

Application of OWL-S to Define Management Interfaces Based on Web Services

Jorge E. López de Vergara¹, Víctor A. Villagrà², and Julio Berrocal²

¹ Departamento de Ingeniería Informática, Universidad Autónoma de Madrid,
Escuela Politécnica Superior, Francisco Tomás y Valiente, 11, E 28049 Madrid, Spain
jorge.lopez_vergara@uam.es

² Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid,
E.T.S.I. de Telecomunicación, Av. Complutense, s/n, E 28040 Madrid, Spain
{villagra, berrocal}@dit.upm.es

Abstract. Some network management trends are currently analysing the application of several generic technologies that include Web Services and Ontologies. Web Services can provide an interface to access to managed resources. On the other hand, ontologies provide a way to represent management information. Web Services interfaces can be defined using OWL-S, an ontology of services that semantically describes the set of operations a Web Service provides. This can be useful in configuration management, where each network resource defines the way it can be configured. This paper presents a proposal to describe with the OWL Service ontology the management interfaces based on Web Services. To illustrate this approach, an example is provided in which OWL-S is used to specify the processes needed to configure a resource.

1 Introduction

Network management has evolved in the last decade from proprietary interfaces to standard integrated management interfaces. These interfaces have been based on management specific protocols, such as SNMP or CMIP. This has helped to the development of management applications, among other reasons because it is easier to find a developer with knowledge on these standards than one who can code proprietary interfaces.

However, as stated in [1], the technology market has learnt that implementation costs can be reduced by using generic technologies independent of the application domain, and management applications have not been an exception: it is even easier to find a developer with general communication abilities than with specific network management skills. Nowadays, generic Internet technologies such as HTTP or XML are being adopted to manage network resources.

One of the most promising technologies in this scope is the one related to Web Services, given that they can provide a distributed processing environment with the advantage that they are based on well-known technologies, used broadly by developers. This is one of the reasons why Web Services is a key technology in the evolution of network management systems, solving the complexity problems of other similar approaches such as CORBA, also used to implement management interfaces.

On the other side, one of the pillars of the network management is the definition of the information related to managed resources. Applying the same idea of using generic technologies, these definitions have evolved from using network management specific languages, such as SMI or GDMO, to other generic ones, such as XML schemas or lately, ontology languages such as OWL [2].

At this point, both technologies, Web Services and OWL, meet: it is possible to leverage semantic capabilities of the Web Ontology Language to define the interfaces of Web Services. Instead of using WSDL (Web Services Description Language) [3], the ontology of services known as OWL-S [4] can be used. This ontology can be used to define composite processes by following a well defined structure, formalizing the way in which a set of Web Services operations are invoked. If it is applied to network management, managed resources can semantically describe how they can be managed. In addition, the parameters of each operation can also be classes defined in the ontology that semantically describes the management information.

This article proposes the application of OWL-S to define management interfaces based on Web Services. For this, next sections show how to apply Web Services and Ontologies to network management. Then, OWL-S service ontology is presented, explaining how it can help in the definition of management interfaces. For it, an application example is provided. Conclusions finally remark most important ideas and future works.

2 Web Services in Network Management

A Web Service is a software system designed to support interoperable machine-to-machine interaction over a network, as described in [5]. The main difference with other systems that comply with this definition is that Web Services are an instantiation of the Service-Oriented Computing paradigm [6]. Web Services describe their interfaces with WSDL and their interactions are based on SOAP messages serialized in XML over HTTP. WSDL are XML documents that include definitions of data types, messages, port types or operations, bindings with transport protocols and services. SOAP messages are structured in an envelope with a header, used to transmit metadata, and a body, which is suitable to support RPC-like communications. Also, a third technology that is usually linked to Web Services is UDDI (Universal Description, Discovery and Integration), which allows a service to be registered and later be found by clients.

