A Dynamic e-Tourism Application Using Semantic Web Services

MERAZI Afaf [1], MIMOUN Malki[2]
1 Evolutionary Engineering and Distributed Information Systems Laboratory
Computer Science Department, University of Sidi Bel Abbes
B.P 89 Sidi Bel Abbes, 22000 Algeria
[1]Afaf.Merazi@univ-sba.dz
[2]Malki@univ-sba.dz

Abstract : Semantic Web Services (SWS), the emerging convergence of Web Services with Semantic Web, is the next major generation of the Web (and of the Internet), in which e-services and business communication become more knowledge-based and agent-based. This paper discusses how SWS technologies have a particularly high chance to revolutionize one particular industry ”travel, i.e., its on-line aspect” which is called ”Dynamic Packaging” (DP). DP means dynamically (i.e., in real-time) putting together – and pricing – a package of several major travel components, e.g., air flight legs, hotel nights, car rental days, etc., from heterogeneous suppliers and heterogeneous information sources or back-end reservation services, even as those provide frequently changing availability or prices. To this end, this paper focus on approaches to semantic discovery and composition planning of semantic Web services, and briefly comment on their interrelationships. We propose an architecture that enables the integration of tourism data sources and creation of dynamic packages using Semantic Web services.

Keywords: Dynamic Packaging, Semantic Web services, Web Services, information sources, semantic discovery, semantic composition planning.

Introduction

Travel planning and booking is the most successful business model on the Web [1]. However, planning an individual trip on the Web is still a time consuming and a complicated endeavor. Most of the huge number of travel sites provides isolated information about flights, hotels, rental cars, weather or they relate that information in a very restricted manner letting the consumer/end user the heavy task of putting all the pieces together. There exists currently no integrated service for arranging personalized trips to any desired destination, relying on distributed information sources which have to be reasonably combined. Recent approaches build on mediators that turn Web sources into structured data sources. Those mediators are the critical component of the whole system because they have to be build individually and kept up to date.

”What is needed is an individual travel agent which is able to arrange journeys to virtually any place using first hand information from a huge set of different Web sources.” [2].

The objective of this paper is (a) to give a sense of how Web Services (WS) and Semantic Web Services (SWS) can contribute to help travel providers ensuring effective customer decision support online (b) How the potential adoption of SWS technologies can potentially change the current travel industry business models. The travel industry possesses key attributes that make Web services deployments attractive and inevitable (c) to provide a brief romp through the fields of SWS discovery and composition planning. We classified existing approaches, discussed representative examples. Despite fast paced research and development in the past years world wide, SWS technology still is commonly considered immature with many open theoretical and practical problems as mentioned above. However, its current convergence with Web 2.0 towards a service Web 3.0 in an envisioned Internet of Things helds promise to effectively revolutionize computing applications for our everday life.

1. An early Winning Area for SWS: Dynamic Packaging

” An industry buzzword for enabling the consumer to build a customized itinerary by assembling multiple components of their choices and complete the transaction in real time.” (Stephanie Lofgren).

DP is different from prepackaged travel. It is important to understand that DP and prepackaged travel are two concepts that are very different. Prepackaged travel relies on selling to the customer a complete package that includes usually flights, accommodations, car rental etc.. These packages are made sometime months in advance and published in brochures or sold online. These packages allow the different actors of the travel industry from producers to resellers to offer ”mass-market” products and to operate relatively simple business processes that allow them to have higher margins. These ”mass packages” offer: a) Fixed
itineraries, b) Inflexible dates, c) Limited options. But as they are made months in advance, they also often hinder the optimization of revenues through yield-management techniques that are based on adjusting price and availability to demand in realtime.

In DP, the process is different even if the result could seem to be the same to the end customer: here, the components are"drawn from the inventories of the travel producers and combined to satisfy a particular customer requirement which is collected during an interactive dialog".

