Semantic Awareness in Automated Web Service Composition through Planning

Ourania Hatzi[1], Dimitris Vrakas[2], Nick Bassiliades[2], Dimosthenis Anagnostopoulos[1] and Ioannis Vlahavas[2]

[1] Department of Informatics and Telematics, Harokopio University of Athens, Greece
{raniah, dimosthe}@hua.gr
[2] Department of Informatics, Aristotle University of Thessaloniki, Greece
{dvrakas, nbassili, vlahavas}@csd.auth.gr

Abstract. PORSCE II is a framework that performs automatic web service composition by transforming the composition problem into AI planning terms and utilizing external planners to obtain solutions. A distinctive feature of the system is that throughout the entire process, it achieves semantic awareness by exploiting semantic information extracted from the OWL-S descriptions of the available atomic web services and the corresponding ontologies. This information is then used in order to enhance the planning domain and problem. Semantic awareness facilitates approximations when searching for suitable atomic services, as well as modification of the produced composite service. The alternatives for modification include the replacement of a certain atomic service that takes part in the composite service by an equivalent or a semantically relevant service, the replacement of an atomic service through planning, or the replanning from a certain point in the composite service. The system also provides semantic representation of the produced composite service.

Keywords: Automatic Web Service Composition, Semantic Web Services, AI Planning, Semantic Awareness, Semantic Matching Relaxation.

1 Introduction

Web services play an important role in the Web today, as they accommodate the increasing need for interoperability and collaboration between heterogeneous systems that expose their functionality over a network. The web services technology provides a way to communicate and interact with such information systems through a standard interface, which is independent from platform and internal implementation. In many cases, as the user requirements shift towards more complex functionality, they cannot be fulfilled by a simple atomic web service. This shortcoming can be handled by web service composition; that is, the appropriate combination of certain atomic web services in order to achieve a complex goal. The task of web service composition becomes significantly difficult, time-consuming and inefficient as the number of available atomic web services increases continuously; therefore, the capability to automate the web service composition process is proved essential.

Automated web service discovery, invocation, composition and interoperation is significantly facilitated by the existence of semantic information in the atomic web services description [13]; such information represents knowledge about the actual meaning of services. The incorporation of semantics in the description of web services is accommodated through the development of a number of standard languages such as OWL-S [3] and SAWSDL [5], leading to the notion of semantic web services, which are defined, evolve and operate in the Semantic Web. The existence of semantics facilitates composition using intelligent techniques, such as AI Planning. Without the presence of semantic information, a high degree of human expertise would be required in order to compose web services meaningfully and not based on circumstantial syntactic similarities.

The focus of this paper is on the incorporation of the semantic information in the web service composition process and its effects on various aspects of the PORSCE II framework. PORSCE II aims at automated semantic web service composition by utilizing planning techniques. The process exploits information in the OWL-S descriptions of atomic web services, translates the web service composition problem to a planning problem, exports it to PDDL [2] and invokes external planning systems to acquire plans, which constitute descriptions of the desired composite service. Each composite service is evaluated in terms of statistic and accuracy measures, while a visual component is also integrated, which accommodates composite service visualization and manipulation. Modification in the composite web service is performed by atomic service replacement, either with an alternative equivalent atomic service, or through finding a sub-plan that can substitute it. If necessary, the composite service can also be modified through replanning. Finally, in order to provide full-cycle support, and render the result of the composition process independent from planning, the composite service is translated back to OWL-S, presenting the user with a description in the same standard as the initial atomic services and facilitating composite service deployment.

Semantic awareness in PORSCE II is achieved by exploiting semantic information extracted from the OWL-S descriptions of the available atomic web services and the corresponding ontologies, and analyzing this information based on semantic distance measures and user-defined thresholds. The derived knowledge is then used to enhance the planning domain and problem, achieve approximate solutions, when necessary, and accommodate intervention to the composite service.

The rest of the paper is organized as follows: Section 2 discusses some related work, while Section 3 describes the system architecture and functionality. Section 4 elaborates on the effects of semantic awareness on various aspects of the framework, along with examples. Finally, Section 5 concludes the paper and poses future directions.

2 Related Work

OWLS-Xplan [7] uses the semantic descriptions of atomic web services in OWL-S to derive planning domains and problems, and invokes the XPlan planning module to generate the composite services. The system is PDDL compliant, as the authors have developed an XML dialect of PDDL called PDDXML. Although the system imports semantic descriptions in OWL-S, the semantic information provided from the domain ontologies is not utilized and semantic awareness is not achieved; therefore, the planning module requires exact matching for service inputs and outputs.