In the network management scope, several initiatives are related to Web Services:

- WBEM (Web Based Enterprise Management) [7] shares some common characteristics with Web Services: messages are serialized in XML over HTTP. However, WBEM was defined before Web Services, and although it makes use of these standards it defines its own syntax, being a management-specific technology. The Distributed Management Task Force is trying to solve this question, and currently has a workgroup to align WBEM with Web Services.
- OASIS consortium, one of the supporters of Web Services related technologies, has started several initiatives to use them in management, specifying what is known as MoWS (Management of Web Services) and MuWS (Management using Web Services) [8]. These specifications define an architecture and a set of XML

schemas with management parameters that extend the functionality initially provided by Web Services.

- Network Configuration Working Group (Netconf) of the IETF is currently defining a specification to configure network resources using XML messages defined in XML schemas [9]. In this case, they propose several application protocols for the communications, such as SSH, BEEP, and SOAP. This variety allows a better adoption of a future standard, but also reduces the application of Web Services features. It is interesting to notice that this specification is only focused on the interaction between a manager and a managed device, leaving the configuration information standardization as a future work, which is being studied by the Netconf Data Model (Netmod) Working Group.
- Several researchers have studied and compared the SNMP framework with Web Services applied to network management. Some of these works are related to the definition of WSDL interfaces based on SNMP MIBs [10, 11], being able to define coarse and fine grained methods to access MIB information. Other ones are focused on performance of Web Services when they are used to retrieve management information [12, 13, 14]. Results show that Web Services performance is comparable to SNMP when SOAP messages are compressed, so this factor should not be taken as a key factor to reject this technology for network management.

Although most of these approaches are related to interfacing network elements, Web Services can also be used in other more general management tasks. As a conclusion, Web Services can be one of the dominant technologies for network management in the near future to access and perform operations on managed resources.

3 Ontologies in Network Management

Other generic technology currently under review in the network management scope is the Semantic Web and its main component: ontologies. They are explicit and formal specifications of shared conceptualizations [15]: they provide the definition of a set of concepts, their taxonomy, interrelation and the rules that govern these concepts. The interesting point that differentiates ontology languages from management information definition languages is that their semantics has been formalized. Then, different ontology representations have been proposed to describe the management information, including OKBC [16], RDF-S [17] or description logics [18].

Prior works of the authors about ontologies in network management have also given some results:

- A framework can be established to compare different management information languages from an ontology viewpoint. All these languages have different levels of expressiveness, so the same resource can be defined in multiple ways depending on the used model. As ontologies provide the necessary constructs to represent the semantics of specified information, these constructs can be used to compare the semantic expressiveness of management information languages in a neutral way. Using this approach, management information description languages are analyzed and compared based on their semantic expressiveness in [19].

- Solutions applied to integrate different ontologies are not only syntactic translations of different languages: they also deal with the semantics of the information. A method to merge and map management information models of different domains (e.g. CIM schemas and SNMP MIBs) can be defined by applying ontology techniques [20], generating a mapping ontology that contains all the elements defined in both models and their mapping rules. These mapping rules can describe 1:1, 1:n and m:n correspondences.
- Several ontology definition languages can be found in the literature, being OWL [2] currently the most relevant one because of its adoption by the World Wide Web Consortium. Still, they have to be adapted to the management scope, as there are some constructs they do not include. In [21], the use of OWL has been proposed to specify management information, describing a set of mapping rules with constructions of management languages. Besides the mapping specification, other constructions that include all the components of the management information models are proposed to be included in OWL.
- Another advantage of ontologies is the ability of expressing behaviour of the defined information. For this, OWL provides the Semantic Web Rule Language (SWRL) [22], allowing the definition of horn clauses inside the ontology. A proposal to include behaviour rules in management information with this language is given in [23]. Thus, usual behaviour definitions included implicitly in the management information definitions and explicitly in policy definitions can then be expressed formally, and included with those definitions, which allows a manager reasoning and working with them.

Finally, this work provides an application of OWL-S, an ontology of services, to improve current approaches based on the use of Web Services for network management, as shown in next section.

4 OWL-S and Network Management

In this section, the benefits of using the ontology of services OWL-S to describe Web Service management interfaces are provided. For this, OWL-S is first described, and later its usefulness in network management is given.