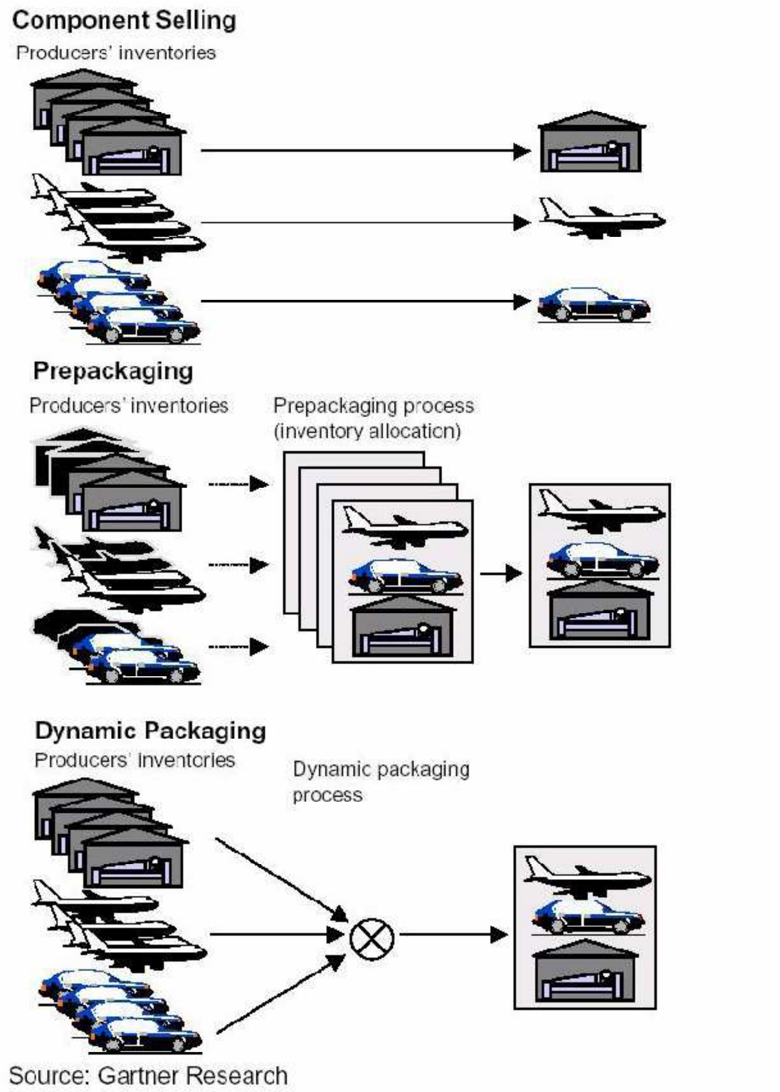

Figure 1 : Difference between DP and component selling

2. What are Semantic Web Services?

Web services: The current Web is not only a repository for static data, but furthermore offers interfaces to Web-accessible services to the human user, ranging from simple dynamically generated pages for pure information provision to more complex services. The next step after making the data on the Web machine processable is facilitating the direct interaction of applications, i.e. services, over the Web. Making this vision real should not solely be viewed in the context of the Web as such, but has high potential benefits in the areas of Enterprise Application Integration and Business-to-Business Integration, being the two most prosperous application areas of current Information Technology. Current technologies around SOAP [3], WSDL [4] and UDDI [5], often subsumed under the term ”Web services” only partly solve this integration problem by providing a common protocol (SOAP), interface description (WSDL) and directory (UDDI), but operating at a purely syntactic level.

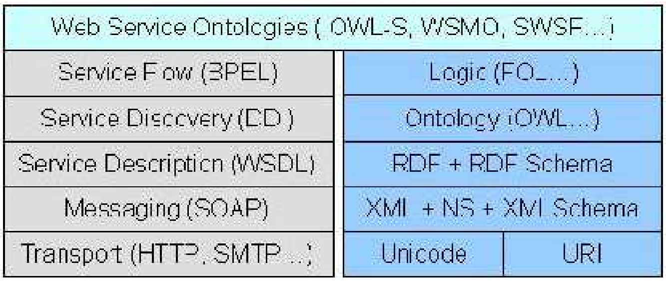

Figure 2 : Semantic Web services stack

Semantic Web services: The goal of what is called semantic Web services (SWS) [6] is the fruitful combination of Semantic Web technology and Web services. By using ontologies as the semantic data model for Web Service technologies adoption of Semantic Web technologies shall be adopted, i.e. Web services shall have machine-processable annotations just as static data on the Web. Semantically enhanced information processing empowered by logical inference eventually shall allow the development of high quality techniques for automated discovery, composition, and execution of Services on the Web, stepping towards seamless integration of applications and data on the Web. The W3C Semantic Web Services Interest Group has shown a strong interest in having more integrated semantics inside the Web Services stack, and also provides evidence of a rich variety of research proceeding in this area. This work aims towards the general objective of a more comprehensive, more expressive framework for describing all aspects of services, which can enable more powerful tools and fuller automation of a broad range of Web services activities. Semantic Web services frameworks (Figure. 2), such as OWL Service Ontology (OWL-S) [7] and, more recently, the Web Service Modeling Ontology (WSMO) [8] and the Semantic Web Services Framework (SWSF) [9] aim at providing means to semantically describe all necessary aspects of services in a formal way for creating such machine-readable annotations.