Another system that attempts automatic web service composition through AI Planning is SHOP-2 [6]. The system uses services descriptions in DAML-S, the predecessor of OWL-S, and performs Hierarchical Task Network (HTN) planning to solve the problem. The main disadvantage of this approach lies in the fact that the planning process, due to its hierarchical nature, requires the specification of certain decomposition rules, which have to be encoded in advance by an expert in the specific domain, with the help of a DAML-S process ontology.

Other approaches for automatic web service composition can be found at [8] and [9]; however, they are not further discussed here as they do not deal with semantic descriptions of web services, or incorporating semantics in the composition process.

The main advantage of the proposed framework with respect to the aforementioned systems is the extended utilization of semantic information, in order to perform planning under semantic awareness and relaxation, in order to find better and, when necessary, approximate solutions. Furthermore, PORSCE II does not require any prior, domain-specific knowledge to formulate valid, desired composite services; the OWL-S descriptions of the atomic web services and the corresponding ontologies suffice. Finally, the system is able to handle cases of service failure or unavailability dynamically, through composite service modification, taking into account not only syntactic similarities but also semantics, which is an important feature not covered in the aforementioned frameworks.

3 Framework Architecture & Overview

PORSCE II aims at a higher degree of integration as, additionally to the core transformation and composition component, it contains a visual interface, composite web service manipulation features, and interconnection with multiple external planners. The key features of the framework include:

- Translation of OWL-S atomic web service descriptions into planning operators.
- Interaction with the user in order to acquire their preferences regarding the composite service and desired metrics for semantic relaxation.
- Enhancement of the planning domain with semantically similar concepts.
- Exporting the web service composition problem as a PDDL planning domain and problem.
- Acquisition of solutions by invoking external planners.
- Assessing the accuracy of the composite services.
- Visualizing and modifying the solution by service replacement or replanning.
- Transformation of the solution (composite web service) back to OWL-S.

PORSCE II comprises of the OWL-S Parser, the Transformation Component, the OWL Ontology Manager (OOM), the Visualizer and the Service Replacement Component. An overview of the architecture and the interactions among the components is depicted in Fig. 1.

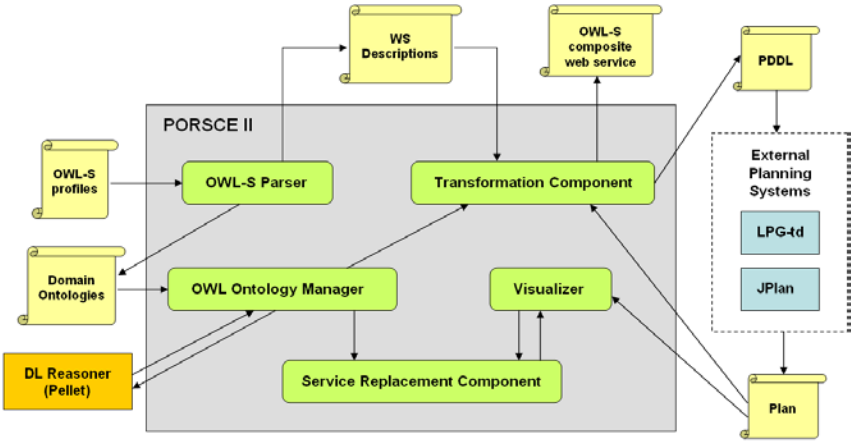

Fig. 1. The architecture of PORSCE II.

The OWL-S Parser is responsible for parsing a set of OWL-S web service profiles and determining the corresponding ontologies that organize the concepts appearing in the web service descriptions as inputs and outputs.

The OWL Ontology Manager (OOM), utilizing the inferencing capabilities of the Pellet DL Reasoner, applies the selected distance measure and thresholds for discovering concepts that are semantically relevant to a query concept. Upon request, it is able to provide the rest of the system with advice on semantically relevant or equivalent concepts, facilitating the implementation of semantic awareness.

The Transformation Component is the core component of the system. It is responsible for a number of operations that include the formulation of the planning problem from the initial web service composition problem, its consequent solving, and the transformation of the produced composite web service back to OWL-S. Throughout the process, it requests advice from the OOM in order to semantically enhance these procedures. The Transformation Component is also responsible for evaluating the produced composite services, according to their semantic distances obtained from the OOM.

