattern illustrates WSC approach uses which types of workflow patterns. Scores 1, 2, 3 and 4 illustrate rating scales of our rubric table namely, needs improvements, satisfactory, good and excellent respectively.

3.2 Feature-Based Evaluation Scheme

There exist some important criteria which have important roles in SWSC processes but cannot be easily evaluated using rubrics, because it is not possible to directly compare the offered solutions on a graded scale. Here we investigate them as feature-based criteria, which we use in conjunction with the analytic rubric represented in previous subsection for the appraisal and analysis of SWSC approaches.

In the rest of this section, features and sub-features items are explained briefly.

Table 2. Analytic rubric table for the appraisal of SWSC approaches (part1)

Criterion	Score	Description
1. Automation	**1**	Semi-automatic: User designs the overall architecture of WSs interactions and describes at a high level the requirements that participating WSs must satisfy. Actual WS discovery and composition take place automatically at runtime.
	2	Somewhat automatic: The system presents the user with its results and the user accepts the one that is most satisfactory for the job at hand.
	3	Mostly automatic: All the WSC procedures are done automatically by the machine and without user intervention; however the approach does not consider the non-functional properties.
	4	Fully automatic: User does not intervene in discovery and composition processes and all the WSC procedures are done automatically by machine which considers the non-functional properties and user preference as well.
2. Scalability	**1**	Approach works only with toy examples, using less than 10 WSs. This approach merely presents its idea in small size with possible development in future works.
	2	Approach can handle 10 to 99 WSs.
	3	Approach presents method for reasonable real-life cases, and can deal with WSs within the range of 100 to 1000.
	4	Approach is scalable to more than thousands of WSs.
3. Adaptivity	**1**	No adaptiveness: Presented method is not able to support any types of adaptation during WSC processes.
	2	Rule-based adaptation: Approach employs methods that rely on predefined event-condition-action (ECA) rules. Rules are activated whenever the events which they are bound to happen in the environment, but they are limited in covering all possible events and scenarios.
	3	Partial adaptation: Approach considers only some aspects of adaptation at runtime, such as, change in non-functional properties and/or addition or removal of WSs during WSC processes.
	4	Full adaptation: Approach is able to cope with all aspects of unpredicted change in functional and non-functional properties of WSs at runtime without interrupting the whole system operation.
	4	Sequential, and-split-join, conditional and iteration: WSC approach uses all the mentioned patterns. In the iteration pattern, WSs can be called repetitively.

1. Accepted SWS description methodology: This item is about methods for describing both functional and non-functional properties of WSs. Two popular ones are top-down and bottom-up methods.

Top-down: WSs are semantically described by providing a high level declarative specification of WS functionality and non-functional properties. Two prominent models which follow this method are Web Service Modelling Ontology (WSMO) and Web Ontology Language for Services (OWL-S). There also exist other special purpose languages for the semantic description WSs, such as DIANE Service Description (DSD) language[9].

Bottom-up: First the service developer generates WSs based on Web Service Description Language (WSDL)[10]. Then existing WSs are semantically annotated by different bottomup annotation models such as Web Service Description Language with Semantics (WSDL-S)[11]or Semantic Annotation for WSDL and XML schema (SAWSDL)[12].

2. Quality of Service (QoS): QoS deals with the quality aspects of a user's interaction with the WSs. The prominent QoS factors associated with the WS composition and execution are mentioned in[13,14]. Below we explain them briefly.

- Response time: Refers to the time needed to complete a user's request.
- Performance: Refers to the speed of the system for completing the user's request and is measured in terms of response time, latency, execution time and throughput. Latency is the round-trip delay time in sending a request by the user and receiving the response from the system, execution time is to the time taken by a WS to fulfil its series of activities and throughput refers to the number of requests which are served in a given period of time. In general, low latency, high throughput, low execution time and fast response time are the desired performance characteristics of WS.
- Cost: Is the amount of money needed in order to execute the related WSs for answering the user's request.

Table 3. Analytic rubric table for the appraisal of SWSC approaches (Part2)

Criterion	Score	Description
4. Dynamicity	1	Static: All processes of discovery, selection and composition of specified WSs take place manually at design time. Thus it cannot be an appropriate solution for the unpredicted user's request.
	2	Somewhat-dynamic: The processes of discovery and selection are performed statically, while the composition of WSs is done at runtime.
	3	Mostly dynamic: Approach creating an Abstract Composition Model (ACM) at design time, while discovery, selection and composition (actual linking of WSs according to the ACM) of WSs are done at execution-time.
	4	Fully dynamic: Approach implementing this type of dynamism is able to handle all the processes of discovery, selection and composition of WSs at runtime.
5.Heterogeneity	1	Approach assumes that all the candidate WSs which participate in WSC processes use the same description language.
	2	Approach requires an adaptor for each pair of cooperating WSs. (In this case with N cooperative WSs are needs to develop N^2 adaptors).
	3	Approach has no formal mediator component, but can solve the heterogeneity problem, in non-systematic ways.
	4	Approach uses in-built mediators to overcome the problems of heterogeneity and enable interoperability between WSs.
6. Workflow Pattern	1	Sequential: The sequential pattern defines sequential execution of WSs. A WS is invoked after the completion of previous one.
	2	Sequential and and-split-join: WSC approach uses both sequential and and-split-join patterns. And-split-join represents WSs that are executed simultaneously. The term join represents the synchronization constructor, which shows that the next WS is invoked when all parallel branches of WSs have been executed.
	3	Sequential, and-split-join and conditional: WSC approach uses sequential, and-split-join and conditional patterns. A conditional pattern represents the exclusive choice of branches to invoke the proper WS. In exclusive choice, exactly one of the conditions is permitted to be true, and the corresponding WS is executed.
	4	Sequential, and-split-join, conditional and iteration: WSC approach uses all the mentioned patterns. In the iteration pattern, WSs can be called repetitively.