3. Creating Dynamic Packaging Based Semantic Web Service

In particular, our approach enables the functionality provided by existing legacy systems from the involved business partners to be exposed as Web services, which are then semantically annotated and published. From the bottom up the four application layers are:

- **Tourism Data Sources Layer:** consists of the existing data sources and IT systems available from each of the parties involved in the integrated application. It typically includes data stored in relational databases (other type of data source are also supported). At this level, we can find information which describes travel or tourism, namely, Computerized Reservation Systems, Global Distribution Systems, Hotel Distribution Systems, Destination Management Systems, and Web sites...

- **Service Abstraction Layer:** exposes the (micro-)functionality of the legacy systems as Web services, abstracting from the hardware and software platforms.

- **Semantic Web Services Layer:** In this paper we will focus on OWL-S as underlying language for annotating Web Services. OWL-S provides an ontological framework based on which an abstract description of a service can be created. It is an upper ontology whose root class is the Service class that directly corresponds to the actual service that is described semantically. The upper level Service class is associated with three other classes: Service Profile (specifies the functionality of a service), Service Model (specifies how to ask for the service and what happens when the service is carried out) and Service Grounding (specifies how the service has to be invoked). In particular, the service model tells a client how to use the service, by detailing the semantic content of requests, the conditions under which particular outcomes will occur, and, where necessary, the step by step processes leading to those outcomes. For nontrivial services (those composed of several steps over time), this description may be used by a service-seeking agent in different ways. The Service Model defines the concept Process that describes the composition of one or more services in terms of their constituent processes. A Process can be atomic (a nondecomposable service), composite (a set of processes within some control structure that defines a workflow) or simple (a service abstraction).

The semantic descriptions of web services make them machine interpretable and offers agents the possibility to automatically compose different services to a new composite service. This is a great benefit, because this composition had to be done manually by humans before. The automatic composition is mainly based on the usual process modeling techniques on the one hand and on AI planning on the other hand. The composition can be divided a three basic steps: (1) the discovery and matchmaking of existing services (2) the plan generation according to the composition goal (3) the execution of the plan and the monitoring of the execution. There are existing different levels of automation. In a semi-automatic plan generation environment, the system supports a human controller by filtering matching services according to the outputs of a previous one and the constraints of the user, but the human controller is responsible of choosing one. An automatic plan generation doesn't need a human controller anymore and the plan is, if possible with existing services, directly constructed by the system at design time and afterwards deployed to an execution engine. The most challenging one is an automatic plan generation with interleaving with execution. The composite service can either be changed at runtime or even constructed at runtime based on the current conditions. Our work focuses on the plan generation and interleaving with execution.

- **Querying Layer:** Our work is based on the latest version of OWL-S. We suggest the usage of SPARQL an expression language for modelling OWL-S preconditions, results conditions and effects which is presented in [10]. This layer is responsible for taking SPARQL query, translating it to"native" language, executing query and returning query results.

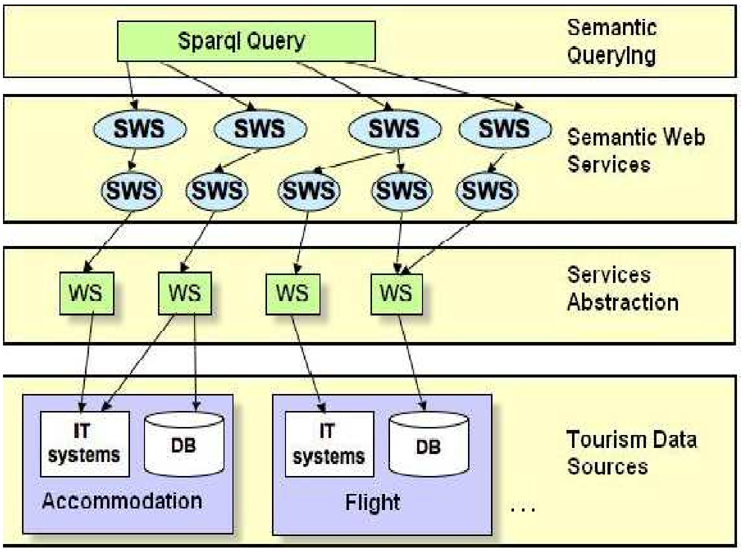

Figure 3 : The generic architecture used when creating Dynamic Packaging based SWS

