Volume 3, Issue 11, November 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Semantics in Web Service: Annotation, Discovery and Composition

Hemant N Yadav*
Student
Department of CSE and IT
L.D.Engineering College
Gujarat Technological University, India

Harikrushna B. Jethva
Associate Professor
Department of Computer Engg.
L.D.Engineering College
Gujarat Technological University, India

Abstract— the Goals of Automation in software development cannot be fulfilled by web services alone. Current web service standard Technologies are lacking in specifying the meaning that can be understood by machine. So the current web service Technologies, when augmented by semantics for each stage of Service Lifecycle, will be able to incorporate Automation by dynamic Discovery and composition of multiple services for Building a complete Process. We here shows the current Technologies in semantic web for annotating Web service for making it Meaningful for machine and can be discoverable by the semantic annotations. As well as Technologies that can be used for composition and Orchestration of Multiple Web services in to fulfil user Goal are also studied and compared.

Keywords—Semantic Web, Web Service, Annotation, Discovery, Composition, Ontology

I. INTRODUCTION

Web services are nowadays a popular technology used in practice when diverse software system integration is in demand. One of the intensively researched areas is the study how Web services can be used to dynamically create functionality, based on the actual requirements. It means whether web services be combined automatically to fulfil the user Goal. Automation needs web services to be understood by machine. To facilitate this kind of Web service utilization, additional metadata depicting the functionality of single services is required. These metadata are provided in a form of semantic annotations. The basic idea is that multiple Web services can be combined together, to form a composite service, supplying more complex needs and functionalities. The particular composition is realized automatically, on the fly, based on the actual goal. This basic Idea needs web services to be discovered automatically by machines and also get combined automatically based on user goal specification [1].

The Semantic Web is an extension of the current web in which information is given well defined meaning, better enabling computers and people to work in co-operation [2]. Inspired by the desire of exchanging machine-processable information, the idea of Semantic Web has been proposed to extend the current Web infrastructure to represent data of well-defined meaning (Berners-Lee, et al. 2001). The Semantic Web initiative is to study the feasible approaches of introducing metadata to describe meanings of Web resources residing at the decentralized Internet. The semantic descriptions are aimed to be interpreted by programs without misinterpretation of available data.

Web Services offer a promising approach to accomplish a loose coupling of processes across organizational boundaries. Web Services technologies present specifications that cover the details required for an automated interoperation among client agents and services on the Web, with a minimum interference of human agents. A Web Service may provide any of the following or their combinations:

- Static information, e.g. retrieving geographic or statistical data;
- Digital processes, e.g. unit conversion or currency exchange; or
- Actual services with concrete effects, e.g. booking a flight or selling a book.

On the other hand, Semantic Web offers computer interpretable semantic knowledge to facilitate a smarter selection of services and assists combining them to build composite services or applications. Such objectives can be achieved by describing the capabilities of a service using semantic descriptions. Programs on the Web will be able to find each other (other Web Services) by matching their requirements with the capabilities of available services. Semantic Web technologies can be applied to describe provided capabilities and/or desired requirements of a service. [3]

II. SEMANTIC ANNOTATION OF WEB SERVICE

Web Services are defined as a "Software System Designed to support interoperable machine-to-machine interaction over network." Web Services are described by WSDL Documents and interact with each other by SOAP messages using HTTP as a Transport Protocol (W3C, 2004a) [4]. The benefit of describing Web services in WSDL format is that WSDL is machine-readable, namely it can be parsed automatically. However, a major limitation of Web services is that service discovery and composition still require manual effort, which becomes a serious burden as the number of Web services grows and the diversity increases [5]. To facilitate the higher automation of service discovery and composition, Web services are advanced the augmentation of Web services descriptions through semantic annotation.

Semantic annotations of web services have been pro-posed as a (partial) solution to web service discovery and composition. These annotations relate the various service elements (i.e. operations, inputs and outputs) to concepts in ontology describing their semantics, form and role. Such semantic annotations can be used by tools to e.g., assist in service discovery and composition [6]. Many solutions have been proposed on How to annotate Web Services? Web Service Annotations needs to Identify Elements of Web Service to be annotated and Specification of the Annotation Languages. Several languages for semantic annotation of Web services are proposed. The most known are OWL-S [7], WSML [8], WSDL-S [9], SAWSDL [10], WSMO-Lite [11]. The deference between them is in terms of the complexity and expressiveness of the WSDL elements. Every language binds the elements of Web services to domain terms. The Main Focus is on the Input and Outputs, the pre-conditions (The Conditions under which the web service to be invoked), and the post Conditions (the conditions which should be satisfied after its execution) [1].The Following Table Summarizes Languages and their Properties.

