KNOWLEGDE DYNAMICS IN SEMANTIC WEB SERVICE COMPOSITION FOR SUPPLY CHAIN MANAGEMENT APPLICATIONS[1]

Ion SMEUREANU[2]
PhD, University Professor, Department of Economic Informatics,
Dean of Faculty of Cybernetics, Statistics and Economic Informatics,
University of Economics, Bucharest, Romania

E-mail: smeurean@ase.ro

Andreea DIOSTEANU[3]
PhD Candidate, University Instructor, Department of Economic Informatics,
University of Economics, Bucharest, Romania

E-mail: andreea.diosteanu@ie.ase.ro

Liviu Adrian COTFAS[4]
PhD Candidate, University Instructor, Department of Economic Informatics,
University of Economics, Bucharest, Romania

E-mail: liviu.cotfas@ase.ro

Abstract: *Semantic Web Service technology can play a vital role in today's changing economic conditions, as it allows business to quickly adapt to market changes. By combining individual services into more complex systems, web service composition facilitates knowledge dynamics and knowledge sharing between business partners. The proposed framework uses a semi-automatic approach as manual web service composition is both time-consuming and error-prone.*

Key words: *semantic web service composition; multi agent systems; fractal theory; location based application; supply chain management*

1. Introduction

Nowadays, the enterprise environment is heavily influenced by the evolution of the IT&C technologies. These changes mainly influence the collaborative economic environment the dynamics of knowledge management and increase the level of competition at a global scale. Therefore, adapting and efficiently taking advantage of these changes represents a real challenge for the top and medium management. The enterprise steering committee has to permanently by updated with the latest discoveries so that to obtain maximum satisfaction.

The 21st century, knowledge management has changed a lot due to software and hardware progress. As a consequence, we can affirm that one of the main features of knowledge management is its dynamism. As part of the enterprise environment, knowledge represents added value and contributes as a key factor in assuring sustainable economic growth and increased profit levels. An intelligent management strategy has to be aware of all these changes and to take action so that to maximize the effective use of knowledge. In order for them to gain market shares and to be competitive it is very important to have efficient knowledge management strategies and to be able to take advantage of the new technologies that are continuously emerging. Furthermore, because companies' profitability is mainly influenced by the manner in which their products or services are perceived by customer we can state the fact that the growth and implicitly the investment strategies become product and customer oriented structures. In the context of dynamic business, maximizing and optimizing business performance is a critical requirement for profitability.

The modern economic behaviors are the most accurate and visible proof of the impact the knowledge dynamics has over the traditional types of needs and opportunities. Moreover, all these transformations that occurred are closely connected and have a great influence over the globalization trend the economy is heading towards. The global market can be characterized by increased levels of interoperability that reflect into the integration of multiple and different information systems which are able to share, manipulate and combine knowledge so that to facilitate the enterprise (or generally speaking organizations) collaboration process.

Having given all the above facts, we can conclude that the very frequently used term "integrated company" is not actual any more. Presently, it is substituted by collaborative information systems composed of business networks that are linked to independent partners that provide individual services and goods.

Knowledge management in such complex business and information structures can be defined as an approach, strategically targeted, so that to motivate the members of an organization to develop and use their cognitive capacities, sources of information, experience and abilities by subordinating their own objectives to the overall objectives. In the organizational environment, knowledge is derived from the information that is processed by those who have the capacity to effective action, by assimilation and mainstream understanding, followed by operationalizing the given contexts (Dragomirescu, 2001).

This paper presents a software agent based framework for modeling and enhancing the performance of knowledge management and increasing its level of dynamism by facilitating enterprise interoperability with the help of an automatic web service composition module.

The first section of the article consists of a short literature review so that to establish the place of our research in the current international research trends. And also draws a parallel among different automatic semantic web service composition solutions. In the future sections we enlarge upon the technologies used for implementing the solution, the architecture of the proposed framework. Furthermore, we will present a case study application that we developed to validate the agent based search module of our architecture.