4.1 OWL-S: An Ontology to Describe Services

OWL-S [4] is an upper ontology of services developed in the semantic web scope that aims at the automatic discovery, invocation, composition and interoperation of Web Services. It has been defined as a part of the DARPA Agent Markup Language program, and later submitted to the World Wide Web Consortium. OWL-S complements WSDL descriptions by supporting a richer semantics, including logical constraints between the input and output parameters of services. Moreover, WSDL documents can specify data types using XML Schema, whereas OWL-S uses OWL classes, which provides a better semantics. In this way, a software agent can select and operate with the Web Service that meets its needs to perform a task.

The ontology OWL-S is composed of several classes. The main class is the service. This service presents a service profile, is described by a service model, and supports a service grounding:

- The service profile provides the information needed to discover a service. It tells what the service does, so that an agent can determine if the service meets its needs.
- The service model tells a client how to use the service, detailing the semantic content of each request, including preconditions, results, and processes to be carried out by the service. These processes can be atomic or composite. Composite processes model the behaviour the client can perform by sending and receiving a set of messages following a set of control constructs (sequence, split, join, choice, if-then-else, iterate, repeat-while, repeat-until).
- The service grounding details how an agent can access a service, defining a mapping with a WSDL document of the Web Service: an atomic process is a WSDL operation, and inputs and outputs are WSDL messages.

One of the initial goals of OWL-S was the automatic monitoring of Web Service execution. This can be useful for the management of these services, where a user could know the state of a request (how long it is taking, exceptions appeared, etc.). However, this feature has not been included in any of the versions released so far.

4.2 Application to Network Management

Once OWL-S has been presented, some ideas can be applied in a Web Services management scenario which makes this ontology useful:

- Management information exchanged between managers and managed elements by means of Web Services can directly be instances of ontology classes, as proposed previously. Web Services can exchange OWL instances instead of XML schema datatypes. This can be useful to avoid several management information translations (SMI or MOF to XML schema, WSDL, OWL, etc.) if the manager can deal with ontology definitions.
- The definition of processes includes a set of preconditions and results that can be expressed as formal logic rules (in SWRL or other languages), specifying the behaviour of such processes. This can help managers to interact with managed resources in an easier way, behaving as described in those rules.
- The description of a service with OWL-S can be used to obtain in runtime how to manage a resource, as it includes both the definition of the needed processes and their mappings to Web Service operations. This can be important in a future scenario in which every resource (which can be devices, applications or services) defines its own management information, not only based on a set of variables, but also on a set of operations, self-describing its manageability. Then, a manager, based on this description can manage the resource easily, thanks to the formality of OWL-S.

Next section applies these ideas to a concrete management functional area: network configuration.

5 Application of OWL-S to Network Configuration

This section shows how OWL-S can be useful in network management, with an example of its application to the configuration of managed resources. It is very common that each network resource has its own configuration. In fact, current standardization efforts of the Netconf Working Group are only focused on the definition of a set of operations, but it is not included how to sequence them, and what information has to be provided, as they depend on each concrete resource. However, it is also very important to specify the way to perform that configuration to reduce the costs of implementing a management application.

Then, as stated before, OWL-S can be useful in this scope, as it can be used to describe how to perform the configuration of a network resource. With this information, a generic management application able of reading such ontologies could manage the configuration of resources based on Web Services, even if it does not know *a priori* how to do it, which can be very helpful in multimedia networks and services. Moreover, configuration information can be defined as a set of classes of a configuration ontology, leveraging existing definitions (for instance, CIM Setting classes), to be used as parameters of the configuration operations. Generic classes can be later specialized for each resource, partially solving the problem of specifying the configuration information. These configuration classes will be defined in OWL, allowing a complete integration with OWL-S, and with an ontology-based management system.

To illustrate these ideas, OWL-S has been applied to describe a network configuration service based on current Netconf works. Although this example is generic, it can be valid to understand how to use this ontology. In a real scenario, each resource will define its own configuration service, with its own preconditions, results processes, and mappings with related Web Service operations, in the same way this generic configuration service has been specified.

To describe in OWL-S this generic network configuration service, several instances of different classes of this ontology have to be defined so that they can semantically describe this service. To help in this work, OWL-S Editor [24] has been used, as shown in **Fig. 1**. This tool is integrated with Protégé [25], an open source ontology editor that in its last version is well integrated with the OWL knowledge model. All diagrams presented in this paper have been generated with this tool, taking the information introduced in the ontology editor.