OWL-S[7]	*Proposed in 2004* *It Includes Service to be Semantically described by* • *Service Profile* *It describes what the service will do? i.e. Input, Outputs, Preconditions, Post Conditions. It is the most important Part of OWL-S service definition.* *It relates to UDDI in Web Service Technology.* *Used for Service Selection and Service Discovery.* • *Service Model* *How it works? Part of OWL-S service is defined by Service Model. It is used to specify Capability Description (Functional Properties like input, output, precondition, and results – IOPS for Process) and Service description (Non-functional Properties like QoS, Security Policy, and Domain Specific Characteristics etc).* *Its Relates to Orchestration and Choreography in Web Service Technology* • *Service Grounding* *It describes, How Service is used/Invoked? The communication protocol, message formats, serialization techniques, and other service specific details are specified in conjunction with WSDL description of web service.* *It Relates to WSDL and SOAP in Web Service Technology.* *It is relatively complex.* *It is very difficult to implement automatic semantic annotation.*
WSML[8]	*Proposed in 2005* *It is specified to define formal statements in WSMO[12] is WSML. WSML is based on well-known logical formalisms: Description Logics & Logic Programming. WSML Syntax has two parts: Conceptual modeling & Arbitrary logical expressions. WSML is describes Ontology, Web Services, Goals and Mediator in WSMO framework.* • *Ontology* *It is used to specify semantic meaning using the vocabulary. Vocabulary can be specified using any of the Ontology available.* • *Web Service* *It is used to specify the functional, non-functional and behavioral aspects of Service. It specifies Pre and Post conditions, assumptions and effects.* • *Goals* *It specifies the user desires in respect to the requested functionality. So Input and Outputs are specified.* • *Mediator* *It defines the interoperability between heterogeneous resources using three level of mediation namely: Data, Protocol and Process.* *It is also Relatively complex.* *Semi-automated tools available for WSML annotations.*
WSDL-S[9]	*Proposed in 2005* *Rather than proposing new model for describing semantic web services, It defines some extensions to WSDL. WSDL 2.0 has the following constructs to represent service descriptions: interface, operation, message, binding, service and endpoint. The Service Definition is given by Interface, operation and message constructs. Binding, service and endpoint constructs provides service implementation. Dynamic discovery, composition and invocation of services is facilitated by service Description elements so WSDL-S does not includes specification for later three elements related to the invocation of web service. All extensional elements can be referring to an external semantic model by a URI.* *A quick summary of the extensibility elements and attributes provided in this proposal are given below:[]*

	• *modelReference* *It is specification of ontology concept for WSDL Elemtent.* • *schemaMapping* *It is used in XSD elements and complex types. It handles the structural differences between the schema elements of a Web service and their corresponding semantic model Reference.* • *precondition & effects* *They are child of operation element of WSDL. Each operations may have at most one Precondition and One effect element. Precondition and Effect can be expressed using logical expressions like 'and', 'or', 'xor'.* • *Category* *It specifies categorization information of the service to be published in registries like UDDI. This information is useful for discovery of web service.* *It is relatively less complex as elements are just extensions to existing WSDL.* *Semi-automated tools available for converting WSDL to WSDL-S.*
SAWSDL[10]	*Proposed in 2007* *SAWSDL is the first standardized technology for semantically annotating web Service by W3C(www.w3.org) , because of capability of interoperation between various Semantic Web Service Efforts like WSMO and OWL-S. It enables us to add semantic pointers to syntactic elements of WSDL, so even developed on WSDL 2.0, it is compatible with WSDL 1.0.* *A summary of the extension attributes defined by SAWSDL is given below:* • *modelReference* *A model reference specifies ontological meaning of element. Rather specifying modelreference for every element within WSDL and XSD, SAWSDL defines its meaning only for wsdl:interface, wsdl:operation, wsdl:fault, xs:element, xs:complexType, xs:simpleType and xs:attribute.* • *SchemaMapping* *It addresses the issues of post discovery in web service. For example, a client may have a given First name and Last name in its data. And Web service takes only one value as a Name - concatenation of First name and Last Name. So The schema mapping would take the client's semantic data and turn it into XML, and in the process it would perform the necessary concatenation to produce the Concatenated name. The extension attributes, named liftingSchemaMapping and loweringSchemaMapping, for specifying mappings between semantic data and XML.* *It is also less complex and is first W3C Recommendation.* *Lack of sophisticated automated tools for creating SAWSDL Annotations.*
WSMO-Lite[11]	*Proposed in 2010* *WSMO-Lite is a service ontology inspired from WSMO for the extension of SAWSDL. It adds semantics in the WSDL using Pre and Post Conditions. Specifically it defiles following semantic classes to be used with SAWSDL.* • *Condition* *Logical expression that must be hold before the service execution, called preconditions.* • *effect* *Logical Expression that shows the effects of Service Execution, called post conditions.* *The WSMO-Lite service ontology represent service contracts at the semantic level as follows* ⇒ *Information Model Descriptions* ⇒ *Functional Descriptions using capabilities and effects.* ⇒ *Nonfunctional Descriptions.*

