Network Management Framework based on Intelligent Agents, Semantic Modeling and Web Services

Ana-Maria GHIRAN
Babeş-Bolyai University Cluj-Napoca,
Faculty of Economics and Business Administration,
Business Information Systems Department
anamaria.ghiran@econ.ubbcluj.ro

During its evolution network management has been approached through a variety of methods, each with advantages and limitations. It is this variety of implemented techniques that makes the network management field very complex with a large diversity of solutions. Although increased difficulty in integrating all these, the expectations are that it could be done in an easier way ("with a mouse click", if possible). Other requirements have also a big influence on network management solutions, like the need for integration in the broader scope of enterprise management. In this paper, a framework for implementing network management is presented that will take into consideration managing the complexity issues, the need for a response in a fast and easy way and the integration within general management objectives. For this, intelligent agents, presentation format through a familiar web interface and semantic assimilation have been investigated.

Keywords: *Semantic Modeling, Autonomic Networks, Network Management, Policy Based Management, Service Based Management, Semantic Services*

1 Introduction

Managing heterogeneous resources and their interoperability with the ever increasing size and complexity of the networks have been facing significant challenges. As a direct consequence, starting with the last decade, as noted by Boutaba et al.[1], it is important for network administrators to be able to use high level directives. More, the network administrators' directives are themselves consequences of other high level requirements from network users, managers, best practices or even legal requirements. But, in most encountered network problems, it is not possible yet for human administrators to specify a request/solution in high level terms and the network to solve/perform it. This is mainly due to the lack of semantic description in the formulated requests but also to the delegation techniques currently used for management task assignments.

Adoption of IT infrastructure in all industry areas has generated a large spectrum of network configurations. There is often the situation when some of these have to work cooperatively (or be managed together) as a service in order to sustain the business objectives of the enterprises. This is a consequence of new user requirements that drive the networks' evolution to a service oriented architecture which is going to be the dominant paradigm for the future applications [2]. Services in network management can have two meanings: first, they are considered a whole set of operations that can be externalized through outsourcing and which are subject to Service Level Agreements (SLA). In this case, management of enterprise networks is getting closer to the management way of telecom networks, where there is a hierarchical structure: element management, network management and service management. In a survey realized by Wallin, 2009 [3] on fifteen major telecom operators, the most important factor that would be considered for their management solution was the management of services.

Second, services can be understood as web services or some web functionality that is used for systems to access and exchange management information. This is an active research and several standards were proposed like Management Using Web Services (MUWS) (http://www.oasis-open.org/specs/#wsdmv1.1) part of WSDM (Web Services Distributed Management) from OASIS, or WS-Management (Web Services for Management) (http://www.dmtf.org/standards/wsman/) from DMTF (Distributed Management Task Force).

Considering the above, a proposed solution will have to include:

- possibility of defining high level statements for network functionalities and for network behavior (defining rules for alignment with business objectives).
- delegation of the management assignments as close to the nodes
- possibility for automatically aggregation of the network functionalities in order to achieve complex management tasks
- using of a standard method for the managed interface to ensure the easy interoperability of network elements.

2 Managing technology through technology

As [7] have identified, the idea of using technology to address the management of systems or networks (technology to manage technology) has been already embraced by many companies in the IT industry that have developed and delivered products based on this concept.

2.1 Autonomic Networks

Among the first that have emphasized the necessity of incorporating the semantic information into network data is Clark et al.[4]. As networks are delivering data they are not aware of its meaning or its purpose and if some combination of events prevents data to reach its destination, the edge may recognize there is a problem but the core cannot tell that something is wrong. [4] They proposed a „knowledge plane" that represents a unified approach over the data layer and the control layer and that allows the knowledge acquisition from distributed sources and in the same time a high level (global) view over the requirements of users, network applications or network designers [4]. Their work suggests a new kind of network which is aware of itself and its surroundings and it would be able to recognize malfunctions, explain them or suggest ways of solving them and finally it should fix the problems itself. This approach is one that contributed to a brand new research direction: towards self management networks.

The concept autonomic network is used to express the fact that a network can operate without or with little human intervention [5] [6]. They define autonomic network is an extension of autonomic computing, which attempts to manage the operation of individual constituencies of a network.

