International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(3): 640-651
The Society of Digital Information and Wireless Communications, 2011 (ISSN: 2220-9085)

Semantic Web Service Composition Approaches: Overview and Limitations

Furkh Zeshan and Radziah Mohamad
Department of Software Engineering, Faculty of Computer Science and Information Systems,
Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia
farrukh05@hotmail.com, radziahm@utm.my

ABSTRACT

Service composition is gaining popularity because the composite service presents the features that an individual service cannot present. There are multiple web services available over the web for different tasks. Semantic web is the advance form of the current web, where all the contents have well defined meanings, due to this nature; semantic web enables the automated processing of web contents by machines. At run time, the composition of these services based on the requester's functional and non-functional requirements is a difficult task due to the heterogeneous nature of results of the services. This paper introduced some requirements that when fulfilled, a successful composition process can be achieved. In order to find the best approach, various composition approaches on these requirements were evaluated. Suggestions were provided on what approach can be used in which scenario in order to gain the best results.

KEYWORDS

Web Services, Semantic Web Services, Composition Approaches.

1 INTRODUCTION

Web services are well defined, self described and reusable software components that can be used over the web using the most silent and stable technologies such as Simple Object Access Protocol (SOAP) as a communication framework, Web Service Definition Language (WSDL) and Universal Description, Discovery and Integration (UDDI) that provides a mechanism to clients to find services [4], [32]. A web service is a set of related functions that can be accessed through programming over the web [14]. The key feature of the web services is that they are loosely coupled, allows ad hoc and dynamic binding and are reusable software components. Web services can be divided into three categories and three entities. The categories are publish, find and bind, while the entities are service requester, service provider and the registry. The roll of facilitator of service outsourcing is one of the most significant aspect of the web that can reduce the overhead of companies and flourish the business [17], [33]. WSDL is the emerging language for describing the present web service technology [17] and presents the syntactic description of the web services. It only present the structure of the data sent and received through the web, but is unable to present the meaning of the data. This makes the automated web service composition difficult as composition, semantic description and execution of web services is necessary for automatic discovery. Existing techniques for web services provides only the syntactic description which as a result, makes it difficult for requester and provider to interpret the meaning of the input and output. Semantic web services are the combination of web services and the semantic web. In the domain of semantic web, Web Ontology Language for Services (OWL-S) [35] and Web Service Modeling Ontology (WSMO) [36] are two prominent techniques used for service composition. Semantic web services are the extension of the existing web services where the information is represented in a well-defined way [11]. Large amount of data over the web is understandable only by the humans and the custom software [11], [15], [16]. The target goal of semantic web is the medium where the data could be shared easily and processed automatically. The key technology for such concept is the web services [12], [13].

Normal web uses HTML for presenting information, issues, images and active links which makes it is easy to understand for human beings but difficult for the machines to understand the presented information. But, semantic web which is the advance form of the normal web and refers to the ontology languages, development frameworks and development tools; it uses semantic annotation (web pages with structured data to facilitate the software / intelligent agents to process the data [39]) for describing some of the parts of the web and the meaning of the message of the web services. With the help of annotations semantic web services infer inherent properties to identify services that meet to the requesters demand during the discovery process.

Semantic web services are used for combining data and services from different sources without losing their meaning. Through the discovery and assembly of web services, semantic web services provide the value-added services to complete the domain tasks [34]. Web services can be combined to provide the unified service with some additional extra values.

To find out the most relevant service among functionally similar that meet the requirements of users is the key issue in the web service discovery [40], [41]; however there is a need to define a set of well defined quality of services criteria and user preferences [42]. Especially the users of business to business application would like to discover the services which meet their non-functional requirements [43]. Typically web services are defined in their functional parameters i.e. input, output parameters, whereas QoS parameters are used to define the behavior of the service. Moreover QoS resolves the issue of best service among the functional similar services by ranking and selection based on non-functional requirements. Hence QoS can be used as main factor for ranking the web services. In selection process the service with high QoS value will be selected first and this step is performed after the functional matching step. Four steps are necessary for successful composition of services [37]:

i. Data and control flow model among entities should be created.

ii. For process activities, the services that discover the matched service with the criteria form the service registry should be bounded.

iii. In order to be available to the clients, the composite services should be published in UDDI.