III. SEMANTIC DISCOVERY OF WEB SERVICES

Web Services proved to be leading technology in software reusability, and to meet requirement of reducing time to market. Rapidly increase in the quantity of web services needs sophisticated mechanism for discovery of Web Service. Current Technologies using UDDI and WSDL is only providing semantic (keyword and category based) match and information retrieval techniques like clustering is used for improving search. The lack of semantics in description creates inefficiencies in exploiting the Web service discovery [13]. Current Service Discovery architecture relies on following architectural component to enable search [14].

- Service Registry: It is the API for service publication and searching. E.g. UDDI
- Service Request: It is the way the user will access the service.
- Matching Algorithm: It is the algorithm which matches Semantic request information with the Details of Service Registry.
- Service Advertisement: It is used to describe the service. E.g. WSDL

Semantic discovery of web services adds new components to existing architectural components for enabling semantic search. Specifications and efforts for developing Service Annotation Ontology (SAO), adds the formal service Description model and specifies the capability. Service description using service ontology enables the Automatic search of web services based on the semantic request and also the syntactic marching. It also relates the Services to the Domain ontology for reducing the ambiguity, pre and post conditions for improving effectiveness. The combination of service Ontology and Domain Ontology is the key to for the semantic Discovery of web services.

Semantic Discovery of Web Service employs following types of match while searching for service based on user request and also defines DoM as "A value from an ordered set of values that express how similar two entities are with respect to some similarity metric." [14]. When service Sp is requested by service Sr it would have, [14].

- Exact Match: If Service Request is exact match of Service Description or Service Request is direct sub class of service description.
- Subsume Match: If Service Request subsumes Service Description it is called subsume.
- Plug In match: If Service Description subsumes Request Description it is called Plug-in.
- Fail match: If there is no relation between Service Request and Service Description.

Semantic Discovery of web service also employs different approaches for matchmaking. As follows [14],

- Direct: When parameters match exactly it returns specific service or Composite service.
- Indirect: When there is no exact match it will compute match based on subsumes and returns composition of web service.
- Logic-based: Description Logics and First Order Logic reasoning
- Similarity-based: Using Information Retrieval Techniques
- Graph matching: Using Graph's shortest path for finding solution for user request.

Apart from these many other approaches available to serve match making but these are the most popular ones.

Following are the Tools for web service Discovery. [14]

1) *OWL-S/UDDI Matchmaker :* It searches OWL-S Services and used OWL as a Domain Ontology. It employs Description Logic based Subsumtions matching approach. It is available in a Stand-alone version as well as web-based version and also it is open source and Developed in JAVA. It can be accessible via http://projects.semwebcentral.org/projects/owl-s-uddi-mm/
2) *STWS : IBM Semantic Tools for Web Services*: It searches WSDL-S Web services using OWL domain ontology. It employs AI planning technique for computing a composite service when there is no single service is not sufficient to fulfill user request. It is available as a Eclipse Plug-in and exploits WordNet Lexicons. http://www.alphaworks.ibm.com/tech/wssem
3) *OWLS-MX : Hybrid OWL-S Web Service Matchmaker:* It searches OWL-S web services and uses OWL Domain Ontology. It employs logic based matching and syntactic similarity based matrix for matching user requirements with service description. It is also open source and developed in java and available at, http://www.dfki.de/~klusch/owls-mx
4) *Lumina: METEOR-S Web Service Discovery Infrastructure (MWSDI):* It searches WSDL-S web Services and using OWL domain ontology. The semantic discovery is based on semantics during whole life cycle of the web service. It is available as a Eclips Plug-in and it is also open source and developed in JAVA. http://lsdis.cs.uga.edu/projects/meteor-s/illumina/
5) *WSMX: WSMO Discovery Component:* It searches WSMO services using WSML ontology. It is a part of WSMO Reference implementation. It is also open source and available at http://www.wsmx.org/