- Availability: Is the readiness of the WS to accept and process requests. High availability shows that the WS is ready to use most of the time.
- Security: Since WSC processes work on the public Internet, loss, theft and modification of information is a real risk therefore security is a very important aspect of WSC processes that must be given full attention. The service provider should have different levels of security depending upon the needs of the service requestor. Sub-aspects of security include confidentiality, traceability, authorization and non-repudiation.

3. Composition methods: Different approaches use various methods to combine the WSs in order to satisfy the user's demand. Due to the large number of composition methods, we use the classification scheme proposed in[6]. The authors divide composition methods into four groups, namely AI planning, workflow-based, model-based and mathematic-based methods.

- AI-planning: Employs Artificial Intelligence planning (AI-planning) algorithms in order to combine WSs. The approach solves the problems of WSC by designing the set of actions for achieving the goals and generating a plan.
- Workflow-based: In this approach first an Abstract Composition Model (ACM) is designed, either manually or automatically by the workflow generator tools in accordance with the user's request. ACM specifies control and data flow among the tasks. Secondly, an algorithm is employed to find specific WSs that are matched to the tasks and bind relevant WSs together respectively.
- Model-based: This approach uses modelling languages like (UML, Petri net, etc.) to model service composition and overcome to the problems of complex requests.
- Mathematical-based: This approach presents its solu-

tion for solving the WSC problems based on mathematic structure and techniques such as: graph-based techniques, logic-based techniques and techniques based on process algebra.

- Other Methods: The last group of composition methods comprises all the approaches that do not fit in the aforementioned list and represents other methods for solving the WSC problem.

4. Execution of composite services: These approaches present different ways to execute the qualified composite service, either directly via in-built components or by the help of other standards.

- Self-execution: All the processes of composition and execution of WSs are done within the presented approach's components.
- Execution by other standards: The approach addresses discovery and composition of WSs to create composite services but devolves the execution of the composite service to other outstanding standards such as BPEL[15].

5. Personalization: Web personalization is the process of customizing WSs so that they match the particular user's needs and preferences[16]. Preference and context-awareness are two main factors of personalization.

- User Preference: By paying attention to the end user's desire or intention in service selection, it is possible to improve the quality of presented WSs and better achieve end user's satisfaction.
- Context Awareness: Context refers to the information about the end user and its environments, such as, name, address, current location of the user and type of device that the customer is using. Authors of[17] categorize methods for awareness of end user's context as: personal profile oriented, usage history oriented, process oriented and other methods.

4. Brief Introduction to SWSC Approaches

This section briefly reviews five state-of-the-art SWSC approaches are introduced, namely: DynamiCoS[18], PORSCE II and VLEPPO[1920],Bartalos[21], Top-k ASC[22] and Tang et al[23]. These approaches present different methods for solving the problems of SWSC. They have been selected after a comprehensive survey of the literature on current SWSC approaches.

4.1 DynamiCoS

Dynami CoS (Dynamic Composition of Services) [23] is a user-centric framework which was created for combining WSs at runtime to answer the user requirements. In this framework, services are created and published by the service provider at design-time but the processes of discovery, selection and composition are performed at runtime. As shown in Figure 2, DynamiCoS architecture has five modules, namely service creation, service publication, service request, and service discovery and service composition. In order to achieve automation in WSC processes, the framework semantically annotates the WS as a seven-tuple < ID, I, O, P, E, G, NF >, where ID, I, O, P, E, G and NF stand for service identifier, inputs, outputs, preconditions, effects, goals, and non-functional properties respectively.

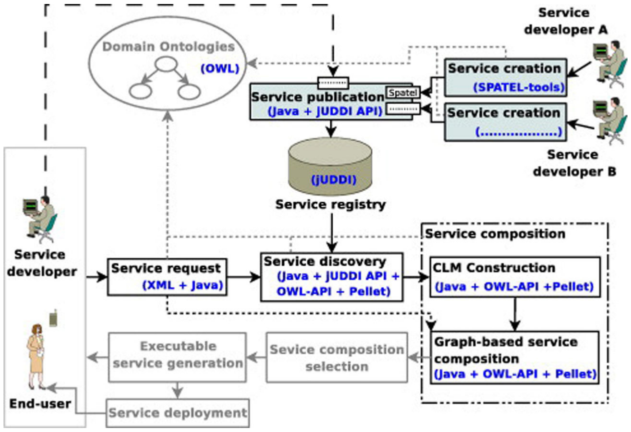


Figure 2. DynamiCoS Architecture[23].

The composition processes of DynamiCoS are:

- Requested WSs are discovered based on exact and partial matching of Inputs, Outputs, Preconditions and Effects (IOPE) concepts of WSs.
- Description of all discovered WSs are organized in a pre-processed structure called Casual Link Matrix (CLM)[24] which stores all possible semantic connections (or causal links) between the discovered services' input and output concepts.
- A graph-based composition algorithm[25] is used to find set of composed services according to the user demand via the prepared CLM matrix.