iv. Control and data flow should be managed during the composite service invocation. Although semantic web is gaining popularity, the supporting technologies are still far from the final

product, making it an emerging field of research [3].
Since the last decade, considerable work has been done on semantic web services composition [2], [22], [23], [24], [25], [26], [27], [38] but more research is required to address the issue of heterogeneity in automatic (minimal user intervention) web services composition as web services provided by different companies (having their own business rules) provide heterogeneous results. In this paper, we compared and categorized such approaches into two categories; semantic web composition approaches with Quality of Services (QoS) support and semantic web composition approaches without QoS support. The objective of this paper is to identify the best approach that can be used for semantic web services composition. The rest of the paper is organized as follows. Section 2 presents an overview of web service composition approaches. In Section 3, comparative evaluation criteria are discussed. Finally, we present the comparative evaluation remarks in section 4 and the conclusion in section 5.

2 SEMANTIC WEB SERVICE COMPOSITION APPROACHES

In semantic web services composition, machines can automatically select, integrate and invoke different web services in-order to achieve the user specified task according to the user constraints. Automatic composition of web services involves both routine and complex tasks to be performed on the web without user involvement and hence saving time for composing and integration of information. Semantic web service composition is a widely-studied field since the last decade and different composition techniques have been identified [5], [6], [7], [8], [9], [10]. Most of the work done on semantic web service composition approaches can be classified into two categories; Semantic web composition approaches with QoS support and Semantic web composition approaches without QoS support. Short descriptions of these approaches are given below.

2.1 Terms and Definitions

In this section we have presented the definitions for common acronyms used in this paper.

2.1.1 Universal Description Discovery and Integration (UDDI)

UDDI contain description of web services and mechanism for requester to get access to the services publication and description. It contains data with combination of white, green and yellow pages. White pages contain information like company name, address, contact information etc; yellow pages contain information like business category, business type etc; while green pages presents that which kind of services the business offer.

2.1.2 Simple Object Access Protocol (SOAP)

SOAP [44] is XML based communication protocol to access the web services over the network using transport protocol like HTTP. SOAP is a language and platform independent.

2.1.3 Web Service Description Language (WSDL)

WSDL [45] specifies the mechanisms to access a Web service over the network. WSDL files are stored in UDDI as a registry so that a web service requester locates them. It is an XML based language which provides syntactic description of web services.

2.1.4 Web Ontology Language for Services (OWL-S)

OWL-S [46] is a high level language (XML based) used for describing web services properties. It consists of three parts, service profile, process model and grounding. Service profile includes general information and is used to describe what the service will do; process model describes how the service will perform it"s functionally while grounding describe links with industry standards. Its main goal is to enable users to automatically discover, invoke, compose and execute web services under certain conditions [47].

2.1.5 Web Service Modeling Ontology (WSMO)

WSMO [36] is used for describing the semantics of web services [48]. It consists of four parts goals, ontologies, mediators, and Web services. Goal defines the user desires. Ontologies define formal semantics for the terms describing data to achieve interoperability among other WSMO elements. Mediator is used to handle interoperability problems between different WSMO elements while web service part describes the functional behavior, precondition, post condition, control flow etc. of an existing deployed service.

2.1.6 Quality of Services (QoS)

Usually Quality of Service (QoS) defines the non-functional requirements of a service such as response time, price, availability and so on . QoS properties can be divided into two subcategories measurable (throughput, response time, and latency etc.) and non-measureable (reputation and security etc.). Considering QoS aspects when deciding which services to include in a service composition schema is important when functional requirements are satisfied by more than one service.

2.1.7 Semantic Web Services

Semantic Web services consist of two terms, the semantic web and the web services. Semantic web means adding machine processable semantics to data. With the help of well defined semantics, machines can understand the information and process it on behalf of human user whereas web services aim at global infrastructure for distributed computation and for integration of various applications and automation of business processes. Semantic web services promises for providing the most suitable service to the user.

2.1.8 Web Service Composition

Composing existing Web services to deliver new functionality is a requirement in many business domains. Service composition extends the notion of service discovery by enabling automatic composition of services to meet the requirements of a given a high level task description. With semantic markup of services, the information necessary to select and compose services is available via the semantic descriptions of the requirements and capabilities of services. This enables automatic composition of web services.

2.1.9 Causal Links

In functional composition approach of web services where composition consists of partial order and web services are semantically linked is considered as a composition of casual links whereas the Causal Link matrices (formal model) are used to find the best composition [49].