Autonomic computing or self management as defined by IBM [6] is a projection of four of the functional requirements of the OSI network management, FCAPS (Fault Management, Configuration Management, Accounting Management, Performance Management and Security Management): self- heal (self-recovery) for fault management, self-configuring for configuration management, self-optimize for performance management, self-protect for security management.

The research in autonomic computing has already been concretized in a few architectures: FOCALE (Foundation-Observation-Comparison-Action-Learn-rEason) [6], IBM Architecture [7], ANEMA (Autonomic Network Management Architecture) [8].

FOCALE has been proposed by [5] [6] who sustained the need for ontological modeling to capture knowledge relating to network capabilities, environmental constraints, business goals and policies together with reasoning and learning techniques in order to realize autonomic network management. The ultimate goal is in the "capability of network entities to self govern their behavior *within the constraints of business goals* that the network as a whole seeks to achieve". These will enable humans to focus more on business logic and less on low-level device configuration processes. Their approach assumes that any managed resource (which can be as complex as a whole network) can be transformed in an Autonomic Component by em-

bedding an Autonomic Manager. More Autonomic Computing Elements are grouped into Autonomic Management Domains and then into Autonomic Management Environments. [6]
IBM proposed architecture [7] has a layered design in which the upper layers contain Autonomic Managers (AM) and the lowest layer include managed resources. The management interfaces for these resources is recommended to be implemented with the WSDM standard so the managed devices can be access as web services. All layers share a knowledge source, which provides a common domain model.
In ANEMA architecture [8] the high level objectives formulated by network administrators are captured and expressed as Utility Functions Policies. Utility Functions Policies have been first proposed by [[25]] in order to augment the policies that describe the desired states that a system wants to achieve (called Goal Policies) with a real-valued scalar desirability for the each state. Using Goal Policies the system is responsible for computing the action that will cause the system to move from current state to desired state [[25]]. In ANEMA architecture Goal Policies will describe the high level management directives for network functionality, which are needed for the utility functions. There are also Behavioral Policies that describe the behavior of the network equipment to achieve the given Goal Policies. These policies aim at helping the autonomic entity to manage its behavior based on its context information.
The important thing to be noticed from these architectures is the existence of a layered design and some form of describing the actions that define a self-managed entity.

2.2 Management decisions delegation

The progress towards the autonomic systems implies delegation of the management decision making from the human administrators to the managed components. The most common approaches for this are using the policy based management, simple code mobility, or employing mobile agents . Policy based management [9] specifies the rules that govern the devices' behavior and is making use of a specific protocol (like COPS – Common Open Policy Service) for distributing the policies that must be enforced.
Code mobility and mobile agents can both reduce the network traffic concerning the managed information and allow that the decisions to be made locally based on the processing of the information at the device level. The main difference between a mobile agent and code mobility is that a mobile agent can not only transfer the code but can also save the current state during the migration process.
Network management using mobile agents was intensive studied at the beginning of year 2000. But it failed to gain acceptance mainly due to the security problems that had to be treated as well. [10] Another issue was their usage in large networks where there could appear latency problems in gathering the information (as the mobile agents are travelling to many nodes) or even increase in network traffic (as the mobile agent that is collecting data is getting bigger) consequently the performances were worse than using a traditional management method like SNMP. Various techniques in optimization the migration routes of mobile agents like those described by [11] [10] and security improvements [10] can encourage their usage again.
Recently, the delegation of management tasks from the management stations towards the network, or "embedding management intelligence in the network" resulted in a new concept called "in-network management", which forms an entire research area in 4WARD Project (http://www.4ward-project.eu/).

3. Proposed Model

The model will offer the needed functionalities for an efficient network management. This is understood that the pursued objectives are regarding mainly the quick response time and that the involvement of the resources will not hamper normal network operation.
The architecture is depicted in figure 1. As the case of the other autonomic network

management architectures, there is also a layered structure. At first level, network administrators can introduce the policies that are governing the network, following the directives he/she gets from managers or best practices and legal requirements. Also the network managers can ask for management information at any time, also in an abstract language.
The next level contain Autonomic Managers (AM), which will be responsible for employing the high level policies established by the network administrators and ensure they are kept consistent. In order to exercise their responsibilities, the Autonomic Managers are enquiring an ontology repository in order to get the low level description of the management task assignments. Based on this description AM instantiates some specific Task Agents based on the skeletons provided by an Agent Generator. Task Agents might need to exhibit mobility and move to a Managed Device. Using a generator for the task agents is very important considering the diversity of jobs required in assuring a good network management. At some time, only a few basic functionalities will be required, and other times more advanced features.
The next level includes the Managed Devices that will host the Task Agents and possibly a semantic web service. The nodes will be able to offer to other interested nodes some information regarding the managed objects.
The lowest level is filled by the Managed Resources that are presenting their functionalities in different interfaces. In order to be accepted in practice the solution must complement the existing methods implemented for network management that means it must offer support for traditional protocols like SNMP.