4.2 PORSCE II and VLEPPO

PORSCE II and VLEPPO[24,25] is a framework for modelling the SWSC problem as a planning problem, expressed in the Planning Domain Definition Language (PDDL)[26], and then applying a variety of external planners to get a solution plan for obtaining the end user's goal. Figure 3 depicts

the architecture of this approach which contains two software systems, namely PORSCE II[27] and VLEPPO[28],[29]. Implementation of the approach is performed through the integration of these two systems. The former performs all the tasks related to WSs, such as transforming OWL-S description of WSs to PDDL, accuracy measurement of the composed service, etc., while the latter deals with the planning steps. Key features of this approach are: 1. It is able to be used by the non-expert user through the dialog interface in PORSCE II, 2. It allows the preparation of an approximate composition service, and 3. Independence between the representation and solving parts makes the approach flexible in the choice of external planners.
SWSC steps in PORSCE II and VLEPPO are:

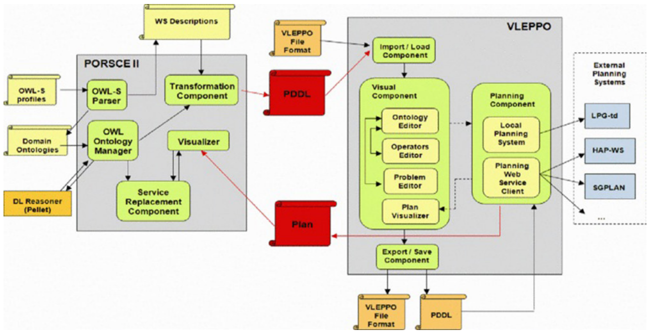


Figure 3. PORSCE II and VLEPPO architecture[24].

- Translation of the WSC problem into the planning problem, which is donein PORSCE II, comprising the transformation of the discovered OWL-S description of WSs into PDDL elements.
- Solving the transformed problem by invoking external planners in VLEPPO.
- Visualization of the generated plans in PORSCEII.
- Selection of one of the generated plan based on statistical methods and accuracy metrics.

In the remainder of this paper, inside tables we shall use the abbreviation PII-V to denote the PORSCE II and VLEPPO approach.

4.3 Bartalos

Bartalos[26] created an approach for composing SWS in a largescale environment by considering functional properties of WSs (Input, Output,Pre/Post conditions) along with QoS attributes. The basic steps of the framework are 1. Finding WSs that can consume the provided inputs such that their preconditions are satisfied, 2. Selecting WSs that provide requested outputs and postconditions by using a backward chaining strategy, and finally, 3. Multiplecomposite services are created based on produced interconnection between initial and final services by considering optimal the QoS value.

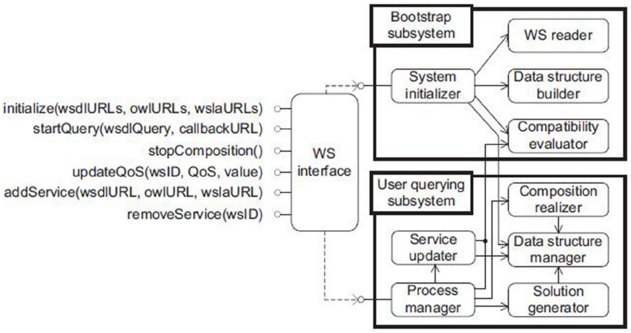


Figure 4. Bartalos Composition Processes Architecture[26].

Figure 4 illustrates the overall composition processes in the Bartalos approach. Its composition process architecture is divided into two main phases: bootstrap and user querying. Bootstrap is the pre-processing stage which is performed before receiving any user request. In this stage all the actual WSs are analysed and linked if there is any relation between them to create a Directed Acyclic Graph (DAG). User querying is done after receiving the goal. The initial and finals WSs are found and then the DAG of WSs found in the pre-processing stage is employed to find a set of suitable composition of services according to QoS attributes.

4.4 Top-k ASC

Top-k ASC (Top-K Automatic Service Composition)[27] is a method which was created for determining best k composition services based on QoS attributes with a large number of WSs. A WS is defined as a three tuple <*I, O,QoS*> where *I, O* denote the semantic concepts of *Input* and *Output*, and *QoS* denotes Quality of Service. QoS itself is defined as an n-tuple$< q_1, q_2, q_{3,} \ldots, q_n >$, where each q_i defines one QoS attribute, such as cost, response time, availability, etc. This framework transforms the WSC problem into a graph searching problem, i.e. each composed service is shown in the form of a DAG. Afterward, the approach, by using a composition algorithm (based on backtracking and depth-first search) can find best k composition services in a parallel way according to the user's request.

The composition procedure of Top-k ASC is depicted in Figure 5. The approach has two phases: run-up and composition. The *run-up* phase, usually done off-line, consists in pre-processing of WSs from a large-scale registry, and then transforming the pre-processed services sets into the rule repository for being efficiently accessible to answer the user's request. The composition phase contains service filtering and parallel composing stages. The service filtering stage fetches rules representation of WSs which are compatible with user request from rule repository and filters out unrelated ones. The parallel composing stage uses the idea of MapReduce[30](a programming modelfor processing parallelizable problems in a large data sets with a huge number of nodes) to find best k composition services in a parallel way, while guaranteeing optimal QoS.