2.1.10 Software Agent

Software agent is the piece of software capable to perform the specified tasks on behalf of user. Software agents are mostly used in artificially intelligent applications.

2.2 Non-QoS-Based Approaches

In this section we have presented introduction of four different non-QoS based approaches.

2.2.1 Lecue et al. Approach [18]

For effective retrieval of composed service author has integrated DL (Description Logic) reasoning and Situation Calculus along with the extended version of Golog (logic programming language) which is sclGolog. For integration they have used DL reasoning between the input, output parameters (DL based description) of service in order to infer causal links (semantic matchmaking between parameters) along with the situation calculus. The extended Golog (offline interpreter suitable for reasoning about conditional parameters) interpreter can compute conditional web service compositions and is able to elaborate a strategy for automated branching by means of causal links and laws.

2.2.2 Rao et al. Approach [19]

Rao presents a mixed initiative framework for semantic web service discovery and composition allowing user involvement over many key decisions by suggesting and identifying the inconsistencies. Users can decide that how much to delegate to supporting functionality. The composition engine combines Web Ontology Language (OWL) ontologies with Jess with a planning functionality based on the GraphPlan algorithm. GraphPlan provides reach-ability analysis to determine whether a given state can be reached from another state and disjunctive refinement. Planning is used to propose composition schemas to the user. According to the authors, this is the most realistic approach.

2.2.3 Sohrabi et al. Approach [20]

Sohrabi proposes the introduction of preferences to planning with Golog (the agent programming language). User preferences and the web service descriptions are expressed using a first-order language and are used in a modified version of a Golog interpreter. Situation calculus and first order logic are used to describe the functional and nonfunctional properties of our Web services. For translating the semantics of the OWL-S author has used the situation calculus and the composition system uses the user preferences to generate preferred solutions. Evaluation results illustrate the effectiveness of introducing preferences to find optimal compositions.

2.2.4 Aydin et al. Approach [21]

Aydin proposes a framework in which event calculus (general logic programming treatment of time and change [6]) is used for solving the web service composition problems. With the help of abduction theorem (abduction is the inverse of deduction and used over the event calculus axioms in-order to get partially ordered sets of events) and event calculus (obtained from domain), proper composition plan which corresponds to the user specific composition of web services can be generated. An abduction theorem generates a series of events as well as a set of temporal ordering predicates, giving partial ordering of events which are more suitable for web service composition.

2.3 QoS-Based Approaches

In this section we have presented introduction of different QoS based approaches.

2.3.1 Lecue et al. Approach.

The author in [28] defines five different types of causal links between the input and output of services: Exact (when the input and output parameter are conceptually equivalent), PlugIn (output parameters are sub concepts of the input parameters), Subsume (output is a super concept of the input), Intersection (if the intersection of the output and input are satisfiable), Disjoint (the input and output are incompatible). Causal Link Matrix (CLM) can be developed from a set of services which only considers the functional properties. The author extends CLM to CLM+ to support the non-functional properties. After receiving a request, the most suitable service is discovered from the service repository, and then the semantic connections between services are stored on CLM+ which can be used to compute the composition that represents the possible service composition that matches the service request.

2.3.2 Ardagna et al. Approach.

Ardagna processes with Adaptive Web Services (PAWS) [29] deploy the annotated Business Process Execution Language (BPEL) process with global and local constraints that refer to the QoS aspects. Service Level Agreement (SLA) is used to express the constraints. Through SLA negotiations, advance service retrieval module finds the best service that has the required interface and does not violate the constraints for each task in the created process. If no service interface matches the requirement, then the mediator resolves the service interface descriptions. Multiple candidate services are selected for each task and for each process; only one candidate service is executed in the BPEL engine. PAWS also allow the faulty services to be replaced with other candidate services and the recovery actions to undo the results of the faulty services.

2.3.3 Roy and Michael Approach.

In [30], the authors utilize the semantic web service languages with the model driven methodology to build composite web services. The methodology guides the developers in four phases. At the end of the first phase, an abstract composite model is obtained, containing all the necessary information that can be used for service discovery and selection. In the second phase, suitable web services are handled where the discovery process depends on the semantic descriptions matchmaking. In third phase, a finalized concrete composite model is obtained (for new composite web service, data transformation step is introduced between the services to handle the mismatch between the output of one service and the input of the other service). In the fourth and last phase, different descriptions of the concrete composition model are used for the composed service.