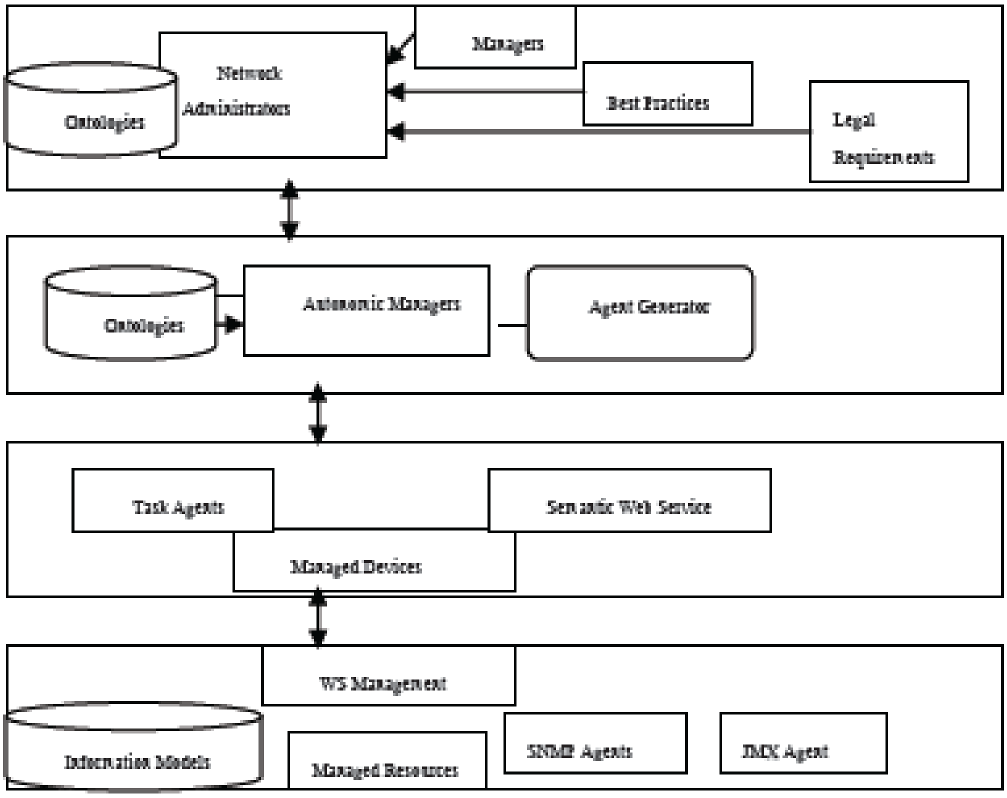

Fig. 1. Proposed architecture

There are a few aspects that need to be further detailed: description of the policies that govern the business environment, description of the shared information model between management and managed entities, description of the network functionalities in order to allow the aggregation of them for achievement of complex management tasks, using of a standard method for the managed interface to ensure the easy interoperability of network elements (e.g. WS management.

3.1 Modeling of the Policy

Policies are rules that describe a choice in the behavior of a system. [12] First policies were applied in security management and several access control policies languages have been developed: XACML[26], XRBAC[27], Ponder [14], Kaos [15], Rei. [16]

There are studies that are pointing out the benefits of using the ontologies for describing the policies especially access control policies [15] [16]. Previous work has also investigated use of ontologies in policy conflict detection [17]

In this framework, policies can be specified on different levels of abstraction, which are organized hierarchically as proposed by [9]. They use the term "Policy Continuum" to refer to the set of different views with various level of abstractions for the same policy.

Policies will be stored in policy repositories as ontologies.