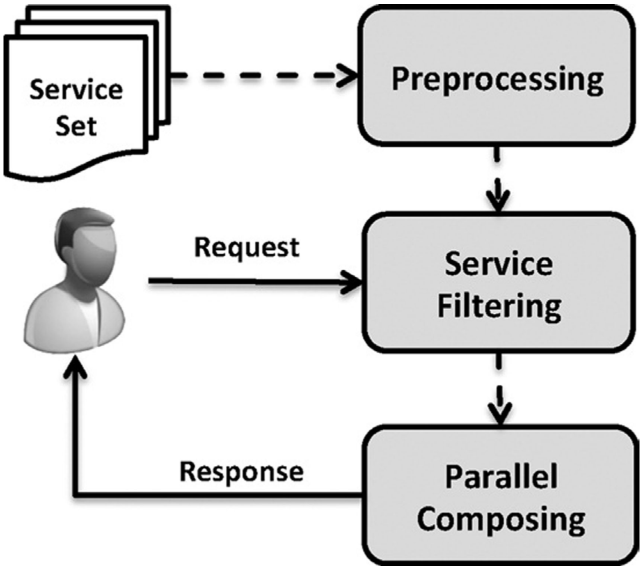


Figure 5. Architecture of Top-k ASC.[27]

Table 5. Evaluation of SWSC approaches based on rubric tables (Part1)

Criterion	Score	Description
1. Automation	4	**DynamiCoS** is an automatic SWSC framework where all the processes of discovery and composition are done automatically.
	3	**PII-V** is a framework which automatically composes SWSs based on AI planning techniques. Furthermore, the most preferable composite service can be selected automatically among the candidate ones based on statistical methods and accuracy metrics without considering any non-functional properties.
	4	**Bartalos** automatically combines SWSs in a large scale environment by considering both functional and non-functional attributes.
	4	**Top-k ASC** automatically determines best k composite services based on QoS attributes.
	4	**Tang et al**. is an automatic WSC framework based on logical interface of Horn clauses and Petri nets.
2. Scalability	3	**DynamiCoS** prototype shows that approach is able to deal with 500 WSs in the registry.
	3	Experimental result shows that **PII-V** is able to handle 1000 WSs.
	4	Experimental result of **Bartalos** demonstrates that it has high scalability, being able to handle around 100,000 WSs.
	4	**Top-k ASC** can handle 20,000 WSs.
	N/S	**Tang et al**. does not give any information about the size of the web service registry.
3. Adaptivity	1	**DynamiCoS** does not support any kind of adaptivity and flexibility in its WSC processes.
	3	In **PII-V**, service replacement component handles problems of service failure or service unavailability by replacing an alternative atomic WS into the composite plan. If it cannot find a suitable alternative WS, it performs the re-planning technique.
	3	**Bartalos** is able to handle three types of changes in WS environment, namely addition/removal of a WS, and change in the QoS of a WS, by designing an algorithm which updates a data structure to handle the dynamic changes in the WS environment.
	1	**Top-k ASC** does not take adaptivity issues into account during the composition processes.
	1	**Tang et al.** is not flexible enough to adapt to any changes that may happen during the composition processes.

Table 6. Evaluation of SWSC approaches based on rubric tables (Part2)

Criterion	Score	Description
4. Dynamicity	4	**DynamiCoS** supports runtime discovery, selection and composition of SWSs.
	4	In **PII-V**, all the processes of discovery, composition and selection of composed services are done at runtime.
	4	**Bartalos** composes SWSs in a large scale environment by considering the QoS attributes at runtime.
	4	**Top-k ASC** performs the processes of creating best k composite services at runtime.
	4	In **Tang et al.** all the composition processes are done dynamically.
5.Heterogeneity	3	**DynamiCoS** framework has two steps for solving the problem of using different WS description languages. First, it specifies an interpreter for each supported description language to extract necessary information of WS such as *I*, *O*, *P*, *E*, *G* and *NF*. Second, it publishes all the extracted WS information into the registry by using the DynamiCos service publication mechanism.
	1	**PII-V** does not consider the problems of incompatibility of WSs in the registry, since it uses the same description language, OWL-S, for all the existing WSs.
	3	In **Bartalos** different types of WS descriptions (WSDL, OWL-S) are parsed by the WS *Reader*. Then the parsed data are processed by the WS processor to build the basic data structure.
	1	**Top-k ASC** assumes that all the WSs use the same description language.
	1	**Tang et al**. does not take the incompatibility of WSs into account since it uses the same semantic description language, SAWSDL, for all existing WSs in the registry.
6. Workflow Pattern	2	**DynamiCoS** uses a graph-based composition algorithm to find a composite service based on user request. A composite service which is represented as a DAG can have sequential and and-split-join constructors.
	2	In **PII-V,** plans which are generated in VLEPPO are based on the sequential and and-split-join control structure.
	2	In **Bartalos,** composition of WSs is presented in DAG with sequential and and-split-join control structure.
	2	Best k composite services in **Top-k ASC** are presented in DAG which can contain sequential and and-split-join structure.
	2	In **Tang et al.,** the represented composite service in Petri net can contain sequential and and-split-join constructors.

Table 8. Summary of the evaluation of SWSC approaches based on rubric tables

Criterion	DynamiCoS	PII-V	Bartalos	Top-k ASC	Tang et al.
Accepted SWS Description Methodology	Top-down	Top-down	Top-down	N/S	Bottom-up
Quality of Service	Cost	N/A	N/S	Response time	Response time, Cost, Availability
Composition Methods	Mathematics-based	AI-planning	AI-planning	Mathematics-based	Model-based
Execution of Composite Services	N/A	N/A		N/A	N/A
Personalization	N/A	N/A	N/A	N/A	N/A

4.5 Tang et al.

Tang et al.[28]presents a framework for composing SWSs based on a logical interface of Horn clauses and Petri nets. In this approach a WS is defined by four-tuple <*I, O, BC, QoS*> where *I, O, BC* and *QoS* stand for the semantic concepts of *Input* and *Output*, set of Behavioural Constrains and Quality of Service respectively. Behavioural Constrains are conditions which ensure correct execution of the WSs. Quality of Service involves attributes such as cost, response time, availability and reliability.