2.3.4 Zeng et al. Approach.

In [31], a goal-directed service composition and optimization framework is presented. The goal directed problem takes three inputs, the domain specific composition rules, the description of the business objectives and the description of the business assumptions. The initial step is called Backward Chaining where the composition rule creates a chain backwards from the business objectives until the initial state is reached and there are no more rules. The second step is called Forward Chaining, where in this step; some additional services are added to the composition schema produced during the first step to complete it, which may be required by the results of some tasks. The contribution of the previous steps contributes to the control flow aspects of the composition. In the final step, called Dataflow Inference, data flow is added to the composition schema.

3 COMPARATIVE EVALUATIONS

While reviewing different semantic web services composition surveys and approaches [1], [2], [22], [23], [24], [25], [26], [27], [38], we have identified six different kinds of requirements (dynamic aspect, adaptability, domain independence, correctness, scalability and non-determinism), which is our contribution that have large impact on automatic web service composition. These are the minimum requirements that must be satisfied in order to make the composition process successful. The criteria are used to check how far the existing approaches support web service composition. The basic ideas of these requirements are as given below.

3.1 Dynamic Aspect

Dynamic aspect ensures that after a composition schema is designed, it will remain consistent and will be executable for long time.

3.2 Adaptability

Adaptation is the process of modifying service-based applications according to the new requirements of the changed environment as specified in the predefined adaptation strategies.

3.3 Domain Independence

The ability of composition approaches to be applicable on different domains in order to solve the problems.

3.4 Correctness

Correctness is the process of checking to ensure that certain properties of the composition maintained.

3.5 Scalability

Scalability is the capacity to increase the computing power to process more transactions in a given period.

3.6 Non-Determinism

Non-determinism may increase the number of composition schemas for a request and must be considered while designing.

4 COMPARATIVE EVALUATION REMARKS

In this section, we have compared the semantic web composition approaches (another contribution) based on the identified requirements (in section 3). We tried to find out their limitations as well as the best category of approaches that satisfies the criteria as completely as possible. Table 1 summarizes the findings.
The limitations of these approaches are given below.

4.1 Dynamic Aspect

Web services composition should be able to adopt changes that may occur in partner's application at any time [51], [52], [53], so, the participants should respond such changes immediately. Moreover, in the competitive environment, business should team up with the best available service among the candidate services at any time to make a short term relationship and end up such relationship when there is no profit in staying together. An ideal situation of web services composition is, where, service composition process adapts environment changes without or the minimal user involvement. Dynamic aspect is the most ignored area by the non-QoS-based approaches and is mostly covered by the QoS-based approaches.

4.2 Adaptability

Adaptability refers to the ability of the application to adapt changes instantly that may occur in service operations signature, message and access etc. In a highly dynamic environment where services are continuously provided, withdrawal and contents are updated by service providers; in such environment applications should be able to adapt such changes in order to take advantage of the new business opportunities. Currently the process of dealing such changes is manual [54], more research is required for automatic detection and handling these changes.

4.3 Domain Independence

Composition approaches should be generic enough to be applicable to different domains to be able to address different sets of problems. This requirement is covered by most of the approaches.

4.4 Correctness

Correctness is the guarantee to provide a certain kind of output under certain set of input and conditions. In this context, only two approaches; one which is the QoS-based approach in [31] and the other one, the non-QoS-based approach in [19], fulfill the requirements. The other remaining approaches fail to meet this requirement.

4.5 Scalability

Scalability is the capacity to increase the computing power to process more

Table 1: Web service composition approaches.

Approaches \ Parameters	Dynamic Aspect	Adaptability	Domain Independence	Correctness	Scalability	Non-Determinism	Semantic Capability	QoS Awareness
Ardagna et al. Approach [29].	√	√	√			√	√	√
Roy and Michael Approach [30].	√						√	√
Lecue et al. Approach [28].	√		√				√	√
Zeng et al. Approach [31].				√			√	√
Aydin et al. Approach [21].			√				√	
Sohrabi et al. Approach [20]			√		√		√	
Rao et al. Approach [19].				√			√	
Lecue et al. Approach [18].			√				√	

transactions in a given period. Even if this requirement fulfilled, not ensures that a composition approach for a given set of services working effectively will also work good for a different set of services. The performance of composition approaches should also be tested and the parts that affect the performance must be tuned to ensure the maximum scalability for the given performance. Scalability is related to performance. Sohrabi et al. [20] is the only approach that fulfills this requirement.