The purpose of the Visualizer is to facilitate comprehension of the results, by providing the user with a visual representation of the plan, which in fact is the description of the composite service. The composite service is visualized as a web service graph, that is, a graph G=(V, E), where the nodes in V correspond to all the atomic services in the plan and the edges ($x \rightarrow y$) in E, where x and y are nodes in V, define that web service x produces an output that serves y as an input.

Finally, the Service Replacement Component enables the user to employ a number of alternative techniques in order to replace a specific atomic web service in the composite service sequence. In order to be able to perform replacement approximately, in cases than no exact matching candidates can be found, the Service Replacement Component requests semantic information by the OOM.

More on the interoperability between the systems and the functionality of PORSCE II can be found at [11], while the system, along with test cases, is available online at http://www.dit.hua.gr/~raniah/porsceII_en.html.

4 Effects of Semantic Awareness

In order to achieve semantic awareness, the system needs to be aware of semantic equivalences and similarities among syntactically different concepts, used as inputs and outputs of the web services. Semantic awareness is implemented by including all the required semantic information in a pre-processing phase, and letting the planning system handle the problem as a classical planning problem. The advantages of this approach include independency from the planner and minimization of the interactions between the planning system and the OOM.

The OOM provides the rest of the system modules with advice on demand about the equivalent and the semantically relevant ontology concepts to a query concept. A semantic distance can be assigned to each pair of ontology concepts, based on their hierarchical relationship (subclass, superclass, sibling, etc) and semantic distance metrics (edge-counting or upwards cotopic) [11].

The semantic awareness affects the system in four main aspects:

- enhancement of the planning domain and problem with semantic information
- inclusion of semantically equivalent and relevant services during composition
- search among semantically equivalent and relevant services for replacement
- semantic representation of the composite service

The examples that will be used in this section are extracted from a case study which combines and modifies the *books* and *finance* domains of the OWLS-TC (version 2.2 revision 1) semantic web service descriptions test sets [1]. The implemented scenario concerns the electronic purchase of a book. The user provides as inputs a book title and author, credit card info and the address that the book will be shipped to, and requires a charge to their credit card for the purchase, as well as the shipping dates and the customs cost for the specific item. The initial state of the planning problem is produced by the inputs of the composite service, while the goal state is produced by the desired composite service outcomes. A concise presentation of the inputs and outputs of the web services of interest for this scenario is provided in Table 1.

Table 1. Inputs and outputs of the web services of interest for the specific examples.

Service	Inputs	Outputs
BookToPublisher	Book, Author	Publisher
CreditCardCharge	OrderData, CreditCard	Payment
ElectronicOrder	Electronic	OrderData
PublisherElectronicOrder	PublisherInfo	OrderData
ElectronicOrderInfo	Electronic	OrderInformation
Shipping	Address, OrderData	ShippingDate
WaysOfOrder	Publisher	Electronic
CustomsCost	Publisher, OrderData	CustomsCost

4.1 Semantic Domain Enhancement

In a pre-processing phase, the Transformation Component probes the OOM in order to acquire all the semantically relevant concepts for both the facts of the initial state and the outputs of the planning operators. Consequently, semantic enhancement abides by these three rules:

1. The concepts of the initial state together with the semantically equivalent and similar concepts form a new set of facts noted as the Expanded Initial State (EIS).
2. The goals of the problem remain the same.
3. The Enhanced Action Set (EAS) is produced by altering the description of each operator by enhancing its effects with all equivalent and semantically similar concepts. Note that the initial size of the set is preserved.

Fig. 2 shows an example of the planning problem produced by the aforementioned web service composition domain, before and after the semantic enhancement, along with the semantically relevant concepts returned by the OOM. For legibility purposes, a shortened version of the domain described above is used.

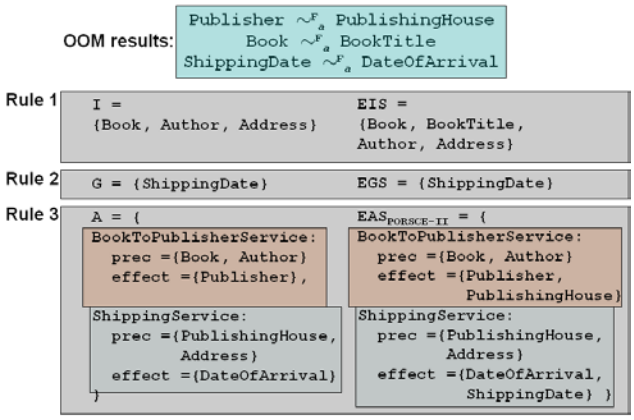

Fig. 2. An example of the semantic domain and problem enhancement.