2. Current SWC solutions Analysis

In order to implement interoperable and collaborative applications, the current research trend is heading towards determining the most suitable model for automatic or semi-automatic composition of web services. There are many proposed software solutions architectures for web service compositions, however up to now none of them has been fully implemented in the real business world due to various drawbacks:

- lack of service availability and accessibility;
- large data volumes caused by numerous service descriptions;
- increased search time in web service directory;

The syntactic description of web service is not sufficient for the automation of the web service composition and discovery process. A more detailed semantic description which can annotate the request on the basis of common semantics is needed.

Collaborative and knowledge based applications are to be composed of a set of semantically annotated web services and no longer be implemented separately. Therefore, the composition of semantic web services has enormous potential in improving the integration and collaboration in a wide variety of applications: business-to-business, location based services, supply chain, etc.

Semantic web service composition mainly consists of designing and implementing complex business logic workflows by organizing semantic annotated web services so that to obtain a single semantic complex web service. In order to enable composition, web service functionalities have to be described in detail by using either semantic or functional annotation. In this manner extra information about their main function or about the way the services behave is provided different from the classical syntactic web service annotation. For further information regarding the functionality of a service such as preconditions, conditions, effects, etc. the RDF description language is used. The RDF format uses semantic elements of ontologies that have to be previously established depending on the application type.

By composing services not only that certain components may be used, but processes can be integrated into applications by modeling the business flow. Automated web services composition techniques can be classified into two categories: static and dynamic. (Chakraborty and Joshi 2001).

The static web service composition technique refers to the fact that business flow designers or business analysts manually implement the composition by predefining business processes and by describing the interaction of their web services' components. Dynamic composition of services is not based on such predefined processes. It is rather based on existing web service retrieval and dynamic assembly in order to meet the initial demanding on the semantic content used in their annotation.

The traditional means for describing web services' functionality do not have enough semantic information to be used in the composition. The new trend in web services development can be characterized by the use of ontologies in order to achieve a complete description of them. This type of semantic annotated web services is called semantic web services (Mcilraith, Son and Zeng 2001). The semantic web services composition problem also focuses on the automatic and flexible discovery of services.

The recent research in semantic web services are largely focused on handling requests for task-oriented services and on obtaining information in distributed environments such as the internet information.

One of the first papers (Zhang, et al. 2008) which tackle with this subject performs a detailed analysis to determine the integrated applications' pattern for modeling business processes within enterprises and introduces a semi-dynamic module for automatic composition of such semantic web services. In order to achieve this objective, the authors use an integrated applications' ontology in the field of modeling business processes within enterprises to provide not only basic semantic concepts, but also concepts and terminologies that are used to describe services and to define abstract business processes.

Abstract business processes represent the static part of the semi-dynamic semantic web services composition and it is dynamically attested at runtime through the automatic discovery process. Such a method combines the advantages of static and dynamic composition of services and carries out the semantic based integration for business processes modeling applications. One of the major problems the automatic service composition faces is related to assembling individual services based on their functional specifications to create added value and satisfy a particular service request.

Most approaches to automatic composition of services are based on artificial intelligence planning techniques (Thakkar, et al. 2002), (Mcilraith and Son, 2002), (Wu, et al. 2003). All these techniques require that all relevant service descriptions are loaded into a reasoning engine. Due to the very large number of service descriptions and the weak link between suppliers and consumers, services are indexed in specific directory. The problem occurs when loading a directory that contains such a large volume of data. To solve this problem it is necessary that the planning algorithms should be changed so that only the relevant descriptions to be dynamically extracted from the directory during the composition process (Constantinescu 2004), (Constantinescu, 2005).

In such an approach, a single service composition involves many complex queries of the associated indexed directory. For example, if composition algorithms such as "forward chaining" are being used, each query processing up to 20% of data stored in the directory even if it has an optimized index structure (Constantinescu, 2005). Taking into account that such a directory is a public resource where bottlenecks may occur the problem becomes more complex and complicated to handle.

In addition to the above presented solutions there are some abstract ones based on Colored –Petri Nets (Qian, Lu and Xie 2007), Petri Nets (Hamadi and Benatallah 2003), mathematical models, etc.