4. Semantic Service Discovery

Semantic service discovery is the process of locating existing Web services based on the description of their functional and non-functional semantics. Discovery scenarios typically occur when one is trying to reuse an existing piece of functionality (represented as a Web service) in building new or enhanced business processes. Semantic service matching determines whether the semantics of a desired service (or goal) conform to that of an advertised service. This is at the very core of any semantic service discovery framework. Current approaches to semantic service matching can be classified according to:

- What kinds and parts of service semantics are considered for matching, and
- How matching is actually be performed in terms of non-logic based or logic based reasoning on given service semantics or a hybrid combination of both, within or partly outside the respective service description framework Non- logic based semantic profile matching.

In our approach, we focus our idea on non-logic based SWS matchmaker who does not perform any logical inferencing on service semantics. Instead, they compute the degree of semantic matching of given pairs of service descriptions based on, for example, syntactic similarity measurement, structured graph matching, or numeric concept distance computations over given ontologies. There is a wide range of means of text similarity metrics from information retrieval, approximated pattern discovery, and data clustering from data mining, or ranked keyword. In this sense, non-logic based semantic service matching means exploit semantics that are implicit in, for example, patterns, subgraphs, or relative frequencies of terms used in the service descriptions, rather than declarative IOPE semantics explicitly specified in the considered logic. Few examples, there is the OWLS-iMatcher [11], The DSD matchmaker [12].

In our previous work, the ”Ontology mapping and semantic querying framework” [13, 14], which imprecisely queries a set of ontologies that are stored as standard formats OWL in a relationnal database with an extension of RDQL, called SPARQL, based on four, lexical, sturucture, taxonomy and aggregation similarities metrics from information retrieval. The results are ranked according to the numerical scores of these methods measurements, and a user-defined threshold. As mentioned above, due to its generic functionality, our work is defined as a service matchmaker and can be used in arbitrary discovery architectures and systems by replacing the OWL ontologies by a set of OWL-S service profiles.

5. A Brief Survey of Semantic Service Composition Planning

Semantic Web service composition is the act of taking several semantically annotated component services, and bundling them together to meet the needs of a given customer. Automating this process is desirable to improve speed and efficiency of customer response, and, in the semantic Web, supported by the formal grounding of service and data annotations in logics.

In general, Web service composition is similar to the composition of workflows such that existing techniques for workflow pattern generation, composition, and management can be partially reused for this purpose [15]. Typically, the user has to specify an abstract workflow of the required composite Web service including both the set of nodes (desired services) and the control and data flow between these nodes of the workflow network. In particular, the mainstream approach to composition is to have a single entity responsible for manually scripting such workflows (orchestration and choreography) between WSDL services of different business partners in BPEL [16, 17]. This is largely motivated by industry to work for service composition in legally contracted business partner coalitions - in which there is, unlike in open service environment, only very limited need for automated service composition planning, if at all. Besides, neither WSDL nor BPEL or any other workflow languages like UML2 or YAWL have formal semantics which would allow for an automated logic based composition. In fact, the majority of existing composition planners for semantic Web services draws its inspiration from the vast literature on logic based AI planning [18].

The service composition problem roughly corresponds to the state based planning problem (I, A, G) in AI to devise a sound, complete, and executable plan which satisfies a given goal state G by executing a sequence of services as actions in A from a given initial world state I. Classical AI planning focuses on the description of services as deterministic state transition (actions) with preconditions, and state altering (physical) effects that are applicable to states based on the evaluation of preconditions and yield new states where the effects are valid. Further, classical planning is performed under the assumption of closed world with complete, fully observable initial states. The goal and all logic based semantic service concepts (IO parameter values, preconditions and effects) defined in a formal ontology (domain or background theory) and outside are converted to one declarative (FOL) planning domain and problem description that serves a given logic based AI planner as input. In particular, service outputs are encoded as special non-state altering knowledge effects, and inputs as special preconditions.