The new problem, namely <EIS, EAS, G> is encoded into PDDL and forwarded to an external PDDL-compliant planning system in order to acquire solutions. Note that the semantic information is encoded in such a way that it is transparent to the external planner, which can solve the problem as any other classical planning problem.

4.2 Semantic Composition

The produced plans, or descriptions of the desired composite service, are consequently imported into the Visualizer Component, where they are transformed into a web service graph and represented visually. While solutions with exact matching of input to output concepts is obligatory in the classical planning domains, in the web services world the case can be different, as it is preferable to present the user with a composite service that approximates the required functionality than to present no service at all. The semantic awareness achieved in the PORSCE II system enables the composition of alternative services that approximate the desired one in case there are no exact matches, by performing semantically relaxed concept matching. Such a case is presented in Fig. 3. The `ElectronicOrderInfoService` produces as output an instance of the concept `OrderInformation`, while the available atomic services that are needed to fulfill the goals (`CreditCardCharge`, `CustomsCost` and `Shipping`) accept as input instances of the concept `OrderData`. Without semantic awareness and relaxation, an approximate matching between these concepts would not be possible, and the approximate plan of Fig. 3 would not be produced. However, under semantic relaxation, these two concepts are annotated as semantically relevant by the OOM. As this is not an exact matching service, the accuracy measure is changed to reflect it, following the accuracy definitions in [11].

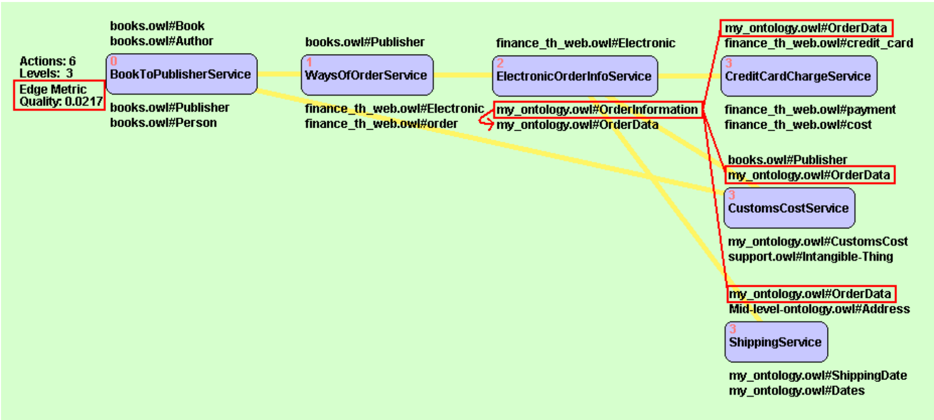

Fig. 3. Approximate composite service.

4.3 Semantic Service Replacement

The simplest alternative for composite service modification is the replacement of an atomic service included in the composite service (plan) with a semantically equivalent or relevant one. The replacement takes into account semantics, therefore the discovery of all actions that could be used alternatively instead of the chosen one is guided by advice from the OOM. An action A is considered an alternative for an action Q of the plan as far as it does not disturb the plan sequence and the intermediate states, that is, $prec(A) \subseteq prec(Q)$ and $add(A) \supseteq add(Q)$.

In cases when none of the semantically equivalent or relevant services that correspond to an atomic service is considered suitable, or in cases where there are no alternative services, the system offers the option to substitute the atomic service with a partially ordered set of atomic services, in the form of a subplan, found through planning. In this case, the world states right before and after the execution of the action being replaced serve as the initial and goal states for the planning process, respectively. The world state right before the application of the selected action can be found by starting at the original initial state and progressively including all the add effects of all intermediate actions up to the selected one. Likewise, the world state after the execution of the selected action can be found by starting at the original goal state, subtracting all add effects and including all preconditions of all intermediate operators, going backwards from the end of the plan to the selected action. Note that the replanning process is bound to return the atomic service being replaced itself, especially if the external planner used produces the optimal plan in each case. In order to prevent that, this specific service has to be removed from the set of available services before the replanning process proceeds.

In both cases, the selected alternative substitutes the original service and the new quality metrics are incorporated in the quality metrics of the entire plan.

