# Agent Based Knowledge Management Solution using Ontology, Semantic Web Services and GIS

Andreea DIOŞTEANU, Liviu COTFAS
Economic Informatics Department,
Academy of Economic Studies, Bucharest, Romania
andreea.diosteanu@ie.ase.ro, liviu.cotfas@ie.ase.ro

*The purpose of our research is to develop an agent based knowledge management application framework using a specific type of ontology that is able to facilitate semantic web service search and automatic composition. This solution can later on be used to develop complex solutions for location based services, supply chain management, etc. This application for modeling knowledge highlights the importance of agent interaction that leads to efficient enterprise interoperability. Furthermore, it proposes an „agent communication language" ontology that extends the OWL Lite standard approach and makes it more flexible in retrieving proper data for identifying the agents that can best communicate and negotiate.*
***Keywords:*** *Automated Service Composition, Ontology, Multi-Agent System, Semantic Web Service, Geographic Information Systems (GIS)*

## 1 Introduction

In the 21st century enterprises are confronting with a continuously changing economic context due to the evolution of IT&C technologies and an increased level of competition at a global scale. As a result, 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.

The new economic behaviors are the main consequence of the transformations that affected the traditional types of needs and opportunities and are required by the globalization trend. The global market can be characterized by increased levels of interoperability that reflect in the integration of multiple and different information systems that are able to share, manipulate and combine knowledge so that to facilitate the enterprise (or generally speaking organizations) collaboration process. The term of integrated company is not actual any more. Nowadays, we are speaking about business networks compose of independent partners that offer to the others their specific services or products. Therefore, we can assume that companies become product and customer oriented structures. In the context of dynamic business, maximizing and optimizing business performance is a critical requirement for profitability.

Knowledge management 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 operationalazing the given contexts [1].

Through this paper we present a software agent based framework's architecture for modeling business flow and enhancing the performance of knowledge management by facilitating enterprise interoperability with the help of an automatic web service composition module.

In the first section of the article we make a short literature review so that to establish the place of our research in the current international research trends. In the future sections we enlarge upon the architecture of the proposed framework, the development of a standard ontology that represents the communication language between agents and some sample ontologies that respect the

proposed OWL (Web Ontology Language) Lite [2] extended format. Furthermore, we will present a case study application that we developed to validate the ontology and the agent based search module of our architecture.

## 2 Multi-agent Systems

Multi-agent systems are formed by multiple 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 [3], 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**.**

[4] Illustrates the fact that agents 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.

**Table 1.** Agent properties important for interactions in a distributed multi-agent system [4]

| Property | Description |
|---|---|
| Reactive | Reaction based on its sense |
| Autonomous | Responses based on its own experiences |
| Rational | maximize its own interest |
| Goal-oriented | Pursue a goal |
| Temporally continuous | Deals with continuous |
| Mobile | Able to transport itself from platform to platform |
| Communication | Interactions on another level of abstraction-language |

By analyzing the above presented properties we can depict that there are many domains in which multi agent systems can be used to model pure knowledge, knowledge transformation and exchange flow. Some examples of such implementations are illustrated below.

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 [5], 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.

The current trend in enterprise evolution is heading towards efficient knowledge representation, storing, manipulation and interoperability. Due to enterprise knowledge sharing and knowledge based collaboration we can speak about virtual organizations.

In [6] virtual organizations mean, generally, a grouping of legally distinct or related enterprises coming together to exploit a particular product or service opportunity, collaborating closely whilst still remaining independent and potentially competing in

other markets or even other products/services in the same market. Virtual organizations emerge from innovation ecosystems, where enterprises have the ability and expectation to collaborate closely with one another: collaboration is a key.

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 selection, retrieval, composition and integration. This paper proposes an agent-based service-oriented integration architecture, where enterprise Web services are dynamically discovered on the Internet using agent behaviors and specific ontology for communication and retrieval. In order to demonstrate our findings we implemented a software application for enterprise agent retrieval developed in JADE (Java Agent Development Framework) [7] by using Jane [8] framework for user built in ontology.

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 [9].

During the process of transforming web services into agents or boxing them into agents, the web service description may be retrieved automatically by examining its description in WSDL (web service description language). Converted agent description can be extended with specific ontologies. Agents can be connected to web services by using a gateway agent which keeps two directories – one that serves the agent world and another that serves the web service world, called WSIG (Web Service Integration Gateway).