Four main instances have been defined, as depicted in **Fig. 2**, using the classes presented before in section 0. The first instance is named *netconfService*, and it is an instance of *service:Service*. This is the main instance, which links with *netconfProfile*, *netconfGrounding*, and *netconfConfigurationProcess*. The instance *netconfProfile* represents the property *service:presents* of the *netconfService*, that is, the profile of the service with a general description of the Web Service, including required inputs and preconditions, as well as expected effects of the service. The instance *netconfGrounding* represents the property *service:supports* of the *netconfService*, the grounding of the service with the mappings of the ontology in a WSDL document. Finally, *netconfConfigurationProcess* is a composite process defined in the service model that defines the sequence of atomic processes to be performed when configuring the resource.

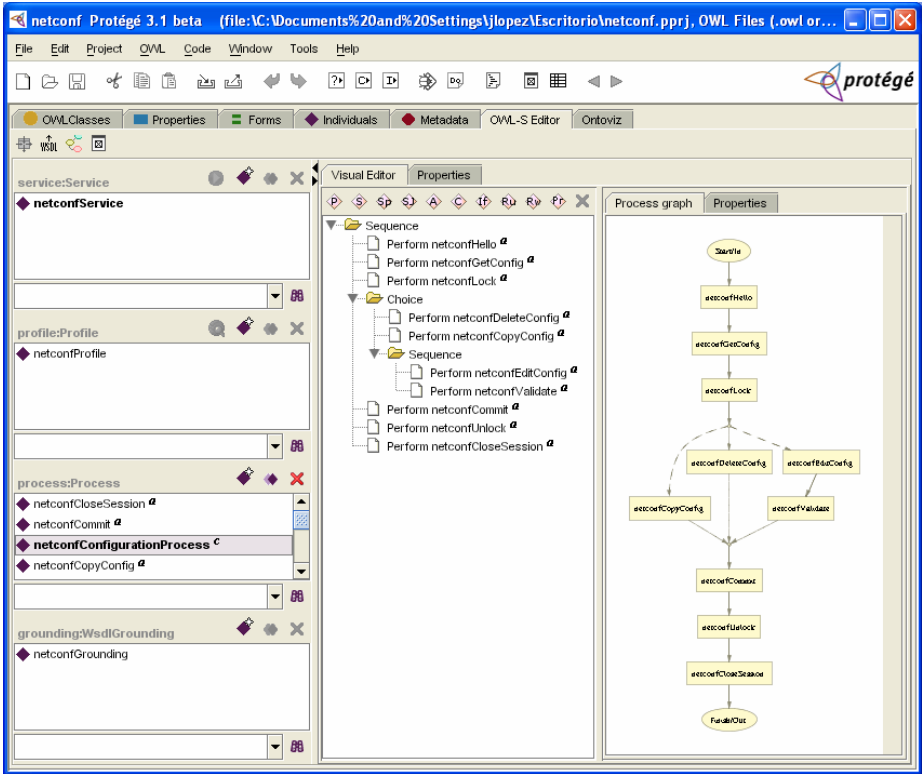


Fig. 1. Editing OWL-S ontology about network configuration with OWL-S editor

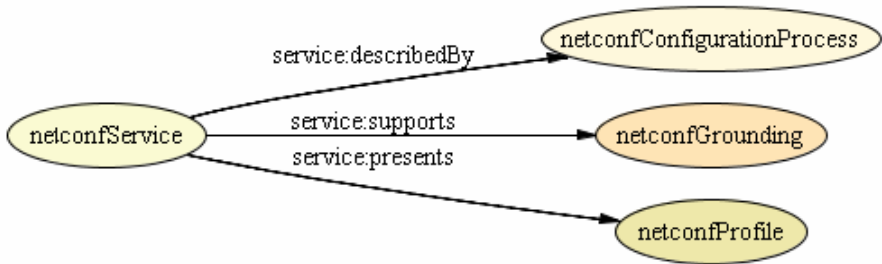


Fig. 2. Instances of OWL-S classes to describe the Network Configuration Service

The service model also includes several atomic process instances, as shown in Fig. 3, where they are linked to the class *process:AtomicProcess*. They are related to different NetConf operations [9]:

- Session maintenance: hello (*netconfHello*), close-session (*netconfCloseSession*)
- Configuration manipulation: get-config (*netconfGetConfig*), edit-config (*netconfEditConfig*), copy-config (*netconfCopyConfig*), delete-config (*netconfDeleteConfig*), validate (*netconfValidate*).