The standard language for this purpose is PDDL (Planning Domain Description Language) but alternative representation formalisms are, for example, the situation calculus [19], linear logic [20], high-level logic programming languages based on this calculus like GOLOG [21], Petri nets, or HTN planning tasks and methods [22].

However, as pointed out in [23], the naive adoption of classical AI planning for service compositions has severe limits. In particular, they are insufficient for planning under uncertainty in open service environments where (a) the initial state is incomplete, and (b) actions may have several possible (conditional) outcomes and effects that are modeled in the domain but not deterministically known at planning time, or unknown outcomes at all that can be determined only at run-time. We survey implemented functional and process level composition planner for semantic Web services that rely on either classical planning or planning under uncertainty in the following.

In general, any AI planning framework for semantic Web service composition can be characterized by:

- The representation of the planning domain and problem to allow for automated reasoning on actions and states,
- The planning method applied to solve the given composition problem in the domain, and
- The service semantics that are used for this purpose.

We can classify existing semantic Web service composition planners according to the latter two criteria, which yields the following classes, see Figureure (Figure. 4):

- **Dynamic or static SWS composition planners** depending on whether the plan generation and execution are inherently interleaved in the sense that actions can be executed at planning time, or not. the majority of SWS composition planners such as MetaComp [24], PLCP [25], RPCLM-SCP [26] and AGORA-SCP [20] are static classical planners. Approaches to dynamic composition planning with different degrees of interleaving plan generation and execution are rare. Unlike the static case, restricted dynamic composition planners allow the execution of information gathering but no world state altering services, hence are capable of planning under uncertainty about action outcomes at planning time. Examples of such composition planners are SHOP2 [27, 22], GOLOG-SCP [21] and OWLS-XPlan1 [28]. Advanced and reactive dynamic composition planners in stochastic domains even take non-deterministic world state changes into account during planning. While advanced dynamic planners like OWLS-XPlan2 [29] are capable of heuristic replanning subject to partially observed (but not caused) state changes that affect the current plan at planning time, their reactive counter-parts like INFRAWEBS-RTC [30] fully interleave their plan generation and execution in the fashion of dynamic contingency and real-time planning.

- **Functional level or process level SWS composition planners** depending on whether the plan generation relies on service profile (data flow/IOPE) semantics only, or process model semantics in addition (data and control flown) [26]. most SWS composition planners perform functional level or service profile based composition (FLC) planning. FLC planning considers services as atomic or composite black-box actions which functionality can solely be described in terms of their inputs, outputs, preconditions, and effects, and which can be executed in a simple request-response without interaction patterns. Examples of FLC planners are SAWSDL-SCP [31] and OntoMat-S [32]. Process level composition (PLC) planning extends FLC planning in the sense that it also the internal complex behavior of existing services into account. Prominent examples are SHOP2 [22], PLCP [33] and OWLS-XPlan [28, 29] . Both kinds of composition planning exploit semantic profile or process matching means that is either inherent to the AI planning mechanism, or provided by a connected stand-alone matchmaker.