IV. COMPOSITION OF SEMANTIC WEB SERVICES

Generally more than one service is combined to fulfill user request. The combined entity of web service is called Service Composition. Service composition is the most important activity of satisfying user request. Manual way of composing web service needs discovery, selection, Input and output to be mapped to user request. Publication of millions of web services needs Automation in the satisfying service request by composition. The semantic mark-up of web services can enable automation of this composition process by encoding the information necessary to select, compose, and respond to services at the service websites [15].

Several approaches for Semantic Web Service composition are available such as ontology-driven [16], agent-based [17], context-based [18], iterative [19], BPEL4WS based [20], template-based [21], logic-based [22], AI (Artificial Intelligence) planning-based [23] etc. All of them have their pros and cons but none of them is able to enable fully automated semantic web service composition. Web Service composition is such a complex task that requires many stages to be carefully designed and investigated. Web Service Composition needs following four steps to be executed successfully [25].

- Modeling of Data and Control flow
- Binding of the candidate services, for Composition, with the Process Activities.
- Advertisement of composition in the Service Registry
- Execution control and Monitoring of web service composition.

Following is the comparative study of the composition approaches. It is based on the different aspects required for the web service composition.

A. Ontology-Driven Web Service composition[16]

1) *Concept:* The composition is created from atomic services which are semantically annotated. The composition is described by DAML-S Ontology. Compositions are created using semantic interface matching between Candidate services and service request. Service Compositions are indexed and requester has to select the best composition.
2) *Semantic Support:* It is completely based on the semantic annotations for matching service request. But it has capability for extending it to simple atomic services without automation.
3) *Automation:* It supports semi-automated service composition, i.e. compositions are created using semantic interface matching and then user has to select the optimum composition.
4) *Control and Data Flow:* Automatically created by Process Composer which uses Interfafe-matching Automatic composition (IMA).
5) *Composition description language:* DAML-S
6) *Service Advertisement:* Service Profile in DAML-S is used for advertisement/publishing of composition in Service registry.
7) *Execution control:* Process execution component provides execution and monitoring.
8) *Interoperability with Other approaches:* DAML-S is lacking with interoperability with other approaches. But if the composition is create with interoperable service ontology like WSMO-Lite or SAWSDL it will provides interoperability with other approaches.

B. Agent based web service composition[17]

1) *Concept:* Agent-based technology considers each service as an intelligent agent. This concept leads to the Web Service Composition as a multi-agent system. Various agents in multi-agent system cooperate and negotiate possibly from geographically distributed network to form a software system.
2) *Semantic Support:* It depends on the requirement and approach used to build multi-agent system.
3) *Automation:* Multi-Attribute negotiation approach is used for automation.
4) *Control and Data Flow:* Automatic by communicating agents (Services).
5) *Composition description language:* No Specific composition description language is specified. DAML-S, DSMS-S, and OWL-S are used in several approaches.
6) *Service Advertisement:* Based on the composition language selected.
7) *Execution control:* No Execution control and Monitoring provided. Even coordinator based execution is specified, but the role of it varies.
8) *Interoperability with Other approaches:* As it does not specify any composition language, it can be designed to be interoperable with other approaches.

C. Context based web service composition[18]

1) *Concept:* Service compositing are developed using the ‘context’ information of service requester. Several approaches specified for context aware composition are reviewed in [18].
2) *Semantic Support:* The main idea is based on semantics of context awareness.
3) *Automation:* Automation of context searching and developing is available.
4) *Control and Data Flow:* It is based on the underlying technique for composition representation.
5) *Composition description language:* Not specified.
6) *Service Advertisement:* No specific Standard.
7) *Execution control:* Not provided.
9) *Interoperability with Other approaches:* Not provided.