The WSIG has a DF (directory facility) and a UDDI (Universal Description, Discovery, and Integration). An agent can be registered with the WSIG agent and internally mapped to UDDI services, while a web service can be registered with the WSIG agent and internally mapped to DF [4].

## 3 Proposed framework architecture for developing agent based collaborative applications

The framework that we propose is a general development environment for both business and GIS agent based, distributed software applications that enable organizations' interoperability and collaboration. In this article we will present the use of the proposed framework in developing agent based supply chain application for a construction company.

According to [10] the supply chain is a worldwide network of suppliers, factories, warehouses, distribution centers, and retailers through which raw materials are acquired, transformed, and delivered to customers. Supply-chain management is the strategic, tactical, and operational decision making that optimizes supply-chain performance. The strategic level defines the supply chain network; that is, the selection of suppliers, transportation routes, manufacturing facilities, production levels, warehouses, and the like. The tactical level plans and schedules the supply chain to meet actual demand. The operational level executes plans. Tactical- and operational-level decision-making functions are distributed across the supply chain.

To optimize performance, supply-chain functions must operate in a coordinated manner. However, there are certain problems that affect the supply chain flow, most of them unpredictable or hard to discover in an incipient phase without keeping track of the previous experiences in an organized manner. For example, let's assume that we

are dealing with a construction firm that needs certain materials. There are several suppliers and the constructor has to make the right choice. For this, he has to take into consideration the following aspects: price, quality, transportation costs, geographic risks, previous experience with certain suppliers, etc. All this information represents added value for the transaction and transforms into knowledge that can and should be stored in a rule format in the knowledge database.

In recent years, new software architectures for managing the supply chain at the tactical and operational levels have emerged. According to the new trend, the supply chain is perceived as a set of intelligent (software) agents, each responsible for coordinating one or more activities in the supply chain and each interacting with other agents in planning and executing their responsibilities.

An agent is an autonomous software process, targeted to achieve a specific objective and operates asynchronously, communicating and coordinating with other agents as needed.

Having illustrated the above facts, our main objective is to develop an agent based knowledge manipulation framework that facilitates enterprise collaboration and interoperability.

The solution uses the following technologies:

- semantic web and ontology
- multi agent systems
- semantic web services

The architecture of the proposed framework consists of the following modules:

- ontology design component with a user friendly interface
- agent based web service search module
- knowledge database
- semantic web service composition module
- user interface for displaying the results

Below we present each component of the framework's architecture in general terms because it can be used to develop applications in different domains of activity. In the next section we will exemplify the use of the framework by implementing an agent based solution for supply chain management application in a construction company and we will emphasize on the agent discovery module.

1. *The ontology design component*

By using this component users will be able to design their own ontology by using intuitive graphic elements. The standard use for the ontology is extended OWL Lite. Generally, speaking the format of our proposed ontology is described by the following rules:

- all classes are derived from the general class THING
- a class called TRADED_THING may stand for: requested product/service in a supply chain, geographic web services, etc. The RequestedThing class may contain DataProperty elements for describing specific features of the requested object. The use such properties is necessary for "partner agents" discovery
- AREA class that contains the definition of a polygonal geographic area given by its corners' GPS coordinates. To be more specific, this class has a property called GEOGRAPHIC SHAPE. For this property we have several instances containing geographic coordinates.
- DOMAIN class that contains the definition of a particular domain of interest (for example activity domain in a supply chain application). The Domain will also contain instances called ACTIVITY.

By using instances we increase the level of communication flexibility and we can extend the agent search space.

This type of ontology can be used as agents' "communication language" in different types of software agent based application.

As a whole each class represents a specific semantic network, and the relationships between its elements: InstanceOf, SubclassOf, PropertyOf model the human language semantic connections.

2. *The agent based web service search module*

This module is in charge with agent discovery based on a specific requirement that can be depicted from the ontology. This

agent search module inquiries a sort of "yellow pages" service that enables agents to publish one or more services they provide so that other agents can find and successively exploit them.