In (Qian, Lu and Xie 2007), the authors focus on automatically synthesizing desired composite service through available services. The first contribution of this paper is a general description model of services. This paper also proposes a Colored Petri Net (CPN) based service model MOAP which indicates the relationship between messages (data) and service behaviors. The second contribution of the paper is an effective technique for automatic service composition.

In (Hamadi and Benatallah 2003) the authors propose a Petri net-based algebra for composing Web services. The operators that are used for composing web services are directly link to Petri nets by expressing their semantics in terms of this particular type of nets. The first result of such link is the fact that every service can be expressed as a Petri Net. The advantage brought up by the use of Petri Nets is related to their ability to simulate workflow patterns, operations and properties of web service composition.

The non-abstract web service composition techniques can be grouped into several guidelines according to the research trends that they follow.

Table 1. Semantic web service composition techniques and their objectives

Web service Composition technique	Objective
AI-planning	Finding a path (action plan, sequence of actions) from the initial state to a preset state (target state).
Chaining techniques	Finding dependencies between services in so that to synthesize a composition plan that matches the request

In (Talantikite, Aissani and Boudjlida 2008) we can see a complex approach to web service composition that combines graph based abstract models, chaining algorithm of expert system and semantic annotations. The authors use an inter-connected network of semantic web services that are semantically annotated with OWL-S. The concepts of the ontology enable them to measure the similarity between the outputs and inputs. The advantage brought by this approach is that the composition algorithm can find several composition plans in only one exploration phase. Afterwards the composition plans are being filtered according to precise quality criteria composed of: similarity, time and memory space quality metrics.

3. Technologies Used for Developing the Business Application Framework

3.1. Semantic Web Services

Web Services are internet based technologies that enable the process of making connections. *Services* represent the elements that are being connected with the help of web services. A service is the endpoint of a connection. Moreover, a service has some type of underlying computer system that supports the connection offered. The combination of services – both internal and external to an organization determines a service-oriented architecture.

Services are platform-independent software entities that can be described, published, discovered, and loosely coupled in different ways. They can perform different and complex functionalities from answering simple requests to executing sophisticated business processes requiring peer-to-peer relationships among multiple layers of service consumers and providers. Furthermore, they have features that permit software reengineering and also software reusability and transformation into network-available service.

Presently, the most highly used web services technologies only provide syntactic descriptions, making it difficult for requesters and providers to interpret or represent statements such as the meaning of inputs and outputs or applicable constraints.

As it was stated in the first section, enterprise interoperability and knowledge dynamics represent some of the most important guidelines in developing efficient business flow modeling application. Due to this fact, a lot of research was carried on in this domain. The current trend is to determine the most suitable applications' architectures that fulfill the previously mentioned requirements.

Such applications should be characterized by:

- an increased level of platform-independence for each component;
- interoperability;
- scalability ;

- flexibility;

Web services are software applications that are characterized by these features and are available in the distributed environment of the Internet and are based on XML (eXtensible Markup Language). The functionality of web services is usually presented in a syntactic manner by using standards like: UDDI (Universal Description, Discovery and Integration), WSDL (Web Service Description Language), SOAP (Simple Object Access Protocol).

UDDI is a virtual registry that exposes information about Web services.

WSDL provides a XML based model format for describing web services functionality. WSDL represents a syntactic interface for web services. The description provided by WSDL is separated into two parts: an abstract description of functionality and a particular one that illustrates the details of the web service ("how" certain behaviors are being implemented and "where). WSDL describes only the syntactic interface of Web services.

SOAP is an XML based protocol that is used to exchange structured information in a decentralized and distributed environment. The role of the XML is to define an extensible framework of messages. With the help of such a framework messages, that are to be exchanged through other related protocols, are being constructed. SOAP is not related to any particular programming model and is independent from any specific semantics (Gudgin, et al. 2003).

However, the syntactic description does not provide all the necessary information about a service especially for the web service composition process. For this reason, the semantic annotated web services are used. In order to achieve this objective, they provide functionalities for creating, managing and transforming semantic mark up. The semantic annotations are considered to be conditions and effects of web services and are explicitly declared in the Resource Description Format (RDF) using terms from pre-agreed ontologies. Consequently, it enhances the ability of smart agents to understand, transform and deliver messages over the web.