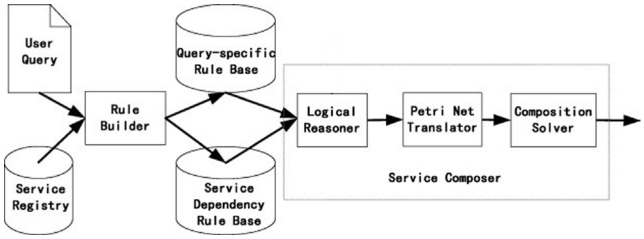


Figure 6. Framework of Tang et al[28].

Figure 6 illustrates SWSC processes of Tang et al.

which involve the following steps:

- Before accepting any request from the user, the Rule Builder component that works based on the hypergraph theory[31] generates dependency rules between the existing SWSs in the registry. These rules have the structure WS_1 WS_2 … WS_k– WS_z which means whenever all of $WS_i (1 \le i \le k)$ have finished their execution, then WS_z can be invoked. Rules are stored in the Service Dependency Rule Base.
- When an end user request comes, rule builder creates Horn logic rules for inputs and outputs of the request. Rules are indicated – WS_p and WS_q – for inputs and outputs respectively, and are stored in the Query Specific Rule Base.
- The Logical Reasoner applies an algorithm based on forward chained deduction for propositional logic (PL-FC-ENTSILS?) on both service dependency and query specific rules. This algorithm determines whether there exist any compositions of services that can satisfy user request. If such a composition exists, it returns a set of Horn clauses rules which are necessary for the composition.
- The Petri Net Translator takes the selected rules and converts them into the Petri net representation, and finally,
- The Composition Solver part generates composite services by using structural analysis techniques, such as T-invariant, of Petri nets.

5. Actual Comparison of Approaches

This section presents the actual comparison and appraisal of the mentioned SWSC approaches based on the RFSWS framework which we presented in Section 3. Tables 5, 6 and 8 depict the evaluation of the aforementioned approaches according to the rubric tables and feature-based scheme respectively.

5.1 Rubric-based Appraisal and Comparison

Tables 5 and 6 illustrate tabular comparison and appraisal of SWSC approaches based on rubric Tables 2 and 3.

A summary of the findings is presented in Table 7 (Assume that all criteria have the same importance). Results show that among the evaluated SWSC approaches, Bartalos[26] is the winner of rubric-based evaluation with the highest total grade of 20. Most of the studied approaches except PORSCE II and VLEPPO obtained highest score in the automation and dynamism criteria. These approaches achieved this score since they carried out all the steps of SWSC processes automatically and at runtime along with considering non-functional properties of WSs beside the functional ones.

In the second criterion (scalability), Bartalos and *Top-k ASC* received the best score among the approaches. They obtained this score since they were able to deal with more than thousands of WSs in the registry. Both of the mentioned approaches pre-process existing WSs in the registry to determine dependencies between WSs and thereby enhance response time and the performance of discovery and composition processes. Bartalos seems better than Top-k ASC and Tang et al. In pre-processing of WSs as it takes different types of WS description languages into account.

All the mentioned approaches obtained the same score in the workflow pattern criterion. Approaches merely support sequential and and-split-join control structures for creating composite services and cannot handle complex patterns such as conditional and iteration.

Lastly, adaptivity guarantees flexibility of approaches against any changes that may happen during the composition processes. Bartalos and PORSCEII and VLEPPO provide methods to handle such changes. Bartalos, comparedto PORSCE II and VLEPPO, is able to adapt itself to more changes, namely addition/removal of a WS, and change in the QoS of a WS.

Table 7. Evaluation of SWSC approaches using the feature-based table

Criterion	Dynami	PII-	Bartalos	Top-k	Tang
	CoS	V		ASC	et al.
Automation	4	3	4	4	4
Scalability	3	3	4	4	N/S
Adaptivity	1	3	3	1	1
Dynamicity	4	4	4	4	4
Heterogeneity	3	1	3	1	1
Workflow Pattern	2	2	2	2	2
Total score	**17**	**16**	**20**	**16**	**12**

5.2 Feature-based Appraisal and Comparison

Table 8 depicts the evaluation of SWSC approaches based on supported features described in Table 4. Unlike the summary evaluation rubric Table 7, it is hard to quantitatively compare features of the approaches, since they employ a variety of methods, tools and languages.

The first feature (accepted SWS description methodology) determines what kinds of semantic description languages are used by the approaches. Bartalos and PORSCE II and VLEPPO use WSs which are semantically described in OWL-S. On the other hand, DynamiCoS which is a service description language neutral framework, allows service providers to use different semantic description languages for describing the WSs. Spatel language is employedin the DynamiCoS prototype for the semantic annotation of WS operations. Furthermore OWL is also used in this prototype for describing ontologies. In contrast, Tang et al. accepts the bottom-up WS description language, SAWSDL.

The second row of Table 8 depicts the QoS aspects of the approaches. Bartalos considers QoS attribute values for each atomic WS, and the best composite service is found based on the optimal aggregated QoS values of each atomic WS which participates in the composition. However it does not clearly define what kinds of QoS attributes are used in this approach. On the other hand, the DynamiCo *S*prototype uses an OWL ontology which defines non-functional properties of WSs, but only cost is used in the prototype. Tang et al. uses the QoS attributes cost, availability and response time for WSs.