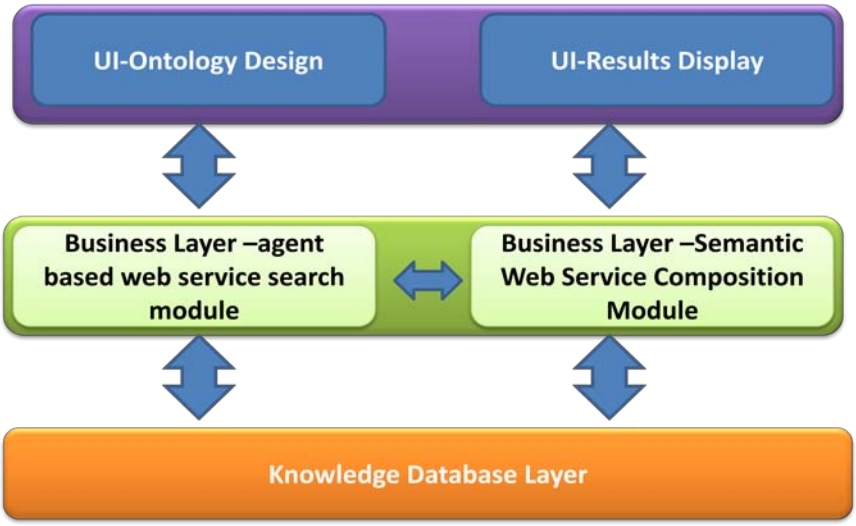


**Fig. 1.** Framework architecture schema

The framework's architecture is presented in the figure 1. Agents are interacting with DF by exchanging ACL messages using a proper content language (the SL0 language) and a proper ontology (the FIPA-agent-management ontology) according to the FIPA specification.

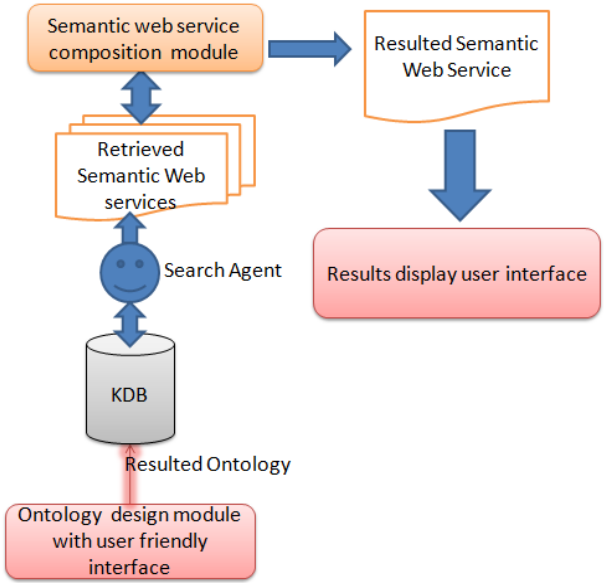


**Fig. 2.** Application flow

In order to implement agents we used JADE a java based platform that provides agent implementation specific methods. In Jade the DF agent inquiry is handled by using jade.domain.DFService class by means of which it is possible to publish and search for services through method calls.

An agent wishing to publish one or more services must provide the DF with a description including its ID, possibly the list of languages and ontologies that other agents need to know to interact with it and the list of published services. For each published service a description is provided including

the service type, the service name, the languages and ontologies required to exploit that service and a number of service specific properties. The DFAgentDescription, ServiceDescription and Property classes, included in the jade.domain.FIPAAgentManagement package, represent the three mentioned abstractions.

*3. The Knowledge database*

The knowledge database stores ontologies and rules for agent discovery. Furthermore, it contains data related to previous experiences so that to facilitate the decision making process and improve decision quality.

*4. The Semantic web service composition module*

This module is in charge with automated semantic web service composition. After identifying the appropriate agents that publish the required services, this module combines them so that to obtain the a required output or finalize a deal when talking about supply chain applications.

*5. User interface for displaying the results*

The interface displays in a user friendly manner the results of web service composition.

A standard working flow by using the framework is presented in figure 2.

## 4 Case Study for Agent based Web Service Search Module

The application that we developed in order to validate the above presented framework and the proposed ontology refers to modeling the supply chain activity for a construction company. Through this article we are going to insist on the development of the agent search module based on a specific ontology format that we designed and that was presented in the previous chapter.