Regardless of the number of levels on which the policies are structured, the policy refinement (the translation from the high level policies to the low level policies) is one of the biggest challenges in the policy-based management. A straightforward way was a manual mapping, but not efficient. Several methods have been proposed for achieving this in an automated way [18] [19] [20]. [18] is using an architecture based on web services and composition techniques. The managed resources are exposing their functionalities as web services referred as low level services. Operations regarding management tasks can be viewed as composition of low level services and are referred as high level services. They are using OWL-S (OWL Services) [21] in order to describe the semantic of the services. Having this description, the Policy refinement engine is able to break down the high level policies from a user interface and return a refined policy in terms of OWL-S description. [18]

As [19] identifies equally important is also the translation from low level policies into high level policies, introducing the concept of Policy Interoperability for this bidirectional mapping. [20] also propose a method for bidirectional translation between policies described in OWL ontologies using the rules written in SWRL (Semantic Web Rule Language) [22].

We have also showed in [17] that SWRL can be successfully used to complement ontology languages like OWL when these are not sufficient descriptive for some aspects, like describing behaviors.

SWRL are in the form of implications between an antecedent (body) and a consequence (head), both are positive conjunctions of atoms. Atoms can be of the form C (x) , P(x,y), sameAs(x,y), or differentFrom (x,y), where C is a concept from an OWL ontology, P is a property and x, y are either variables, OWL individuals, or OWL data values. Whenever the conditions from the antecedent part hold, the consequence part must be true as well [22]. This means that new information can be easily generated related to the OWL individuals.

Considering the fact that OWL ontologies are likely to be used for modeling concepts ranging from business objectives to managed information, SWRL as described in [20] is the right solution to express the mapping between policies.

3.2 Information Model

There are several information models available, such as SMI (Structure of Management Information) used with the SNMP protocol, CIM (http://www.dmtf.org/standards/cim/) (Common Information Model) developed by DMTF that meant to be independent of the protocol used, or DEN-ng (Directory Enabled Networks – next generations) used by [6] in FOCALE architecture which is a complex model including terms of the business objects.

Even if the need of having the information models described in an ontology was recognized by the research field [23] [20], all the information models are quite large and would not be feasible to have them translated in an ontology developed for a certain solution.

Rather this will have to be offered through a standard institution like DMTF or OASIS.

3.3 Services and Modelling of Services

Managed resources, managers, knowledge sources and other components must be composed together and therefore, these systems must be based on open industry standards. [7]

DMTF and IETF have showed an interest in using web services in management systems as the traditional protocol SNMP is rarely used for managing systems or applications [28].

Web Services have evolved already into a mature technology with two important standards: SOAP (Simple Object Access Protocol) – protocol for exchanging messages using XML for encoding, and WSDL (Web Service Description Language) – used to define the web services also by means of XML.

Composition of Web Services has been realized until now at syntactic level through orchestration – combining available services by adding a central coordinator (the orchestrator) that is responsible for invoking and combining sub-activities or through choreography – this does not assume a central coordinator but rather defines complex tasks by describing the conversation followed by each participant [29].

Definition of the services must include a semantic description if we want to enable the automatic execution of them. Among the proposals for semantic web services, the most relevant are: OWL-S (OWL Services) [21], and WSMO (Web Service Modeling Ontology) [24].

OWL-S defines a service in an OWL ontology. It describes which are the inputs and outputs parameters (it can change/process data by transforming a certain input in an output) and preconditions (what needs to be true before the service is invoked) and effects (what needs to become true after the execution of the service) [21]. OWL-S does not have the possibility to define conditions but it can work with a rule language like SWRL, which can also be used to express the rules in policies.

WSMO [24] constitutes of four different elements for describing Web Services: an ontology that provide the concepts used by the other elements, the goals of a web service user, web service descriptions, and a range of mediators that can be used in binding services, goals, ontologies.

As OWL-S has been submitted to the W3C, it is more likely to be used in description of semantic web services.

3.4 Practical Implementation

The proposed model uses JADE (http://jade.tilab.com/) Multi Agent Framework, which is the one of the most used platform for development of multi agent systems. For the ontology creation, the choice is for Protégé (http://protege.stanford.edu/) which allows developing OWL ontologies and more, it offers a convenient add-on "Bean Generator" (http://protege.cim3.net/cgi-bin/wiki.pl?OntologyBeanGenerator) that enables the transformation from OWL ontologies in Jade ontologies.