4.6 Non-Determinism

Refers to a case where for a given request, different composition schemas may be obtained. Non-determinism should be considered while designing a composition approach. Only the Ardagna et al. Approach in [29] from the QoS category fulfils this requirement, the other approaches ignore this very important requirement.

Table 1 shows that lack of scalability, non-determinism and adaptability are the major drawbacks of all the approaches. No approach without QoS support satisfies the requirements of dynamic aspect, non-determinism and adaptability, while no approach with QoS satisfies the requirements of scalability. Most of the approaches with QoS support are focused on the area of dynamic aspect while most of the approaches without QoS support focus on domain independence. A research that could complement the limitations of each approach categories is hence very much needed.

Table 1 show that the Ardagna et al. Approach [29] is the best approach among the rest because most of the parameters are satisfied by it. Semantic web approaches with QoS support are much better compared to the approaches without QoS support because it fulfills the maximum requirements that were identified in order to achieve successful composition of web services.

5 CONCLUSIONS

An overview of state of the art research in semantic web composition approaches was discussed in this paper. The existing works are classified into two categories; approaches with QoS support and approaches without QoS support. Six evaluation criteria were proposed for the purpose of systematically comparing the two approach categories to semantic web services composition; scalability, non-determinism, dynamic aspect, adaptability, domain independence, correctness, semantic capability and QoS awareness. Nevertheless, we do not claim that this classification is exhaustive. In each category, we give the introduction and comparison of selected approaches. This paper identified that semantic web composition approaches with QoS support are much better compared to the approaches without QoS support. The problems of the approaches without QoS support are the satisfaction of dynamic aspect, non-determinism and adaptability. On the other hand, the approaches with QoS support suffer from the scalability problem. An approach that complements each of the categories is therefore needed to achieve successful web services composition.

6 ACKNOWLEDGEMENTS

We would like to thank Malaysia Ministry of Higher Education for sponsoring the research through the grant with vote number 78600 and Universiti Teknologi Malaysia for providing the facilities and support for the research.

7 REFERENCES

1. Ran, S.: A model for web services discovery with QoS. ACM SIGecom Exchanges 4(1), 1–10 (2003)
2. Koehler, J., Srivastava, B.: Web service composition: Current solutions and open problems. In: Presented at the Proceedings of the ICAPS 2003 Workshop on Planning for Web Services, pp. 28–35 (2003)
3. Cabral, L., Domingue, J., Motta, E., Payne, T.R., Hakimpour, F.: Approaches to Semantic Web Services: an Overview and Comparisons. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 225–239. Springer, Heidelberg (2004)
4. http://uddi.org/pubs/uddi-v3.00-published-20020719.htm
5. Benatallah, B., Dumas, M., Sheng, Q.Z., Ngu, A.H.H.: Declarative composition and peerto-peer provisioning of dynamic Web services. In: Proceedings of the 18th International Conference on Data Engineering 2002, pp. 297–308 (2002)
6. Benatallah, B., Medjahed, B., Bouguettaya, A., Elmagarmid, A., Beard, J.: Composing and Maintaining Web-based Virtual Enterprises. In: Workshop on Technologies for E-Services (2000)
7. Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., Shan, M.-C.: Adaptive and Dynamic Service Composition in eFlow. In: Wangler, B., Bergman, L.D. (eds.) CAiSE 2000. LNCS, vol. 1789, pp. 13–31. Springer, Heidelberg (2000)
8. Lazcano, A., Alonso, G., Schuldt, H., Schuler, C.: The WISE approach to electronic commerce. International Journal of Computer Systems Science & Engineering (2000)
9. Muth, P., Wodtke, D., Weissenfels, J., Dittrich, A.K., Weikum, G.: From Centralized Workflow Specification to Distributed Workflow Execution. J. Intell. Inf. Syst. 10, 159–184 (1998)
10. Schuster, H., Georgakopoulos, D., Cichocki, A., Baker, D.: Modeling and Composing Service-Based and Reference Process-Based Multi-enterprise Processes. In: Wangler, B., Bergman, L.D. (eds.) CAiSE 2000. LNCS, vol. 1789, pp. 247–263. Springer, Heidelberg (2000)
11. Sandro Hawke, I.H., Prud‟hommeaux, E., Swick, R.: W3C Semantic Web Activity (2010)
12. Berners-Lee, T.: Services and Semantics: Web Architecture, W3C (April 2004)
13. Bussler, C., Fensel, D., Maedche, A.: A conceptual architecture for semantic web enabled web services. SIGMOD Rec. 31:24-29 (2002)
14. Tsur, S., Abiteboul, S., Agrawal, R., Dayal, U., Klein, J., Weikum, G.: Are Web Services the Next Revolution in e-Commerce? (Panel). In: Proceedings of the 27th International Conference on Very Large Data Bases, pp. 614–617. Morgan