The problem that we modeled is the following: "A construction company desires to buy brick from construction materials suppliers. The construction company is represented by an agent called Customer."

The customer agent will inquire the DF agent so that to find the most suited supplier companies' agents. The query consists of specifying the requested agents type (Construction Material Suppliers), the type of ontology (MyOntology) and the language.

The customer agent's ontology will contain the following information:

- ***Domain*** - contains information related to the activity domain of the Customer agent: construction
- ***Traded Thing*** - contains information related to the company's yield: product type and technical features. For a company that solicits bricks, a technical feature is the maximum supported temperature.
- ***Area*** - contains information regarding the geographic position of the company.

**Sample Customer Ontology code:**

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:j.0="http://constructionfirm.com/"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
  <rdf:Description rdf:about="http://constructionfirm.com/maxTemperature">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#unsignedInt"/>
    <rdfs:domain rdf:resource="http://constructionfirm.com/Brick"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://constructionfirm.com/geogrphiCoordinates">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="http://constructionfirm.com/Area"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </rdf:Description>
  <rdf:Description rdf:nodeID="A0">
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:
cardinality>
    <owl:onProperty rdf:resource="http://constructionfirm.com/maxTemperature"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Restriction"/>
```

```
  </rdf:Description>
  <rdf:Description rdf:about="http://constructionfirm.com/Domain">
    <rdfs:label xml:lang="EN">Domain</rdfs:label>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://constructionfirm.com/Area">
    <rdfs:label xml:lang="EN">Area</rdfs:label>
    <rdfs:subClassOf rdf:resource="http://constructionfirm.com/Brick"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://constructionfirm.com/requestedThing">
    <rdfs:label xml:lang="EN">TradedThing</rdfs:label>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://constructionfirm.com/RequestedBrick">
    <j.0:maxTemperature>150</j.0:maxTemperature>
    <rdf:type rdf:resource="http://constructionfirm.com/Brick"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://constructionfirm.com/Brick">
    <rdfs:label xml:lang="EN">Brick</rdfs:label>
    <rdfs:subClassOf rdf:resource="http://constructionfirm.com/requestedThing"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
</rdf:RDF>
```

On the other hand there are ConstructionMaterialSupplier agents which use the same type of ontology:

- ***Domain*** - contains information regarding their activity
- ***TradedThing*** - stores information about all the products offered by the supplier together with their technical features.
- ***Area*** - stores information about the geographic points of sales. The information is described by using Instances and DataProperty. The points of sale are given like geographic shapes defined using the GPS coordinates of their corners.

The proposed application is implemented in a distributed manner that enables agents running on different platforms to communicate over the internet. The communication between agents is based on OWL ACL messages compliant with FIPA [11] standards.


**Fig. 3.** Adding delivery area from Supplier's Agent GUI

In order to find the best offer a series of steps are required:

**Step1:** The customer agent queries multiple DF agents to retrieve the list of available supplier agents.

**Step2:** The customer agent sends its demand to the supplier agents using the previously described ontology. The demand specifies

the requested delivery addresses using GPS coordinates, the supplier's activity domain, the requested products and there technical features. In our case, the customer will specify that the required product is subClassOf "Brick" and that it should resist to a certain temperature.

**Step3**: The suppliers analyze the customer's demand and respond in case of match. Firstly, each supplier agent checks if his activity domain matches the domain specified in the demand. They go on by comparing the geographic position of the customer and their points of sales covered surface. The available delivery areas are stored as geographic shapes (polygon) using GML (Geographic Markup Language) [12] and can be defined using the GUI (figure 3).

In case of match, customer agents analyze the object of trade technical features. By using the ***inference*** facilities offered by Jena [6] they can determine whether their offers match the yield of the customer.