Approaches, in order to achieve user's desired goals, combine WSs in variousways. DynamiCoS and Top-k ASC employ different mathematical techniques to compose the WSs. In DynamiCoS composite services are generated based on a CLM matrix via a graphic composition algorithm, whilst in Top-k ASC the WSC problem is transformed into a graph searching problem. Top-k ASC uses a composition algorithm based on the combination of backtracking and depth-first search. In PORSCE II and VLEPPO and Tang et al., similar to Top-k ASC, the WSC problem is transformed into a different formalism. PORSCEII and VLEPPO uses AI-planning method and WSC problem is transform into a planning problem. Then, WSs are composed using classical planning techniques. *Tang et al.* uses a model-based composition method. In the Tang et al. WSC problem is transformed into logical interface problem of Horn clauses and Petrinets, and then a forward chaining algorithm is used to find composite services.

Finally, despite the importance of last two features (execution of composite services and personalization); most of the approaches do not deal with them. Only Bartalos, via Solution Generator supplies composite service in executable BPEL format.

Table 4. Features and sub-features that will be used to evaluate approaches

Feature	Value space
1. Accepted SWS Description Methodology	(a). Top-down (b). Bottom-up
2. Quality of Service	(a). Response time (b). Performance (c). Cost (d). Availability (e). Security (f). Reliability
3. Composition Methods	(a). AI-planning (b). Workflow-based (c). Model-based (d). Mathematics-based (e). Other methods
4. Execution of Composite Services	(a). Self-execution (b).Execution by other standards
5. Personalization	(a). User preference (b). Context awareness

6. Discussion

Below, we discuss the shortcomings and advantages of the studied approaches. Chart 7 depicts graphically the scores of each studied approaches determined by the rubric table evaluation.

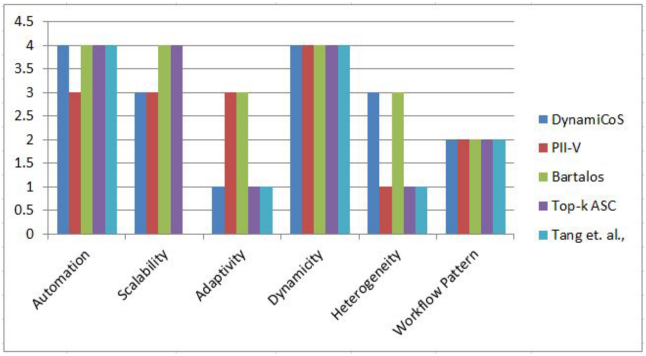


Figure 7. Chart evaluation of SWSC approach based on rubric tables.

Important issues in SWSC are listed as follows:

- User request
- Web services related issues
- Discovery
- Composition
- Adaptivity

The first step of WSC processes is the user making known its request in some intuitive manner, such as text in a formal language, a natural language, graphical / visual way, etc. Since the aim of SWSC approach is the composition of WSs to meet the user's demand, users who are not expert programmers should also be taken into account.

Most of the prior work on WSC (including the five approaches evaluated here) are not capable of receiving the user request in some intuitive manner. Except in PORSCE II and VLEPPO system which graphical / visual ways are used and users can define web service compositions using the visual interface of *VLEPPO*. Because of its importance, recently more attention has been paid to this subject. For example, in[32,37] the authors use mechanisms to map a user request words to basic functionalities of the system by extracting the workflow of the composed service from the user's request.

In the second step of WSC processes, system should be able to deal with the following WS problems:

- Variety of description languages.
- Large number of WSs.

For the first problem, there exist many approaches which have obtained promising results by using either non-systematic methods (as in the case of DynamiCoS), or by using mediators (as in the case of[33]).

For the second problem, pre-processing techniques have been extensively employed in SWSC approaches which are able to handle large number of WSs. Bartalos and Tang et al. are examples of those approaches.

In the discovery part, requested WSs are extracted from the WSs registry to obtain the user's desired goals. Since different approaches employ a variety of methods and languages to select compatible WSs with user request, (e.g Author of [34] used FLORA-2 language), we cannot estimate which approach is better compared to the others. Authors of [35] surveyed different Web service discovery techniques. However, we can say that an approach which can perform both exact matching and partial matching as required by the circumstances and additionally considers non-functional properties as well as functional ones can obtain more promising results than the others.

In the composition part, similar to the discovery part, due to the diversity of methodologies used, it is hard to decide on the "best" methodology. In the evaluated approaches, a WSC problem is represented internally as a graph-based or model-based problem with sequential and and-split-join control structures. It is understood that employing the conditional and iteration patterns along with sequential and and-split-join is a very complicated task. Approach[39] does in fact apply the conditional pattern (if-then-else) along with sequential and and-split-join ones in their composition plan, but in this method the user designs participating WSs interaction manually.

Lastly, adaptivity is a significant issue which has to be taken into account during the composition processes as well. Among the evaluated approaches, Bartalos has the highest rank due to it is flexible and adaptive response against some (not all) changes in WS environment. The mentioned changes are addition / removal of a WS, and change in the QoS of a WS.

7. An Idealized Approach

An idealized approach has to perform all the steps of the WSC processes in an automated manner. Specifically, it should be able to contain the following manner:

- Accept requests both from expert and inexpert users,
- Deal with large number of WSs in the registry (scalability),
- Handle diverse WSs in the registry (heterogeneity),
- Automatically employ partial matching as well as exact matching of requested WSs with the WSs in the registry at runtime,
- Consider both functional and non-functional properties of WSs in the composition processes.
- Involve sequential, and-split-join, conditional and iteration patterns in the control structure of the composite service, and
- Adaptable and respond to significant changes at execution time in volatile environments.

The survey paper[36] reviews WSC approaches based on two composition methods: workflow and AI-planning, and claims that most of the WSC approaches employ AI-planning techniques to compose the WSs.