3.2. Semantic Web Service Composition

The main goal for using semantic web services is that to increase the possibility of automated service discovery, composition invocation and monitoring over the web and in this manner to assure interoperability and collaboration between different business flow modeling applications.

The semantic web community is conducting complex studies in order to determine the most efficient methods for synthesizing web services' complex behaviors and determining an universal semantic representation for the transformation that occur. In this way, web services will be easier to discover on the internet by specific subscribers and also it will be easier to assure web service composition and interaction in Service Oriented Architecture based solutions.

The applications that implement web service composition have a SOC (Service Oriented Computing) architecture that is based on SOA. This type of architecture uses services to support the development of rapid, low-cost, interoperable, evolvable and distributed applications.

As it is stated in (Papazoglou, et al. 2007), according to SOC research road map, SOC provides a logical separation of functionality into three planes:

- service foundations at the bottom

- service composition
- service management and monitoring

This logical stratification is based on the need to separate:

- basic service capabilities provided by a middleware infrastructure and conventional SOA from more advanced service functionality needed for dynamically composing services,
- business services from systems-centered services
- service composition from service management.

3.3. Multi-agent Systems

Multi-agent systems are formed by a certain number of agents that interact with each other. The major advantage of such systems consists of the fact that simple individual behaviors combine into complex ones. Furthermore, another important feature of multi agent systems refers to their ability of decomposing complex problems into more easily manageable sub problems. According to (Bodea and Mogoş 2007), this idea can be applied in many research domains such as for decomposing complex based geospatial problems or supply chain management problems: negotiations, discovery, offer analysis etc. Moreover, they can be used to complex data mining in logistic applications.

In (Genong, et al. 2009) it is illustrated that agents can communicate and interact with each other through ontology language which stands for communication languages. Agents can be managed and discovered through a centralized directory, peer-to-peer discovery, or hybrid mechanism. Agent mobility provides a mechanism to extend stabilities and sustainability of semantic web services in a distributed environment.

One of the major uses of multi-agent systems is related to manufacturing companies. In this case, agents are used to gather information over the web and to transform it into knowledge by adding value. However, if agents are used without combining with semantic web services technology, they fail to respond to the continuously changing business environment in the nowadays knowledge driven society. According to (Weiming, et al. 2007), the failure is associated to the fact that they function on a predefined agreement without being flexible. On the other hand, pure web-based technologies, including web services, cannot fulfill the needs of virtual enterprises applications, because: they do not offer the possibility to automatically discover corresponding services at run time. Also, web service description offers only a technical presentation of the features offered and not a semantic one. Last but not least the description of business processes and security features implemented by using web services are not very reliable because this domain is still in an incipient phase.

Intelligent software agents have been used in enterprise independent software systems integration process, not only to assure an approach for functional integration, but also to facilitate the use of business intelligence and collaboration among enterprises for their communication, interaction, cooperation, pro-activeness, and autonomous intelligent decision making.

In order to achieve the objectives of the current enterprise interoperability trend, we propose a framework that combines web services and software agents so that to provide an efficient service discovery, retrieval, composition and integration. This paper proposes an agent-based service-oriented integration architecture, where enterprise Web services are dynamically discovered using agent behaviors and specific ontology for communication and retrieval.

Multi-agent systems are closely connected to web service technology because they represent interoperable, portable and distributed solutions. Agents and web services may be related in different ways: agents use web services, web services are in fact agents or agents are composed of, deployed as, and dynamically extended by web services (Martin, et al. 2005).

4. Web Service Composition Framework

The proposed framework combines the template-based and AI Planning web service chaining approaches. Based on the composition request and user experience, both automatic and semi-automatic web service composition can be used. In order to facilitate the composition process, web services are managed by intelligent agents. This transforms the RESTful web services into Stateful entities (OASIS 2005) that can maintain their state between calls. Using many intelligent agents, each following its own goal during the composition process helps distribute the composition effort and also avoids performance bottle-necks. All web-services must be previously annotated using OWL-S in order to identify their purpose. While WSDL (Booth and Canyang 2007) provides a syntactic description containing information regarding the structure of the input and output parameters, semantics provide a description of what the web service actually does.