D. Iterative web service composition[19]

1) *Concept:* It is based on functional specification refinements iteratively. Initially requester starts with the possibly incomplete or high level description of service requirements. In response to the initial request subset of the available service component is realized. This enables the requester to formulate the request with more detailed and accurate specification of request, which leads to the actual service composition.
2) *Semantic Support:* No support for Semantic compositions.
3) *Automation:* Automation is not available.
4) *Control and Data Flow:* It is based on WS-BPEL activities.
5) *Composition description language:* WS-BPEL and WSDL.
6) *Service Advertisement:* Using WSDL description of WS-BPEL composition.
7) *Execution control:* Not provided.
8) *Interoperability with Other approaches:* It is not providing.

E. BPEL4WS based Web Service Composition[20]

1) *Concept:* Static composition is approach. BPEL4WS provides activities that can be used modelling and describing service composition.
2) *Semantic Support:* No semantic support is provided.
3) *Automation:* Yet no effort found. So manual composition is only solution.

4) *Control and Data Flow:* Manual using Activities like if, loop, service binding etc.
5) *Composition description language:* BPEL4WS
6) *Service Advertisement:* Using WSDL for the composition.
7) *Execution control:* Not provided.
8) *Interoperability with Other approaches:* Not Available.

F. Template Based Web Service Composition[21]

1) *Concept:* The main idea is to use preconfigured work flow as a template for designing functionalities. Then binding the updated web services with the existing templates. Templates are described in WS-BPEL.
2) *Semantic Support:* Not available.
3) *Automation:* Manual effort required.
4) *Control and Data Flow:* Manual using WS-BPEL activities.
5) *Composition description language:* WS-BPEL
6) *Service Advertisement:* Using WSDL of composition.
7) *Execution control*: Not provided.
8) *Interoperability with Other approaches:* No interoperable.

G. Logic Based Web Service Composition[22]

1) *Concept:* It uses the Propositional linear logic(LL) for computing the composition using the atomic services, which matches the service request. The system is having following components.
   a. Translator: To convert presentation of Web Services to axioms in LL.
   b. LL Theorem Prover: proves that user's request can be satisfied with available atomic web service.
   c. Semantic Reasoner: Relates the concepts in the semantic descriptions of user request and web service semantics.
2) *Semantic Support:* It supports both semantics and simple web services for composition.
3) *Automation:* It is implemented fully automatic.
4) *Control and Data Flow:* Automatically created when composition is computed.
5) *Composition description language:* DAML-S or BPEL4WS.
6) *Service Advertisement:* Depends on composition language.
7) *Execution control:* Not provided.
8) *Interoperability with Other approaches:* Depends on the composition language used. As using DAML-S and BPEL4WS is not providing interoperability with other approaches.

H. AI Planning based web Service Composition[23]

1) *Concept:* The problem of service composition is seen as a problem of workflow design, where workflow is viewed as a AI planning Problem. Thus the composition is the task of designing work flow that will move towards fulfilment of user request by executing each work flow activity. AI Planning is used for automation in workflow generation.
2) *Semantic Support:* Provided by Web Service Modeling Framework (WSMF) and OWL-S.
3) *Automation:* Provided using AI Planning for work flow design.
4) *Control and Data Flow:* Automated based on the composition language used.
5) *Composition description language:* OWL-S and WSML
6) *Service Advertisement:* Available based on Service composition language.
7) *Execution control:* Provided using AI Planning.
8) *Interoperability with Other approaches:* It is not interoperable with other approaches.

V. CONCLUSION & FUTURE DIRECTIONS

The Application of semantic web and web services gives the rise to semantic web services. We have summarised different approaches available for adding semantic annotations in web services. Semantic annotations are the first and foremost requirement of creating machine process able software components. Different approaches for semantically discovering requested service improves the accuracy of matching. Different discovery approaches available for discovery of semantic web services. The best approach is chosen based on the application. Finally different approaches for creating composite web services for fulfilling user request are investigated. Dynamic and Automated composition approaches are the only solution for satisfying the user requests by creating composition millions of web services on the fly.

Semantic Web Services is the dream for the Automated Software Reusability between heterogeneous resources across the world. Realization of this dream needs Interoperable, efficient and light weight semantic annotation Language and tool for automatic annotation. Also the Efficient Semantic Web Service Discovery needed to improve the match making process and selection criterion. After all provision of Dynamic Semantic Web Service Composition has the capability to fulfil the Automated Software Reusability, applications based dynamic web service composition models needs to be developed.