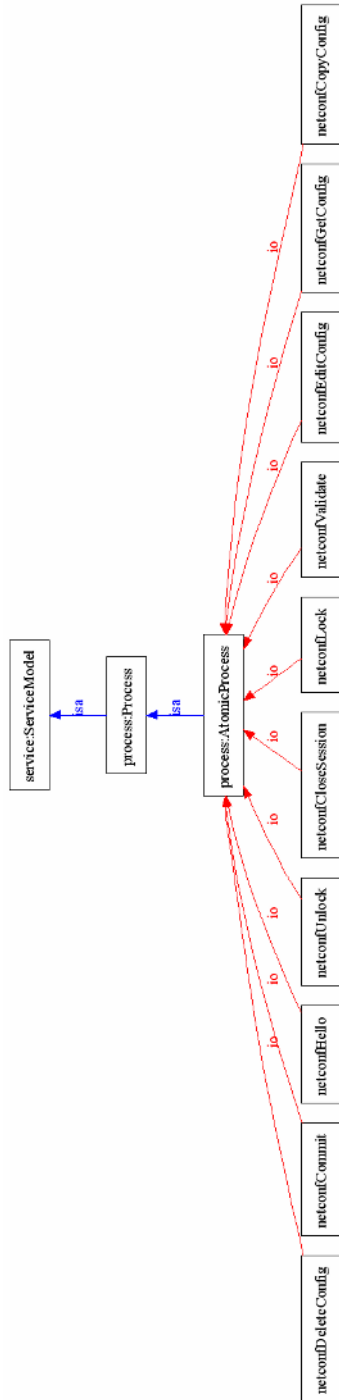


Fig. 3. Atomic process instances

- Transaction management: lock (*netconfLock*), commit (*netconfCommit*), unlock (*netconfUnlock*)

Each of these processes has its own input and output parameters, preconditions and results. For example, the process *netconfEditConfig* has as inputs a target datastore, a configuration subtree, and the kind of operation (merge, replace, create, delete). These parameters can be defined using OWL classes as parameter types. For instance, the *Datastore* class can be specialized in *Startup*, *Running* or *Candidate*, as shown in Fig. 4. A precondition for *netconfEditConfig* can logically express in SWRL the need of obtaining a lock on the edited configuration before performing this process.

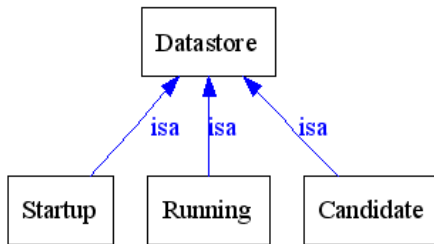


Fig. 4. Datastore class specialization

Then, the *netconfConfigurationProcess* can be composed of this set of atomic processes, defining a control flow similar to the one depicted in Fig. 5. In this case, the following sequence of processes has been defined:

1. The first process is a *netconfHello* to open the NetConf session.
2. Then, a *netconfGetConfig* is performed to obtain current configuration.
3. Next, there is a *netconfLock*, to lock it from other managers.
4. At this point, there is a choice with three different possibilities:
 - *netconfDeleteConfig*, to delete the configuration.
 - *netconfCopyConfig*, to copy the configuration.
 - The sequence *netconfEditConfig* and *netconfValidate*, to edit and validate the configuration.
5. Then, once one of the processes of this choice has been performed, the new configuration is committed with a *netconfCommit*.
6. Next, the configuration is unlocked with a *netconfUnlock*.
7. Finally, the session is closed with a *neconfCloseSession*.