Jade agents that need to reason on ontologies and apply SWRL rules must integrate the rule engine Jess (http://www.jessrules.com/). To embed Jess in a jade agent , which is actually coded in Java, one have to simply create a jess.Rete object. (the class that implements the rule based inference engine)

Integration of Jade with web services can be done using an add-on like WSIG (Web Services Integration Gateway) (http://jade.tilab.com/index.html), which allows the automatic and bidirectional registration, discovery and invocation of agents and web services. Invocation of web services from Jade Agents does not imply necessary to use this add-on, as there are other tools as well to support the invocation of services from java code.

Protégé can also used as a base for the OWL-S Editor (http://owlseditor.semwebcentral .org/download.shtml), and describe semantically the web services. OWL-S VM (http://projects.semwebcentral.org/projects/owl-s-vm/) can provide some support to execution of the semantic web services but it is not sufficiently documented and looks that its development is abandoned. The existing technologies for web services compositions like WSBPEL (Web Service Business Process Execution Language) (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel) are mature enough to offer at least a syntactic composition.

4. Conclusions

The future networks will have to be based on knowledge that will be described in ontologies. OWL ontologies can achieve the integration of management information from a semantic point of view. The management framework will also have to include information regarding the business objectives that can be described as well in an ontology.

An important issue of modelling through ontologies is how the information is going to be added or updated into these ontologies. This is addressed by using intelligent agents that are employed for gathering or extracting the management information from the network devices.

The definitions of the management behaviors can be expressed in a rule language that uses the ontology concepts like SWRL.

Managed resources will present their functionalities as semantic web services (e.g.in terms of OWL-S) in order not only to provide a maximum decoupling between the inner representation and the presented interface, but also to allow to be modeled further as set of services, which can be called to accomplish a management task.

References

[1] R. Boutaba, J. Xiao, "Network Management: State of the Art," in Proceedings of the 2002 World Computer Congress, pp. 127-146

[2] J. Domingue and D. Fensel, "Problem solving methods in a global networked age," *Artif. Intell. Eng. Des. Anal. Manuf.*, Vol. 23, No. 4, Nov. 2009, pp. 373-390.

[3] S. Wallin and V. Leijon, "Telecom Network and Service Management: An Operator Survey," *In Proceedings of the 12th IFIP/IEEE international Conference on Management of Multimedia and Mobile Networks and Services: Wired-Wireless Multimedia Networks and Services Management*, Venice, Italy, October 26 - 27, 2009. T. Pfeifer and P. Bellavista, Eds. Lecture Notes In Computer Science, Vol. 5842. Springer-Verlag, Berlin, Heidelberg, pp.15-26.

[4] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the internet," *In Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications*, Karlsruhe, Germany, August 25 - 29, 2003. SIGCOMM '03. ACM, New York, NY, pp. 3-10.

[5] B. Jennings, S. van der Meer, S. Balasubramaniam, D. Botvich, M. Foghlu, W. Donnelly and J. Strassner, "Towards autonomic management of communications networks," *Communications Magazine, IEEE*, Vol. 45, No. 10, pp. 112–121, October 2007.

[6] J. C. Strassner, N. Agoulmine, E. Lehtihet, "FOCALE-A Novel Autonomic Networking Architecture," *The Latin American Autonomic Computing Symposium (LAACS 2006)*, Campo Grande, MS, Brazil, July 18-19, 2006, Available at: http://eprints.wit.ie/189/1/2006_LAACS_Strassner_et _al_final.pdf

[7] IBM, *An architectural blueprint for autonomic computing*, June 2006, Available at: http://www-01.ibm.com/software/tivoli/autonomic/pdfs/AC_Blueprint_White_Paper_4th.pdf checked on 25.05.2010

[8] H. Derbel, N. Agoulmine and M. Salaün, "ANEMA: Autonomic network management architecture to support self-configuration and self-optimization in IP networks," *Comput. Netw.*, Vol. 53, No. 3, Feb. 2009, pp. 418-430.

[9] J. Strassner, *Policy-Based Network Management – Solutions for the Next Generation,* Morgan Kauffman, 2003

[10] D. Gavalas, G. E. Tsekouras and C. Anagnostopoulos, "A mobile agent plat-

form for distributed network and systems management," *J. Syst. Softw.*, Vol. 82, No. 2, Feb. 2009, pp. 355-371.

[11] V. K. Verma, R. C. Joshi, B. Xie and D. P. Agrawal, "Combating the bloated state problem in mobile agents based network monitoring applications," *Comput. Netw.*, Vol. 52, No. 17, Dec. 2008, pp. 3218-3228.