REFERENCES

[1] Bartalos, Peter. *"Effective automatic dynamic semantic web service composition."* Inf. Sci. and Technol. Bulletin ACM Slovakia 3.1 (2011): 61-72.

[2] J. Cardiff, "*The Evolution of the Semantic Web,"* Proc. of 2nd Workshop on Semantic Web and New Technologies (SemWeb09), Puebla, Mexico, CEUR Workshop Proceedings vol. 534, 2009.

[3] Hakimpour, Farshad, Suo Cong, and D. Damm. "*A Practical Tutorial on Semantic Web Services.*" hakimpour. com.

[4] Graham, Steve, et al. Building Web services with Java: making sense of XML, SOAP, WSDL, and UDDI. SAMS publishing, 2004.

[5] Hu, Xiaocao, Zhiyong Feng, and Shizhan Chen. "*Augmenting Semantics of Web Services Based on Public Open Ontology.*" Services Computing (SCC), 2013 IEEE International Conference on. IEEE, 2013.

[6] Belhajjame, Khalid, Suzanne Embury, and Norman Paton. "*Verification of Semantic Web Service Annotations Using Ontology-Based Partitioning.*" (2013): 1-1.

[7] Martin, David, et al. "*OWL-S: Semantic markup for web services." W3C* member submission 22 (2004): 2007-04.

[8] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stoll-berg, A. Polleres, C. Feier, C. Bussler, and D. Fensel, "Web Service Modeling Ontology," Appl. Ontol., vol. 1, no. 1, pp. 77–106, 200

[9] Akkiraju, Rama, et al. "*Web service semantics-wsdl-s*." (2005).

[10] Kopecky, Jacek, et al. "Sawsdl*: Semantic annotations for wsdl and xml schema.*" Internet Computing, IEEE 11.6 (2007): 60-67.

[11] Vitvar, Tomas, et al. "Wsmo-*lite: Lightweight semantic descriptions for services on the web*." Web Services, 2007. ECOWS'07. Fifth European Conference on. IEEE, 2007.

[12] Roman, Dumitru, et al. "*Web Service Modeling Ontology*." Applied Ontology 1 (2005): 77-106.

[13] Xu, Jianjun, et al. "*Semantic Based Web Services Discovery*." Content Computing. Springer Berlin Heidelberg, 2004. 388-393.

[14] Tsetsos, Vassileios, Christos Anagnostopoulos, and Stathes Hadjiefthymiades. "*Semantic web service discovery: Methods, algorithms and tools.*" Semantic Web Services: Theory, Tools and Applications. IGI Global (2007).

[15] Kumar, Sandeep. *Agent-based semantic web service composition*. Springer, 2012.

[16] Zhang, Ruoyan. *Ontology-driven web services composition techniques*. Diss. uga, 2004.

[17] Kumar, Sandeep. *Agent-based semantic web service composition*. Springer, 2012.

[18] Jadeja, Yashpalsinh, Kirit Modi, and Ankur Goswami. "*Context Based Dynamic Web Services Composition Approaches: A Comparative Study*." International Journal of Information and Education Technology 2.2 (2012): 164-166.

[19] Marconi, Annapaola, Marco Pistore, and Paolo Traverso. "*An Iterative Approach for the Process-level Composition of Web Services.*" 3 rd SOUTH-EAST EUROPEAN WORKSHOP ON FORMAL METHODS. 2007.

[20] Wohed, Petia, et al. "Analysis *of web services composition languages: The case of bpel4ws*." Conceptual Modeling-ER 2003. Springer Berlin Heidelberg, 2003. 200-215.

[21] Sirin, Evren, Bijan Parsia, and James Hendler. "*Template-based composition of semantic web services*." Aaai fall symposium on agents and the semantic web. 2005.

[22] Rao, Jinghai, Peep Kungas, and Mihhail Matskin. "*Logic-based Web services composition: from service description to process model.*" Web Services, 2004. Proceedings. IEEE International Conference on. IEEE, 2004.

[23] Digiampietri, Luciano A., José J. Pérez-Alcázar, and Claudia Bauzer Medeiros. "*AI Planning in Web Services Composition: a review of current approaches and a new solution."* SBC 2007 (2007): 983-992.

[24] Digiampietri, Luciano A., José J. Pérez-Alcázar, and Claudia Bauzer Medeiros. "*AI Planning in Web Services Composition: a review of current approaches and a new solution*." SBC 2007 (2007): 983-992.

[25] 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)