If replacement of an atomic service, either by an equivalent or through replanning, is not a suitable option, the user can resort to replanning from a certain point in the plan, or even replanning from scratch. When replanning from a certain point, the world state at that point has to be calculated starting at the original initial state and progressively including all the add effects of all intermediate actions up to that point.

As an example, consider that replacement with equivalent is selected on the `CreditCardChargeService`. The resulting plan after the replacement with a semantically equivalent service is shown in Fig. 4.

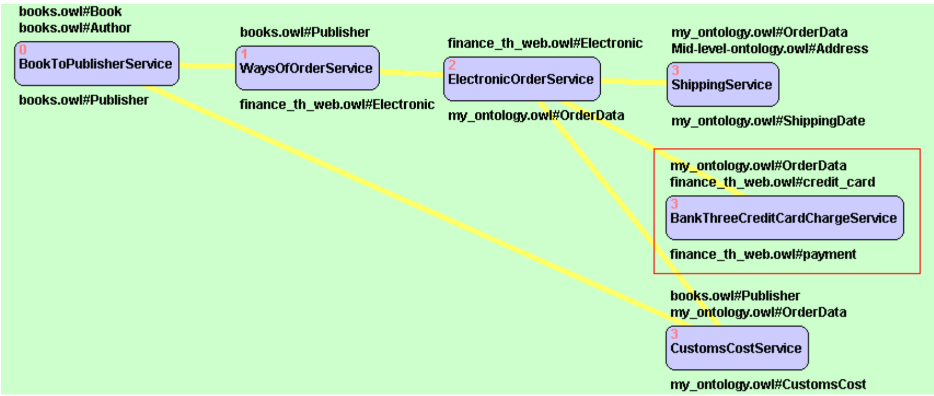

Fig. 4. Composite service after replacement by semantically equivalent service.

If the replacement through replanning alternative is selected, for example on the `CustomsCostService`, the corresponding planner will be re-invoked, finding a new sequence of actions that can substitute the selected service. The user interface for the replacement options is depicted in Fig. 5, while the resulting composite service after the modifications is depicted in Fig. 6.

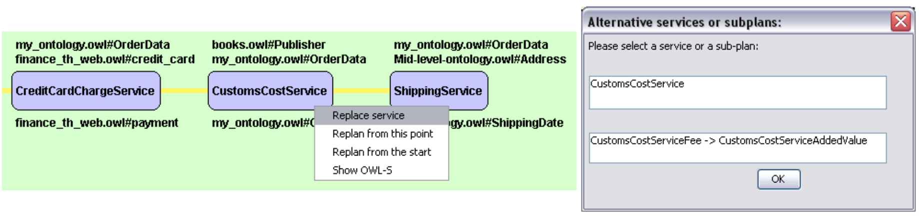

Fig. 5. The service replacement interface.

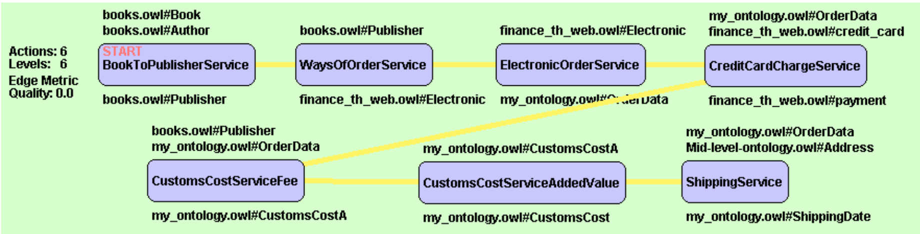

Fig. 6. Composite service after replacement through replanning operation.

4.4 Semantic Composite Service Representation

Semantic descriptions of web services in OWL-S [3] allow their use by software agents. As far as the composition of web services is concerned, OWL-S establishes a framework for semantically defining composite processes or services. A composite process is a set of atomic processes, combined together using a number of control constructs, such as Sequence, Split, Split + Join, Choice, Any-Order, Condition, If-Then-Else, Iterate, Repeat-While, and Repeat-Until. The main reasons for using these constructs while defining a composite web service are: a) to enable the definition of compact services (i.e. through the use of Iterate, Repeat-While and Repeat-Until), b) to facilitate the definition of alternative paths (i.e. through the use of Conditions and If-Then-Else constructs) and c) to speed up the invocation of the composite web service, by allowing multiple atomic processes to be invoked concurrently (i.e. through the use of Split and Split+Join constructs).