The authors of[6] present requirements in automated WSC such as dynamicity, automation level, semantic capability, QoS awareness, scalability, correctness, domain independence, partial observation and adaptivity. They compare state-of-the-art approaches proposed until 2010 based on these criteria.

Vardhan et al.[37] introduce a review paper which has a simple classification of WSC approaches, namely static composition, dynamic Composition and semantics based. It proceeds to evaluate and compare WSC approaches using this classification.

Authors of[38] review only dynamic WSC approaches. They derive a reference model along with some requirements for dynamic WSC techniques, namely query analyser, dynamic selection, composition template, verification model, distributed execution, monitoring module, recovery module, QoS certifier, control agent, semantic based and context source. They then analyse the WSC approaches based on these requirements.

In [5] the authors present a WSC life cycle consisting of three phases, namely definition, service selection, and execution. For each phase, a different set of requirements is given. For definition phase, the requirements are expressibility and correctness. The requirements automation and selectability are for the service selection, and finally adaptability, scalability, monitoring and reliability are for execution phase. Then, they compare WSC approaches using the specified requirements.

Our work is unlike the prior works because:

- We determined aspects of SWSC processes that can be evaluated as performance criteria in rubric tables and defined rubric tables for such evaluation.
- We designed a feature-based evaluation scheme for aspects of SWSC processes that could not be natural evaluated using performance criteria.
- We created a novel framework called RFSWS consisting of rubric tables and the feature-based evaluation scheme for the evaluation and comparison of SWSC approaches.
- We used the RFSWS framework to actually evaluate five state-of-the-art SWSC approaches that contain the recent advancements in SWSC technology.
- We proposed an idealized SWSC approach which can be the yardstick against which any new SWSC approaches can be judged.

8. Conclusion

In this work we presented a novel method for the appraisal and comparison of automatic SWSC approaches. Our method comprises two parts: a rubric table, and a table of features appropriate for evaluating SWSC processes. We called our method RFSWS, which stands for Rubric and Feature-based evaluation framework for Semantic Web Services composition approaches.

Our usage of analytic rubric tables in the framework is the first of its kind in the evaluation of SWSC approaches. Aspects of SWSC approaches that could not be assessed meaningfully using rubrics have been delegated to the feature-based evaluation scheme. When used in concert with the feature-based evaluation scheme, the rubric we generated gives a reasonably complete picture of the capabilities, deficiencies, strong and weak points of a SWSC approaches under review.

In the next stage, we identified five recent, prominent SWSC approaches thorough literature search, explained their methodology, and then proceeded to evaluate them using our newly developed framework RFSWS. One of the approaches, Bartalos, obtained the highest score in the rubric based evaluation, and had more desirable features compared to the others.

In discussion part, we presented the shortcomings and strong points of SWSC approaches evaluated under our framework. We also pointed out some strong aspects of several other WS composition approaches to contrast them with the shortcomings of the evaluated ones. Based on our observations, we then proposed the features and characteristics of an idealized SWSC approach.

In conclusion, we can say that unlike all the previous works, we proposed a novel idea in the context of comparison and evaluation of WSC approaches based on rubrics. Our idea not only can be used as is or enhanced by other future researchers for the evaluation of SWSC approaches, but also it can be adapted by researchers working on other subjects to evaluate methodologies and approaches relevant to their area of investigation.

9. References

1. Belqasmi F, Singh J, Melhem Y, Glitho H. SOAP-based vs. RESTful Web Services: A case study for multimedia conferencing. IEEE internet computing. 2012; 16(4):54–63.
2. Wolf K, Stevens E. The role of rubrics in advancing and as¬sessing student learning. The Journal of Elective Teaching. 2007; 7(1):3–14.
3. Sheng Q, Qiao X, Vasilakos A, Szabo C, Bourne S, Xu X. Web services composition: A decade's overview. Information Sciences. 2014. p. 218-38.
4. Baryannis G, Plexousakis D. Automated web service composition: State of the art and research challenges.Institute of computer science information system laboratory; 2010 Oct, 90 p. Tech Report No: ICS-FORTH/TR-409.
5. Bartalos P, Bielikov M. Automatic dynamic web service composition: A survey and problem formalization. Computing and Informatics. 2012; 30(4):793-827.
6. Roman D, Keller U, Lausen H, De Bruijn J, Lara R, Stollberg M, Polleres A, Feier C, Bussler C, Fensel D. Web service modelling ontology. Applied ontology. 2005; 1(1):77–106.
7. Martin D, Burstein M, Hobbs J, Lassila O, McDermott D, McIlraith S, Narayanan S, Paolucci M, Parsia B, Payne T, et al . OWL-S: Semantic mark-up for Web Services. 2015 Aug 12. Available from: www.w3.org/submission/OWL-S

8. Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American. 2001; 284(5):34–43.
9. Gruber T. A translation approach to portable ontology specifications. Knowledge acquisition. 1993; 5(2):199–220.
10. Nitko A, Brookhart S. Educational Assessment of Students. 3rd edition. Prentice Hall; 2001.
11. Moskal B. Scoring rubrics: What, when and how? Practical Asessment, Research & Evaluation [Internet]. 2000 [cited 2015 Aug 20]. Available from: http://www.pareonline.net/getvn.asp?v=7&n=3.
12. RubiStar. 2015 Aug 20. Available from: rubistar.4teachers.org/
13. Kuster U, Konig-Ries B, Klein M, Stern M. DIANE: A matchmaking cantered framework for automated service discovery, composition, binding, and invocation on the web. International Journal of Electronic Commerce. 2007; 12(2):41–68.
14. Steinmetz N, Toma I. WSML language reference [Internet]. 2015 Aug 12. Available from: http://www.wsmo.org/TR/d16/d16.1/v1.0/
15. Akkiraju R, Farrell J, Nagarajan M, Schmidt M, Sheth A, Verma K. WSDL-S: Web service semantics. 2015 Aug 12. Available from: www.w3.org/Submission/WSDL-S/
16. Farrell J, Lausen H. SAWSDL: Semantic annotations for WSDL and XML schema. 2015 Aug 11. Available from: www.w3.org/TR/sawsdl/
17. Lee K, Jeon J, Lee W, Jeong S, Park S. QoS for web services: Requirements and possible approaches [internet]. 2015 Aug 20. Available from: www.w3c.or.kr/kr-office/TR/2003/ws-qos
18. Karimi M, Esfahani SF, Noorafza N. Improving Response Time of Web Service Composition based on QoS Proper¬ties. Indian Journal of Science and Technology. 2015; 8(16).
19. Jordan D, Evdemon J. Web Services Business Process Execution Language Version 2.0. OASIS standard; 2007.
20. 20. Hong J, Suh E, Kim J, Kim S. Context-aware system for proactive personalized service based on context history. Expert Systems with Applications. 2009; 36(4):7448–57.
21. Rong W, Liu K. A survey of context aware web service discovery: from user's perspective. Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE). 2010. p. 15–22.
22. Silva D E, Pires L, Sinderen M. Towards runtime discovery, selection and composition of semantic services. Computer communications. 2011; 34(2):159–168.
23. Hatzi O, Vrakas D, Nikolaidou M, Bassiliades N, Anagnostopoulos D, Vlahavas L. An integrated approach to automated semantic web service composition through planning. IEEE Transactions on Services Computing. 2012; 5(3):319–32.
24. Hatzi O, Vrakas D, Bassiliades N, Anagnostopoulos D, Vla¬havas I. The PPRSCE II framework: using AI planning for automated semantic web service composition. The Knowledge Engineering Review. 2013; 28(2):137–56.
25. Bartalos P. Effective automatic dynamic semantic web service composition. Information Science and Technology. Bulletin ACM Slovakia. 2011; 3(1): 61–72.
26. Deng S, Huang L, Tan W, Wu Z. Top k automatic service composition: A parallel method for large-scale service sets. IEEE Transactions on Automation Science and Engineering. 2014; 11(3): 891–905.
27. Tang X, Jiang Ch, Zhou M. Automatic web service composition based on horn clauses and petri nets. Expert Systems with Applications. 2011; 38(10):13024–31.
28. Lecue F, Leger A. A formal model for semantic web service composition. In: The Semantic Web-ISWC. Springer-Berlin Heidelberg; 2006. p. 385–98.
29. Silva DG, Pires L, Sinderen M. Supporting dynamic service composition at runtime based on end-user requirements. Proceedings of 7th International Conference on Service Oriented Computing; Sweden. 2009.
30. Aeronautiques C, Howe A, Knoblock C, Ram A, McDermott I, et al. PDDL: The planning domain definition language. Yale center for computational vision and control; 1998 Oct. p. 27. Tech Report No: CVC TR-98-003/DCS TR-1165
31. Hatzi O, Meditskos G, Vrakas D, Bassiliades N, Anagnostopoulos D, Vlahavas I. Semantic web service composition using planning and ontology concept relevance. Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology. SIEEE Computer Society; 2009. p. 418–21.
32. Hatzi O, Vrakas D, Bassiliades N, Anagnostopoulos D, Vlahavas I. VLEPPO: A visual language for problem presen-ta-tion. Proceedings of 26th Workshop of the UK Planning and Scheduling Special Interest Group; PlanSIG. 2007. p. 60–6.
33. Hatzi O, Vrakas D, Bassiliades N, Anagnostopoulos D, Vlahavas I. A visual programming system for automated problem solving. Expert Systems with Applications. 2010; 37(6):4611–25.
34. Dean J, Ghemawat S. MapReducie: Simplified data processing on large clusters. Communications of the ACM. 2008; 51(1):107–13.
35. 35. Claude B. Hypergraphs: Combination of Finite Sets. In: North-Holland Mathematical Library; 1989.
36. 36. Sangers J, Frasincar F, Hogenboom F, Chepegin. Semantic web service discovery using natural language processing techniques. Expert Systems with Applications. 2013; 40(11):4660–71.
37. Domingue J, Cabral L, Galizia S, Tanasescu V, Gugliotta A, Norton B, Pedrinaci C. IRS III: A broker-based approach to semantic web services. Web Semantics: Science, Services and Agents on the World Wide Web. 2008; 6(2):109–32.
38. Ghayekhloo S, Bayram Z. Pre-Filtering Strategy to Improve Performance of Semantic Web Service Discovery. Scientific Programming. 2015. Available from: http://dx.doi.org/10.1155/2015/576463
39. Ghayekhloo S, Bayram Z. Pre-Filtering Strategy to Improve Performance of Semantic Web Service Discovery. Scientific Programming. 2015; In Press.

40. Suchithra M, Ramakrishnan M. A Survey on Different Web Service Discovery Techniques. Indian Journal of Science and Technology. 2015; 8(15).
41. Rao J, Su X. A survey of automated web service composition methods. In: Semantic Web Services and Web Process Composition. Springer-Berlin Heidelberg; 2005. p. 43–54.
42. Vardhan V, Basha M, Dhavachelvan P. An overview of web services composition approaches. International Journal of Computer Applications. 2011; 29(8):10–5.
43. Eid M, Alamri A, Saddik E. A reference model for dynamic web service composition systems. International Journal of Web and Grid Services. 2008; 4(2): 149–68.