4.1. Framework Components

Warping one or more web services as an intelligent agent is also known as agentification (Yang, et al. 2004). Not only web services, but also legacy applications can be "agentificated", thus reducing the costs of implementing the proposed framework by using existing software.

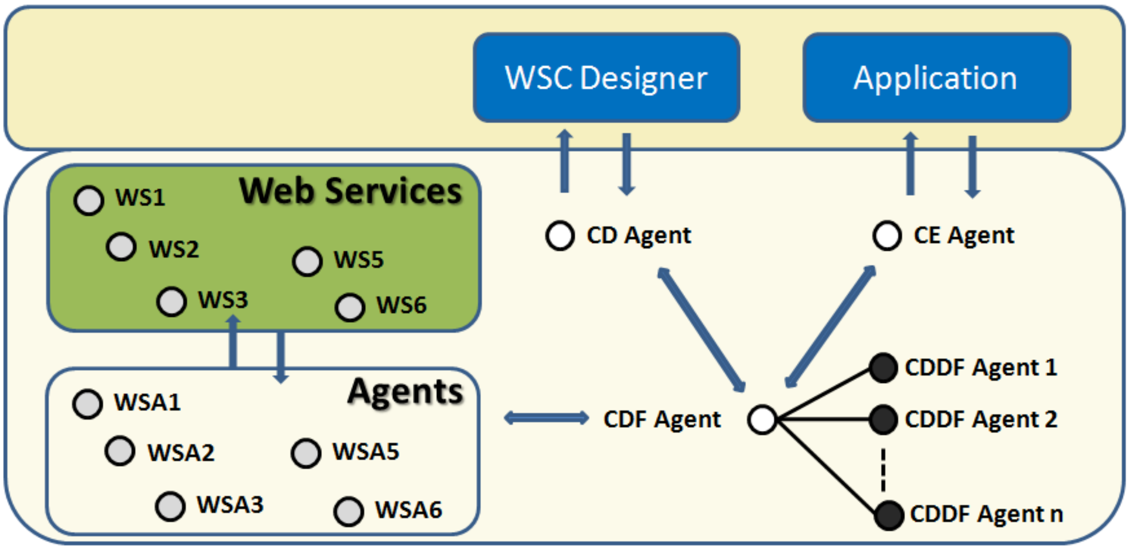

Figure 1. Web Service Composition Framework

The main components and relations between them can be seen in Figure 1.

Agents that can execute a specific task can be found using the Composition Directory Facilitator Agent (CDF) and the several domain specific CDDF. All agents in our platform are registered with at least one CDDF agent. In order to find a web service capable of completing a specific task is need, the CDF agent will be first queried to identify the corresponding CDDFs, which in turn allow searching for agents. For a distributed implementation, multiple CDF agents can be run on different machines.

The Composition Design (CDA) and Composition Execution (CEA) Agents support the design and execution of web service chains. By assigning individual agents to each request, we both improve scalability and avoid creating a performance bottleneck.

The Web Service Composition (WSC) Designer allows creating and editing web service chains in an interactive manner. It also offers the possibility to test and validate the created web service chains as well as displaying several statistics regarding estimated duration and availability.

The communication between agents is implemented using FIPA compliant ACL messages. The proposed framework was implemented using JADE (Telecom Italia n.d.) (Java Agent Development Framework) and JENA (Hewlett-Packard n.d.) frameworks.

4.2. Fractal Web Service Composition

Web service composition implies finding a sequence of services that when called helps achieve a certain goal. In our approach, existing web service chains can be combined in a fractal like manner to easily create new and more complex web service chains. Similar to using components in classic programming, this approach has several advantages like the reduction of the development time and a higher QOS (Quality of Service) as a result of using already tested web services. Also, the solution becomes more adaptable, as, if needed, web services can be replaced with smaller effort.

Implementing a framework that eliminates the need to manually bind available services is particularly important in today's changing economic conditions, as it allows business using the proposed framework to quickly adapt to market changes without waiting for internal IT development projects or for long software vendor release cycles (Bussler 2007).

All web service chains in our framework are stored using WS-BPEL (Web Service Business Process Execution Language) (OASIS 2007) and can effectively be used as building blocks to create new service chains.