Kaufmann Publishers Inc., San Francisco (2001)
15. Berners-Lee, T.: Services and Semantics: Web Architecture, W3C (April 2004)
16. Berners-Lee, T., et al.: The Semantic Web. Scientific American (May 2001)
17. Medjahed, B., Bouguettaya, A., Elmagarmid, A.: Composing Web services on the Semantic Web. The VLDB Journal 12, 333–351 (2003)
18. F., Lecue, A., Leger, Delteil, A.: DL Reasoning and AI Planning for Web Service Composition. In: Web Intelligence, pp. 445–453. IEEE, Los Alamitos (2008)
19. Rao, J., Dimitrov, D., Hofmann, P., Sadeh, N.M.: A mixed initiative approach to semantic web service discovery and composition: Sap‟s guided procedures framework. In: ICWS, pp. 401–410. IEEE Computer Society Press, Los Alamitos (2006)
20. Sohrabi, S., Prokoshyna, N., McIlraith, S.A.: Web Service Composition via the Customization of Golog Programs with User Preferences. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Conceptual Modeling: Foundations and Applications. LNCS, vol. 5600, pp. 319–334. Springer, Heidelberg (2009)
21. Aydın, O., Kesim Cicekli, N., Cicekli, I.: Automated web services composition with the event calculus. In: Artikis, A., O‟Hare, G.M.P., Stathis, K., Vouros, G.A. (eds.) ESAW 2007. LNCS (LNAI), vol. 4995, pp. 142–157. Springer, Heidelberg (2008)
22. Dustdar, S., Schreiner, W.: A survey on web services composition. Int. J. Web Grid Serv. 1, 1–30 (2005)
23. Rao, J., Su, X.: A Survey of Automated Web Service Composition Methods. In: Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, SWSWPC (2004)
24. Marconi, A., Pistore, M.: Synthesis and Composition of Web Services. In: Marco, B., Luca, P., Gianluigi, Z. (eds.) Formal Methods for Web Services, pp. 89–157. Springer, Heidelberg (2009)
25. Küster, U., Stern, M., König-Ries, B.: A classification of issues and approaches in automatic service composition. In: International Workshop WESC 2005 (2005)
26. Hull, R., Su, J.: Tools for composite web services: a short overview. SIGMOD Rec. 34, 86–95 (2005)
27. Milanovic, N., Malek, M.: Current solutions for Web service composition. In: Internet Computing, vol. 8, pp. 51–59. IEEE, Los Alamitos (2004)
28. Lécué, F., Silva, E., Pires, L.F.: A Framework for Dynamic Web Services Composition. In: Gschwind, T., Pautasso, C. (eds.) Emerging Web Services Technology, Birkhäuser Basel, vol. II, pp. 59–75 (2008)
29. Ardagna, D., Comuzzi, M., Mussi, E., Pernici, B., Plebani, P.: PAWS: A Framework for Executing Adaptive Web-Service Processes. In: Software, vol. 24, pp. 39–46. IEEE, Los Alamitos (2007)
30. Groenmo, R., Jaeger, M.C.: Model-driven semantic Web service composition. In: 12th Asia-Pacific Software Engineering Conference (APSEC 2005), p. 8 (2005)
31. Zeng, L., Ngu, A., Benatallah, B., Podorozhny, R., Lei, H.: Dynamic composition and optimization of Web services. Distributed and Parallel Databases 24, 45–72 (2008)
32. Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., Weerawarana, S.: Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. In: Internet Computing, vol. 6, pp. 86–93. IEEE, Los Alamitos (2002)
33. Tsur, S., et al.: Are Web Services the Next Revolution in e-Commerce? (Panel). In: Presented at the Proceedings of the 27th International Conference on Very Large Data Bases (2001)
34. Arroyo, S., et al.: Semantic aspects of web services. In: Singh, M.P. (ed.) The Practical Handbook of Internet Computing, pp. 311–317. Chapman & Hall/CRC, Boca Raton (2005)
35. Ankolekar, A., et al.: Web Service Description for the Semantic Web. In: Presented at the Proceedings of the Semantic Web Conference (2002)
36. WSMO, Web Service Modeling Ontology–Standard (2005), http://www.wsmo.org/2004/d2/
37. Sivasubramanian, S.P., Ilavarasan, E., Vadivelou, G.: Dynamic Web Service Composition: Challenges and techniques. In: International Conference on Intelligent Agent & Multi-Agent Systems (IAMA 2009), pp. 1–8 (2009)
38. Agarwal, V., Chafle, G., Mittal, S., Srivastava, B.: Understanding approaches for web service composition and execution. In: Proceedings of the 1st Bangalore Annual