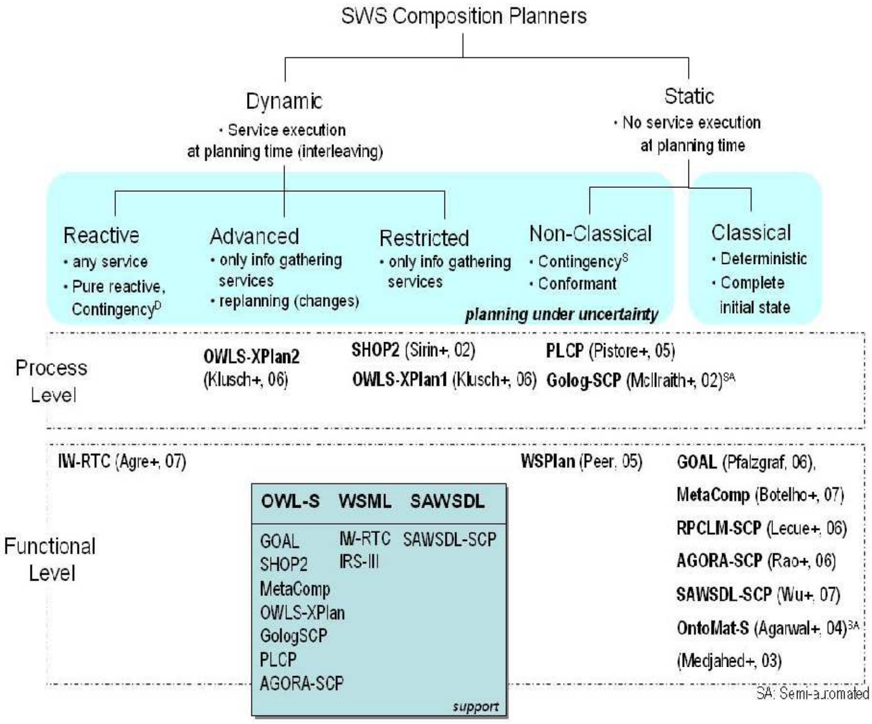

Figure 4 : Classes of semantic Web service composition planners.

As mentionned above, the advantage of this approach, in which we frame our methodology, is the direct use of the Semantic Web formalisms. In this manner, we are able to use methodologies coming from more consolidated research fields exploiting the advantages that Semantic Web guarantees, i.e. a distributed knowledge base and the semantic interoperability. In our work, it is possible to build composer exploiting only the Semantic Web technology to achieve the composition task. Our immediate future plans lie in exploiting SWRL for OWL-S atomic services composition [34]. This work can be considered as a starting point for the solution of a broader issue like the orchestration of SWS.

Conclusion

Semantic Web services research has the overall vision of bringing the Web to its full potential by enabling applications to be created automatically from available Web services in order to satisfy user goals. Fulfilling this vision will radically change the character of all online interaction including the nature of e-Commerce, e-Science, e-Learning, and e-Government. Key to achieving this vision is the provision of SWS platforms able to support the development and use of online libraries of reusable software components indexed through generic and domain specific ontologies.

This paper provided a brief romp through the fields of SWS discovery and composition planning. We classified existing approaches, discussed representative examples and commented on the interrelationships between both service coordination activities. Future work will mainly consist of exploiting SWRL for OWL-S atomic services composition.

References

[1] PhoCusWright: The Online Travel Marketplace 2001 - 2003 (2001).

[2] T. Berners-Lee: Are we done yet? Keynote Speak at WWW10, Hong Kong, China (2001).

[3] World Wide Web Consortium: SOAP Version 1.2. Technical Recommendation, 24 June 2003, http://www.w3.org/TR/soap/, (2003)

[4] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana: Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/wsdl, March (2001).

[5] T. Bellwood, L. Clement, D. Ehnebuske, A. Hately, M. Hondo, Y. Husband, K. Januszewski, S. Lee, B. McKee, J. Munter, and C. von Riegen: OASIS UDDI Spec TC: UDDI Version 2.03 Data Structure Reference. UDDI Committee Specification, 19 July 2002, http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm, (2002).

[6] S. McIlraith, T. C. Son, and H. Zeng: Semantic web services. IEEE Intelligent Systems, Special Issue on the Semantic Web, 16(2):4653, (2001).

[7] A. Barstow, J. Hendler, M. Skall, J. Pollock, D. Martin, V. Marcatte, D. L. McGuinness, H. Yoshida, and D. D. Roure: OWL Web Ontology Language for Services (OWL-S), Nov. 2004. http://www.w3.org/Submission/2004/07/, (2004).

[8] The WSMO working group: Web Service Modeling Ontology (WSMO), W3C Member Submission, 3 June 2005, http://www.w3.org/Submission/2005/SUBM-WSMO-20050603/, (2005).

[9] The Semantic Web Service Language (SWSL) Committee: Semantic web services framework (SWSF) version 1.0, 2005. http://www.daml.org/services/swsf/1.0/, (2005).

[10] M. L. Sbodio, C. Moulin: SPARQL as an expression language for OWL-S. In OWLS: Experiences and Directions Workshop at 4th European Semantic Web Conference (ESWC) (2007)

[11] J. Bae, L. Liu, J. Caverlee, W.B. Rouse: Process Mining, Discovery, and Integration using Distance Measures. Proceedings of International COnference on Web Services ICWS, (2006).

[12] U. Kuster, B. Konig−Ries, M. Stern, M. Klein: DIANE: An Integrated Approachuo to Automated Service Discovery, Matchmaking and Composition. Proceedings of the World Wide Web COnference WWW, Banff, Canada, ACM Press, (2007).