For the purposes of the PORSCE II framework, the use of these constructs is not mandatory, as far as the proper invocation of the atomic processes is concerned. Since the modeling of the web service composition problem to a planning problem, is merely based on the STRIPS formalism, there is no need for defining alternative paths. However, in order to speed up the invocation of the composite service, by allowing the parallel execution of certain atomic processes, we have developed a set of algorithms that translate plans (linear or non linear) to composite web services using the Sequence, Split and Split+Join.

The basic algorithm creates a composite service, given a web service graph G=(V, E), as it was defined in Section 3. The process of obtaining a web service graph from the plan is straightforward and due to space limitations, we will not elaborate on that. The output of the algorithm is either a composite construct of the form sequence(c_1, c_2), or split(c_1, c_2,.., c_n), where c_1 to c_n are either NULL or composite constructs. The next step is to replace the Join construct with Split+Join wherever this is possible. This requires a search in all possible pairs of Split arguments, in order to find a common ending part. The last step is to simplify the composite service by removing NULLs and constructs with single arguments. More information concerning the methodology for the representation of composite services can be found in [12].

5 Conclusions and Future Work

This paper presents the semantic awareness issues on several aspects of the automated web service composition process in the PORSCE II framework, an integrated system which exploits planning in order to approach the automated web service composition problem. Semantic awareness, facilitated by the semantic analysis of the OWL-S descriptions of the web services and ontologies, enables semantic enhancement of the planning problem. Moreover, it enables approximations when no exact solutions can be found, and it permits semantic composite service modification and representation.

Future goals include the extension of the system in order to deploy the produced composite services, through OWL-S execution systems such the OWL-S Virtual Machine [10], and automatically acquire feedback, which can be utilized to partially automate the service replacement procedure. Another goal concerns the exploration of the possibility to accelerate the composition process by asserting the produced OWL-S profiles in the base of the available atomic services. Finally, integration with VLEPPO [4] is a promising direction for visual design of web service composition.

References

1. OWLS-TC version 2.2 revision 1, http://projects.semwebcentral.org/ projects/owls-tc/
2. M. Ghallab, A. Howe, C. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld, D. Wilkins, "PDDL -- the Planning Domain Definition Language". Technical report, Yale University, New Haven, CT (1998).
3. OWL-S 1.1. http://www.daml.org/services/owl-s/1.1/
4. O. Hatzi, D. Vrakas, N. Bassiliades, D. Anagnostopoulos, I. Vlahavas, VLEPpO: A Visual Language for Problem Representation, PlanSIG 2007, pp. 60 – 66.
5. SAWSDL, http://www.w3.org/2002/ws/sawsdl/
6. E. Sirin, B. Parsia, D. Wu, J. Hendler and D. Nau, 2004. HTN planning for web service composition using SHOP2. Journal of Web Semantics, 1(4) 377–396.
7. M. Klusch, A. Gerber, M. Schmidt: Semantic Web Service Composition Planning with OWLS-XPlan. AAAI Fall Symposium on Semantic Web and Agents, USA, 2005.
8. J. Rao and X. Su, 2004. A Survey of Automated Web Service Composition Methods. LNCS, Vol. 3387/2005, Springer, p. 43-54.
9. S. Dustdar, W. Schreiner, 2005. A survey on web services composition, Int. J. Web and Grid Services, Vol. 1, No. 1, pp.1–30.
10. M. Paolucci, A. Ankolekar, N. Srinivasan and K. Sycara, "The DAML-S Virtual Machine," In Proc. of the ISWC 2003, FL, USA, pp 290-305.
11. O. Hatzi, G. Meditskos, D. Vrakas, N. Bassiliades, D. Anagnostopoulos, I. Vlahavas, "Semantic Web Service Composition using Planning and Ontology Concept Relevance", IEEE / WIC / ACM Int. Conference on Web Intelligence (WI'09), Milan, Italy, 2009.
12. O. Hatzi, D. Vrakas, N. Bassiliades, D. Anagnostopoulos, I. Vlahavas, “The PORSCE II Framework: Using AI Planning for Automated Semantic Web Service Composition”, *Technical Report TR-LPIS-303-09 (http://lpis.csd.auth.gr/publications/hatzi_porsceII.pdf)*, LPIS, Dept. of Informatics, Aristotle University of Thessaloniki, Greece, 2009.
13. G. Mentzas & A. Friesen (editors), “Semantic Enterprise Application Integration for Business Processes: Service-Oriented Frameworks”, Business Science Reference, 2009.