This control flow could be enhanced by including if-then-else or repeat-while constructs, which would only let the flow continue if a logic expression is met (for instance, a SWRL expression stating that the configuration is committed only if it has been previously validated).

Once that the set of instances of OWL-S classes have been defined, a network resource can publish the ontology describing its configuration service, and an OWL-S-capable manager can configure it properly, by following the set of processes defined in the service model.

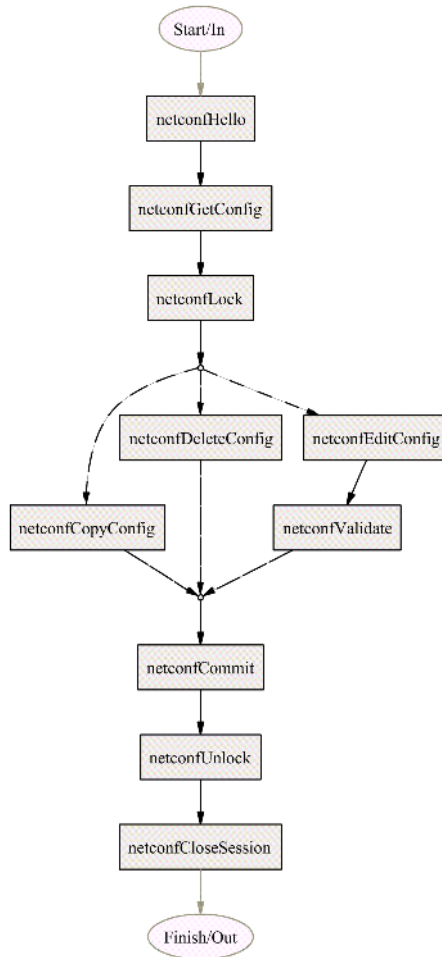


Fig. 5. Control flow of atomic processes conforming the *netconfConfigurationProcess*

6 Conclusions

Web Services provide a generic technology that is currently gaining momentum in the network management community. At the same time, other works have also proposed the use of ontology languages to describe the management information. OWL-S is an upper ontology of services defined to enrich the semantics of Web Services, being also applicable to management interfaces based on Web Services. This paper has presented a proposal that applies OWL-S in this scope.

The OWL-S ontology can be useful to self-describe how to manage a resource, which can be important if that resource does not use standard management information. In this case, a manager can download the OWL-S description of the

management interface, and manage the resource by interpreting this description, which includes the definition of management information as a set of OWL classes, including logic rules describing preconditions and control expressions.

To check the feasibility of this approach, a set of OWL-S classes have been instantiated to describe a generic network configuration service. In this case, the service model included a set of atomic processes, based on the operations defined in the Network Configuration Working Group, and a composite process that represents the flow control of a manager configuring a network resource. As these resources are very different, it can be very useful to describe in such a way how a manager can perform the configuration.

Future works include the improvement of the defined information, containing logic expressions to better model the composite process, as well as the implementation of a manager able of interpreting these OWL-S ontologies and accessing to management interfaces based on Web Services with such descriptions.