[13] S. M. Benslimane, A. Merazi, M. Malki, and Djamel A. Bensaber: Ontology mapping for querying heterogeneous information sources. INFOCOMP Journal of Computer Science, 7(2) : 44-51, June 2008. ISSN : 1807-4545, (2008).

[14] A. Merazi, S. M. Benslimane, and M. Malki: Towards semantic interoperability: An ontology mapping approach. In: Proc. of 4th International Conference on Optimisation and Information Systems (COSI'07), Oran, Algeria, June (2007).

[15] L. Henoque, M. Kleiner: Composition - Combining Web Service Functionality in Composite Orchestrations, (2007).

[16] M. Papazoglou: Web Services. Principles and Technology. Pearson - Prentice Hall, September (2007).

[17] G. Alonso, F. Casati, H. Kuno, V. Machiraju: Web Services. Springer, (2003).

[18] J. Peer: Web Service Composition as AI Planning: A Survey. Technical Report, University of St. Gallen, Switzerland, 2005. Available at elektra.mcm.unisg.ch/pbwsc/docs/pfwsc.pdf, (2005).

[19] S. Narayanan, S. McIllraith: Simulation, verification and automated composition of web services. Proc. of 11th International COnference on the World Wide Web (WWW), Hawaii, (2002).

[20] J. Rao, P. Kuengas, M. Matskin: Composition of Semantic Web services using Linear Logic theorem proving. Information Systems, 31, (2006).

[21] S. McIllraith, T.C. Son: Adapting Golog for composition of semantic Web services. Proc. International Conference on Knowledge Representation and Reasoning KRR, Toulouse, France, (2002).

[22] E. Sirin, B. Parsia, D. Wu, J. Hendler, D. Nau: HTN planning for Web service composition using SHOP2. Web Semantics, 1(4), Elsevier, (2004).

[23] B. Srivastava, J. Koehler: Web Service Composition: Current Solutions and Open Problems. Proceedings of the ICAPS 2003 Workshop on Planning for Web Services, (2003).

[24] M. Pistore, P. Roberti, P. Traverso: Process-Level Composition of Executable Web Services: On-the-fly Versus Once-for-all Composition Proceedings of the 2nd European Semantic Web Conference (ESWC), Heraklion, Crete, LNCS 3532, Springer, (2005).

[25] F. Lecue, A. Leger: Semantic Web service composition through a matchmaking of domain. Proc. of 4th IEEE European Conference on Web Services (ECWS), Zurich, (2006).

[26] E. Sirin, J. Hendler, B. Parsia: Semi-automatic Composition of Web Services using Semantic Descriptions. Proceedings of Intl Workshop on Web Services: Modeling, Architecture and Infrastructure workshop in conjunction with ICEIS conference, (2002).

[27] M. Klusch, A. Gerber, M. Schmidt: Semantic Web Service Composition Planning with OWLS-XPlan. Proc. 1st Intl. AAAI Fall Symposium on Agents and the Semantic Web, Arlington VA, USA, AAAI Press, (2005).

[28] M. Klusch, K-U. Renner: Dynamic Re-Planning of Composite OWL-S Services. Proc. 1st IEEE Workshop on Semantic Web Service Composition, Hongkong, China, IEEE CS Press, (2006).

[29] G. Agre, Z. Marinova: An INFRAWEBS Approach to Dynamic Composition of Semantic Web Services. Cybernetics and Information Technologies (CIT), 7(1), (2007).

[30] Z. Wu, K. Gomadam, A. Ranabahu, A. Sheth, J. Miller: Automatic Composition of Semantic Web Services using Process Mediation. Proceedings of the 9th Intl. Conf. on Enterprise Information Systems ICES 2007, Funchal, Portugal, (2007).

[31] S. Agarwal, S. Handschuh, S. Staab: Annotation, composition and invocation of semantic web services. Web Semantics, 2, (2004).

[32] M. Pistore, P. Traverso, P. Bertoli, A. Marconi: Automated synthesis of composite BPEL4WS web services. Proceedings of the 2005 IEEE International Conference on Web Services, Orlando, USA, IEEE Press, (2005).

[33] D. Redavid, L. Iannone and T. Payne: OWL-S Atomic services composition with SWRL rules. In: Proceedings of the 4th Italian Semantic Web Workshop, 18-20 December, 2007, Dipartimento di Informatica - Universita' degli Studi di Bari - Italy, (2007).