Compute Conference, pp. 1–8. ACM, Bangalore (2008)
39. Fensel, D., Lassila, O., van Harmelen, F., Horrocks, I., Hendler, J. and McGuinness, J. (2000). “The Semantic Web and its Languages”, IEEE Intelligent Systems, Nov/Dec 2000, pp. 67-73.
40. J. Hu, C. Guo, H. Wang, P. Zou. Quality driven Web services selection. Proceedings of the 2005 IEEE International Conference on e-Business Engineering, ICEBE‟05, Beijing, China, 2005: 681-688.
41. Y. J. Seo, H. Y. Jeong, Y. J. Song. Best Web service selection based on the decision making between QoS criteria of service Second International Conference on Embedded Software and Systems, ICESS 2005, Xi'an, China, 2005: 408-419.
42. M. A. Serhani, R. Dssouli, A. Hafid, H. Sahraoui. A QoS broker based architecture for efficient web services selection. IEEE International Conference on Web Services, ICWS 05, Florida, USA, 2005:113-120.
43. J. H. Rao, X. M. Su. A survey of automated Web service composition methods. Lecture Notes in Computer Science, 2005, 3387: 43-54.
44. http://www.w3.org/TR/SOAP
45. W3C: Web Services Description Language (WSDL). http://www.w3.org/TR/wsdl (2003)
46. D. Calvanese, G. Giacomo and M. Lenzerini. Ontology of integration and integration of ontologies Description Logic Workshop 2001, 10-19.
47. D. Martin, OWL-S: Semantic Markup for Web Services, http://www.daml.org/services/owls/1.0/owl-s.html, November 2003.
48. S. Arroyo and M. Stollberg ; „‟WSMO Primer““. WSMO Deliverable D3.1, DERIWorking Draft. Technical report, WSMO, 2004. http://www.wsmo.org/2004/d3/d3.1/.
49. Freddy Le´cue´, Eduardo Silva and Lu´ıs Ferreira Pires; “A Framework for Dynamic Web Services Composition,” Whitestein Series in Software Agent Technologies, 59–75, Birkh¨ auser Verlag Basel/Switzerland 2008
50. Kowalski, R.A., Sergot, M.J.: A Logic-Based Calculus of Events. New Generation Computing 4(1), 67–95 (1986)
51. B. Benatallah, M. Dumas, M. Sheng, and A. H. H. Ngu. Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services. In Proceddings of the IEEE International Conference on Data Engineering, pages 297-308, San Jose, California, USA, February 2002.
52. F. Casati, S. Ilnicki, L.-J. Jin, V. Krishnamoorthy, and M.-C. Shan. eFlow: a Platform for Developing and Managing Composite e-Services. Technical Report HPL-2000-36, HP Laboratoris, Palo Alto, California, USA, 2000.
53. M. Shen, B. Benatallah, M. Dumas, and E. O.-Y. Mak. SELF-SERV: A Platform for Rapid Composition of Web Services in a Peer-to-Peer Environment. In Proceedings of the International Conference on Very Large Databases, pages 1051-1054, Hong Kong, China, August 2002.
54. Book: Service Composition for the Semantic Web, Springer New York Dordrecht Heidelberg London, ISBN 978-1-4419-8464-7, LLC 2011.