References

1. J. Schönwälder, A. Pras, J.P. Martin-Flatin: On the Future of Internet Management Technologies. *IEEE Communications Magazine*, Vol. 41, Issue 10 (2003) 90-97
2. P.F. Patel-Schneider, P. Hayes, I. Horrocks: OWL Web Ontology Language Semantics and Abstract Syntax. *W3C Recommendation* (10 February 2004)
3. R. Chinnici, M. Gudgin, J.J. Moreau, J. Schlimmer, S. Weerawarana: Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. *W3C Working Draft* (3 August 2004)
4. D. Martin, editor: OWL-S: Semantic Markup for Web Services. *W3C Member Submission* (22 November 2004)
5. H. Haas, A. Brown: Web Services Glossary. *W3C Working Group Note* (11 February 2004)
6. M. P. Papazoglou, D. Georgakopoulos: Service-Oriented Computing. *Communications of the ACM*, Vol. 46, No. 10 (2003), 24-28
7. Distributed Management Task Force, Inc.: Specification for CIM Operations over HTTP, Version 1.1. *DMTF Standard DSP0200* (2003)
8. OASIS Web Services Distributed Management (WSDM). http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm
9. M.J. Choi, H.M. Choi, J.W. Hong, H.T. Ju: XML-Based Configuration Management for IP Network Devices. *IEEE Communications Magazine*, Vol. 42, Issue 7 (2004) 84-91
10. J. van Sloten, A. Pras, M. van Sinderen: On the standardisation of Web Service management operations. In: *Proc. 10th Open European Summer School (EUNICE 2004) and IFIP WG 6.3 Workshop, Tampere, Finland (2004)* 143-150
11. T. Drevers, R. van de Meent, A. Pras: Prototyping Web Services based Network Monitoring. In: *Proc. 10th Open European Summer School (EUNICE 2004) and IFIP WG 6.3 Workshop, Tampere, Finland (2004)* 135-142
12. R. Neisse, R.L. Vianna, L.Z. Granville, M.J.B. Almeida, L.M.R. Tarouco: Implementation and Bandwidth Consumption Evaluation of SNMP to Web Services Gateways. In: *Proc. 9th IEEE/IFIP Network Operations and Management Symposium (NOMS 2004)*, Vol.1, Seoul, Korea (2004) 715 - 728

13. G. Pavlou, P. Flegkas, S. Gouveris, A. Liotta: On Management Technologies and the Potential of Web Services. *IEEE Communications Magazine*, Vol. 42, Issue 7 (2004) 58-66
14. A. Pras, T. Dreviers, R. van de Meent, D. Cuartel: Comparing the Performance of SNMP and Web Services-Based Management. *eTransactions on Network and Service Management* (Fall 2004)
15. R. Studer, V.R. Benjamins, D. Fensel: Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering*, 25. (1998) 161-197
16. E. Lavinal, T. Desprats, Y. Raynaud: A Conceptual Framework for Building CIM-Based Ontologies. In: *Proceedings of the Eighth IFIP/IEEE International Symposium on Integrated Network Management (IM'2003)*, Colorado Springs, Colorado, U.S.A. (2003) 135-138
17. J. Shen, Y. Yang: RDF-Based Knowledge Models for Network Management. In: *Proceedings of the Eighth IFIP/IEEE International Symposium on Integrated Network Management (IM'2003)*, Colorado Springs, Colorado, U.S.A. (2003) 123-126
18. G. Lanfranchi, P. Della Peruta, A. Perrone, D. Calvanese: Towards a new landscape of systems management in an autonomic computing environment. *IBM Systems Journal*, Vol. 42, No. 1 (2003) 119-128
19. J.E. López de Vergara, V.A. Villagrà, J.I. Asensio, J. Berrocal: Ontologies: Giving Semantics to Network Management Models. *IEEE Network*, Vol. 17, No. 3 (2003) 15-21
20. J.E. López de Vergara, V.A. Villagrà, J. Berrocal: Benefits of Using Ontologies in the Management of High Speed Networks. *Lecture Notes in Computer Science*, Vol. 3079, Springer-Verlag. (2004) 1007-1018
21. J.E. López de Vergara, V.A. Villagrà, Julio Berrocal: Applying the Web Ontology Language to management information definitions. *IEEE Communications Magazine*, Vol. 42, Issue 7, (2004) 68-74
22. I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, M. Dean: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *W3C Member Submission* (21 May 2004)
23. A. Guerrero, V.A. Villagrà, J.E. López de Vergara: Including management behavior defined with SWRL rules in an Ontology-based management framework. *Proceedings of the 12th Annual Workshop of the HP Openview University Association*, Porto, Portugal (2005)
24. D. Elenius, G. Denker, D. Martin, F. Gilham, J. Khouri, S. Sadaati, R. Senanayake: The OWL-S Editor – A Development Tool for Semantic Web Services. In: *Proc. Second European Semantic Web Conference*, Heraklion, Greece (2005)
25. N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Ferguson, M. A. Musen: Creating Semantic Web Contents with Protégé-2000. *IEEE Intelligent Systems*, Vol. 16, Issue 2, (2001) 60-71