The design of the newly created chain starts in the WSC (Web Service Composition) Designer Module of our framework. The user has the possibility to either create a completely new chain or to modify an existing one. From the user's perspective the web service chain is composed from a succession of actions or sub-goals that must be performed in order to achieve a certain goal (Qian, Lu and Xie 2007). The user must specify the available input data and the requested output information.

For each action defined in the interface the following steps will be performed:

Step 1 – Action Matching: The CDA agent semantically queries the CDF and CDDF agents on all machines searching for Web Service Agents (WSA) that can perform the requested action and begins a parallel negotiation with all found agents. If no matching WSA can be found, the CDA agent will request the user to decompose the action into more elementary actions and will repeat the first step.

Step 2 – Input parameters: Every candidate WSA agent compares it's input data with the internally mapped web service or web service chain input parameters. In case any input parameters don't match, the WSA agent will itself query the CDF and CDDF agents in order to find an agent or a chain of agents capable of performing the required transformation. If still needed the found agents can themselves repeat the procedure. Thus, the web service composition is performed in a manner similar to fractal theory, as each WSA agent can recursively call other agents. In order to limit the search space a maximum number of agents used to model an action can be specified.

Step 3 – Output parameters: Each caller agent, including the CDA verifies the correspondence between the called WSA web service output parameters and the requested output parameters. In case any output parameters don't match, an agent capable of performing the required transformation will be searched using the CDF and CDDF agents similar to Step 2.

Step 4: The best WSA agent or WSA agent chain is selected based on estimated execution time and availability statistics.

The final chain is stored so it can either be called directly or incorporated in new chains. A corresponding WSA is created and registered with the CDF and CDDF agents.

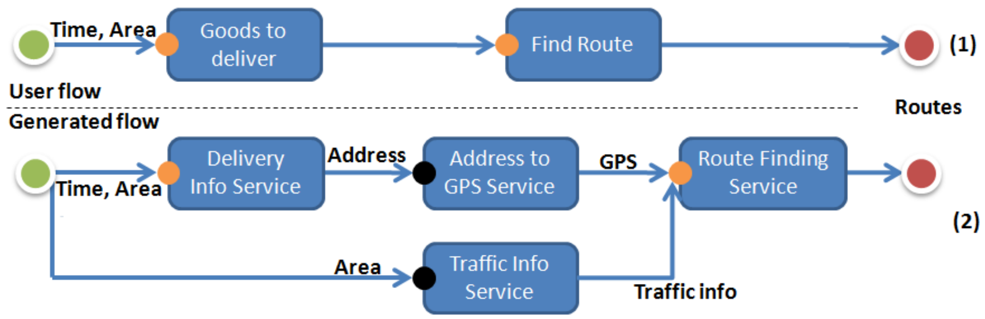

Figure 2. Web service composition using individual services

Figure 2 presents a composition scenario for location based services and supply chain systems in which the user's goal is to create a web service chain that finds the best route to deliver goods to customers. Based on his experience the user adds to actions: identifying the goods to deliver in a specific area at a specific time and finding a route (1). Based on this information, in Step 2, the CDA agent discovers the matching services "Delivery Info Service" and "Route Finding Service". "Address to GPS Service" is added to convert between "Delivery Info Service" output and "Route Finding Service" input type. "Traffic Info Service" supplies additional needed input data for the route finding service (2).