```
Reasoner reasoner =
ReasonerRegistry.getOWLReasoner();
reasoner =
reasoner.bindSchema(GenerateOntology());
InfModel infmodel =
ModelFactory.createInfModel(reasoner,
ontologyModel);
Resource resource=
infmodel.getResource(NS +
"TradedThing");
```

**Step 4:** The customer agent continues to communicate and negotiate with the agents that positively respond. The semantic web service composition module combines these services and transmits the results of the closed deal to the GUI. The search module work flow is presented in figure 4.

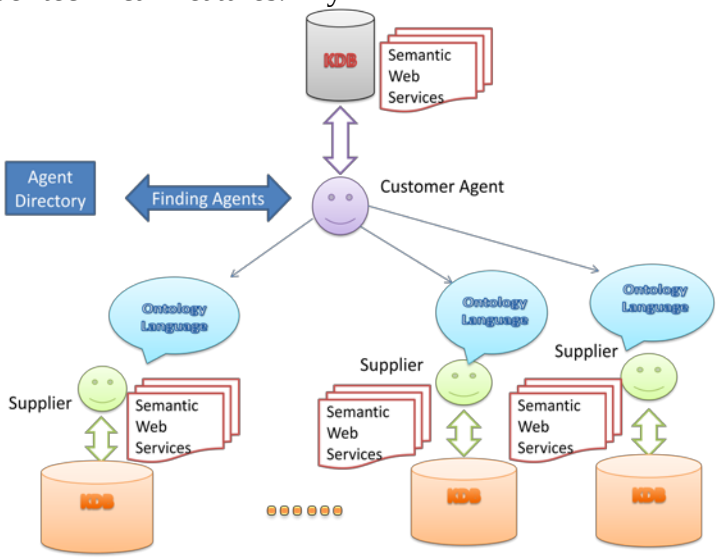


**Fig. 4**. Sample flow for the agent based semantic web service search module in supply chain application

## 5 Conclusions and future work

This article presents a framework's architecture for developing business and GIS applications having as main target assuring an efficient knowledge management by facilitating organization interoperability. In order to achieve this purpose we used semantic web services, agents and ontologies.

In this paper we presented an application for supply chain management developed by using the framework and we insisted more on the agent search module and ontology design. In the future, we will focus on implementing the automatic web service composition module.

## 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 Program for Human Resources Development

2007-2013, coordinated by The Bucharest Academy of Economic Studies.

**References**

[1] H. Dragomirescu, „Organizații bazate pe cunoaştere," *Research Institute for Artificial Intelligence*, 2001, Available at: http://www.racai.ro/ INFOSOCProject/Dragomirescu_st_g06_ new.pdf

[2] World Wide Web Consortium (W3C), *OWL Web Ontology Language*, February 10, 2004. [Online]. Available at: http://www.w3.org/TR/owl-features. [Accessed: September 3, 2009].

[3] C. Bodea and R. Mogos, "An Electronic Market Space Architecture Based On Intelligent

Agents And Data Mining Technologies," *Informatică Economică Journal*, Vol. 11, No. 4, pp. 115-118, 2007.

[4] G. Yu et all, "Multi-Agent Systems for Distributed Geospatial Modeling, Simulation and Computing," *Handbook of Research on Geoinformatics*, Pennsylvania: Information Science Reference, 2009, pp. 196-205.

[5] W. Shen et all, "An agent-based service-oriented integration architecture for collaborative intelligent manufacturing," *Robotics and Computer-Integrated Manufacturing*, No. 23, pp. 315-325, 2007.

[6] Information Society Technologies, "Enterprise Interoperability Research Roadmap Final Version," pp. 4, July 2006.

[7] Telecom Italia, *Java Agent Development Framework*, [Online]. Available at: http://jade.tilab.com/. [Accessed: October 3, 2009].

[8] Hewlett-Packard, *Jena Semantic Web Framework*, [Online]. Available: http:// jena.sourceforge.net. [Accessed: September 3, 2009].

[9] D. Martin et all, "OWL-S and Agent-Based Systems," *Extending Web Services Technologies: The Use of Multi-Agent Approaches*, New York: Springer, pp. 53-77, 2005.

[10] M. Fox, M. Barbuceanu and R. Teigen, "Agent-Oriented Supply-Chain Management," *The International Journal of Flexible Manufacturing Systems*, No.12, pp.165-188, 2000.

[11] Foundation for Intelligent Physical Agents [Online]. http://www.fipa.org. [Accessed: September 12, 2009].

[12] Open Geospatail Consortium, *Geography Markup Language | OGC*, September 9, 2009. [Online]. Available at: http://www.opengeospatial.org. [Accessed: September 17, 2009].

**Andreea DIOŞTEANU** 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.

**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 assistant lecturer within the Department of Economic Informatics. Amongst his fields of interest are geographic information systems, genetic algorithms and web technologies.