[12] T. Moses, eXtensible Access Control Markup Language, Ver. 2.0. OASIS Standard, 2005.

[13] J.B.D. Joshi, R. Bhatti, E. Bertino and A. Ghafoor, "Access-control language for multidomain environments," *IEEE Internet Computing*, Vol. 8, No. 6, 2004.

[14] N. Damianou, N. Dulay, E. Lupu, M. Sloman, "The Ponder Policy Specification Language," *Proceedings of the International Workshop on Policies for Distributed Systems and Networks*, 2001, Lecture Notes In Computer Science; Vol. 1995, pp. 18-38.

[15] J. M. Bradshaw Uszok, M. Johnson, R. Jeffers, A. Tate, J. Dalton, and S. Aitken – "KAoS Policy Management for Semantic Web Services," *In IEEE Intelligent Systems*, 2004, Vol. 19, No. 4, July/August, pp. 32-41

[16] L. Kagal, T. Finin and A. Joshi, "A Policy Language for a Pervasive Computing Environment," *In Proceedings of the 4th IEEE international Workshop on Policies For Distributed Systems and Networks,* June 04 - 06, 2003. POLICY. IEEE Computer Society, Washington, DC, 63.

[17] A. M. Ghiran, G. C. Silaghi, N. Tomai, "Ontology-Based Tools for Automating Integration and Validation of Firewall Rules." In Witold Abramowicz (ed.), 12 BIS Conference, Lecture Notes in Business Information Processing, 21, pp. 37-48. Springer Verlag, 2009.

[18] T. Klie, B. Ernst, and L. Wolf, "Automatic Policy Refinement Using OWL-S and Semantic Infrastr ucture Information," in Proc. 2nd IEEE Int. Workshop on Modelling Autonomic Communications Environments (MACE), San Jos´e, CA., USA, Oct. 2007

[19] S. Magrath, R. Braun, F. Cuervo, "Policy Interoperability and Network Autonomics," *Proc. 1st IFIP Int. Workshop on Autonomic Communication (WAC 2004)*, Berlin, October 2004.

[20] A. Guerrero, V. Villagrá, J. E. López de Vergara, A. Sánchez-Macián, J. Berrocal, "Ontology-based Policy Refinement Using SWRL Rules for Management Information Definitions in OWL," *Proceedings of the 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM'2006),* Dublin, Ireland, 23-25 October 2006, Published in Lecture Notes in Computer Science, Vol 4269, pp. 227-232, Springer Verlag.

[21] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan and K. Sycara, "OWL-S: Semantic Markup for Web Services," W3C Member Submission, 2004.

[22] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, W3C Member Submission (2004)

[23] J. E. López De Vergara, V. A. Villagrá, J. Berrocal, "Applying the web ontology language to management information definitions," *IEEE Commun. Mag.* Vol. 42, pp. 68-74, 2004.

[24] D3.1v0.1 WSMO Primer, WSMO Final Draft April 2006

[25] J. O. Kephart and R. Das, "Achieving Self-Management via Utility Functions," *IEEE Internet Computing*, Vol. 11, No. 1, Jan. 2007, pp. 40-48.

[26] T. Moses, eXtensible Access Control Markup Language, Ver. 2.0. OASIS Standard, 2005

[27] J.B.D. Joshi, R. Bhatti, E. Bertino, A. Ghafoor, "Access-control language for multidomain environments," *IEEE Internet Computing*, Vol. 8, No. 6, 2004, pp. 40-50.

[28] J. Schonwalder, A. Pras, J. P. Martin-

Flatin, “On the future of Internet management technologies,” *Communications Magazine, IEEE*, Vol. 41, No.10, pp. 90-97, Oct. 2003.

[29] M. H. ter Beek, A. Bucchiarone and S. Gnesi,.” Formal Methods for Service Composition,” *Annals of Mathematics, Computing and Teleinformatics*, Vol. 1, No. 5, pp. 1-10, 2007.

Ana-Maria Ghiran has got a bachelor degree and a master degree in Business Information Systems at the Faculty of Economics and Business Administration of the Babes-Bolyai University Cluj-Napoca in 2003 and 2005 respectively. She is currently expected to graduate her PhD Studies and her work focuses on the IT Infrastructure Audit using Artificial Intelligence. She is employed as an Assistant Teacher since 2008.