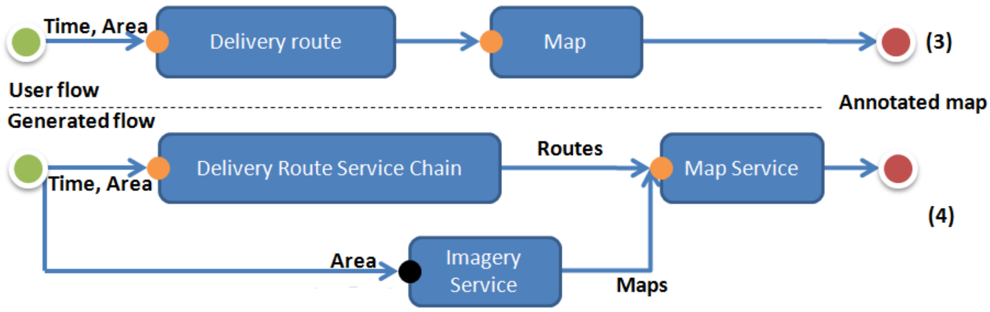

Figure 3. Web service composition using previously defined web service chains

In case the route should also be displayed on a map, a new web service chain can be created as shown in Figure 3. Based on users added actions (3), the framework selects the previously defined chain and the "Map Service" capable of creating annotated maps based on information and map images. "Imagery Service" is automatically added to supply the parameters required by the Map Service.

Conclusions

Manual web service composition is both time-consuming and error-prone. The proposed framework allows the implementation of distributed semantic web service chains by using a semi-automatic approach. Organizations implementing such a solution will be able to better cooperate and share their knowledge. Moreover, such systems can easily be extended with new web service as they become available.

References

1. Bodea, C. and Mogos, R. **An Electronic Market Space Architecture Based On Intelligent Agents And Data Mining Technologies**, Informatica Economica Journal XI, no. 4, 2007, pp. 115-118
2. Booth, D. and Canyang, L. **Web Services Description Language (WSDL) Version 2.0 Part 0: Primer**, June 26, 2007, http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626 (accessed January 12, 2010)
3. Bussler, C. **The Fractal Nature of Web Services**, Computer, 40, 2007, pp. 93-95
4. Chakraborty, D. and Anupam, J. **Dynamic Service Composition: State-of-the-Art and Research**, Department of Computer Science and Electrical, Maryland University, Baltimore, 2001
5. Constantinescu, I. **Flexible and efficient matchmaking and ranking in service directories**, IEEE International Conference on Web Services (ICWS-2005), Florida: IEEE, 2005, pp. 5-12
6. Constantinescu, I. **Large scale, typecompatible service composition**, IEEE International Conference on Web Services (ICWS-2004), San Diego: IEEE, 2004, pp. 506-513
7. Dragomirescu, H. **Organizatii bazate pe cunoastere**, Research Institute for Artificial Intelligence, 2001, http://www.racai.ro/ INFOSOCProject/Dragomirescu_st_g06_new.pdf (accessed March 2009)
8. Genong, Y., Liping, D., Wenli, Y., Peisheng, Z. and Peng, Y. **Multi-Agent Systems for Distributed Geospatial Modeling, Simulation and Computing**, in "Handbook of Research on Geoinformatics", Pennsylvania: Information Science Reference, 2009, pp. 196-205
9. Gudgin, M., Hadley, M., Mendelsohn, N. and Moreau, J.-J. **W3C Recommendation**, W3C Recommendation, June 2003, http://www.w3.org/TR/soap12-part1/ (accessed January 2010)
10. Hamadi, R. and Benatallah, B. **A Petri net-based model for web service composition**, in "Proceedings of the fourteenth Australasian database conference", Adelaide, 2003, pp. 191-200
11. Martin, D., Burstein, M., McIlraith, S., Paolucci, M. and Sycara, K. **OWL-S and Agent-Based Systems**, In Extending Web Services Technologies: The Use of Multi-Agent Approaches, New York: Springer, 2005, pp. 53-77
12. Mcilraith, S. and Son, C. T. **Adapting Golog for composition of semantic web services**, 8th International Conference on Principles and Knowledge Representation and Reasoning (KR-02), San Francisco: Morgan Kaufmann, 2002, pp. 482–496
13. Mcilraith, S., Son, T. C. and Zen, H. **Semantic Web services**, Intelligent Systems, March 2001, pp. 46-53

14. Papazoglou, M. P. , Traverso, P., Dustdar, S. and Leymann, F. **Service-Oriented Computing: State of the Art and Research Challenges**, Computer 40, no. 11, November 2007, pp. 38-45
15. Qian, Z., Lu, S. L. and Xie, L. **Colored Petri Net Based Automatic Service Composition**, Asia-Pacific Service Computing Conference, The 2nd IEEE. Tsukuba Science City: IEEE, 2007, pp. 431-438
16. Talantikite, N. H., Aissani, D. and Boudjlida, N. **Semantic annotations for web services discovery and composition**, Computer Standards & Interfaces, 2008, pp. 1108–1117
17. Thakkar, S., Knoblock, C., Ambite, L. J. and Shahabi, C. **Dynamically composing web services from on-line sources**, AAAI-2002 Workshop on Intelligent Service Integration, Edmonton, 2002, pp. 1–7
18. Weiming, S., Qi, H., Shuying, W., Yinsheng, L. and Hamada, G. **An agent-based service-oriented integration architecture for collaborative intelligent manufacturing**, Robotics and Computer-Integrated Manufacturing, no. 23 (2007), pp. 315-325
19. Wu, D., Parsia, B., Sirin, E. and Hen, J. **Automating DAML-S web services composition using SHOP2**, 2nd International Semantic Web Conference (ISWC-2003), Sanibel Island: Springer, 2003, pp. 195–210
20. Yang, H., Chen, F., Guo, H. and Xu, B. **Agentification for web services**, COMPSAC, New York, 2004, pp. 514–519
21. Zhang, K., Xu, R., Zhang, Y., Sai, Y. and Wang, X. **An Ontology Supported Semantic Web Service composition Method in Enterprise**, IEEE International Multi-symposiums on Computer and Computational, IEEE, 2008, pp. 222-227
22. * * * (W3C), World Wide Web Consortium, **OWL Web Ontology Language**, February 10, 2004, http://www.w3.org/TR/owl-features (accessed February 2010)
23. * * * Hewlett-Packard, **Jena Semantic Web Framework**, n.d. http:// jena.sourceforge.net (accessed December 3, 2009)
24. * * * OASIS, **Web Services Business Process Execution Language Version 2.0**, April 2007, http://docs.oasis-open.org/wsbpel/2.0/ (accessed January 8, 2010)
25. * * * Telecom Italia, **Java Agent Development Framework**, n.d. http://jade.tilab.com/ (accessed October 19, 2009)
26. * * * OASIS, **Stateful Web Services**, March 7, 2005, http://xml.coverpages.org/stateful WebServices.html (accessed January 12, 2010)

[1] **Acknowledgements**
This article is a result of the project „Doctoral Program and PhD Students in the education research and innovation triangle”. This project is co funded by European Social Fund through The Sectorial Operational Programme for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies.

[2] Ion SMEUREANU has graduated the Faculty of Planning and Economic Cybernetics in 1980, as promotion leader. He holds a PhD diploma in "Economic Cybernetics" from 1992 and has a remarkable didactic activity since 1984 when he joined the staff of Bucharest Academy of Economic Studies. Currently, he is a full Professor of Economic Informatics within the Department of Economic Informatics and the dean of the Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies. He is the author of more than 16 books and an impressive number of articles. He was also project director or member in many important research projects. He was awarded the Nicolae Georgescu-Roegen diploma, the award for the entire research activity offered by the Romanian Statistics Society in 2007 and many others.

[3] Andreea DIOSTEANU has graduated the Faculty of Economic Cybernetics, Statistics and Informatics in 2008 as promotion leader, with an average of 10. She is currently conducting research in Economic Informatics at Bucharest Academy of Economic Studies and she is also a pre-Assistant within the Department of Economic Informatics and .NET programmer at TotalSoft. During the bachelor years she par-ticipate in many student competitions both at national and international level obtain-ing a lot of first and second prizes. The most important competitions she was finalist in were Microsoft International Imagine Cup Competition, Software Design section (national finalist); Berkley University and IBM sponsored ICUBE competition were she qualified for the South Eastern Phase-Novatech. Furthermore, she also obtained the "N.N Constantinescu" excellence scholarship in 2007-2008 for the entire student research activity.

[4] Liviu COTFAS is a Ph.D. student and a graduate of the Faculty of Cybernetics, Statistics and Economic Informatics. He is currently conducting research in Economic Informatics at Bucharest Academy of Economic Studies and he is also an Academic Preparator within the Department of Economic Informatics. Amongst his fields of interest are geographic information systems, genetic algorithms and web technologies.