Towards an Ontology-based Security Management

Bill TSOUMAS, Dimitris GRITZALIS[1]
Information Security and Critical Infrastructure Protection Research Group, Dept. of Informatics, Athens University of Economics and Business
76 Patission Ave., Athens GR-10434, Greece
e-mail: {bts,dgrit}@aueb.gr

Abstract. *The paramount complexity of enterprise information leads to hard-to-deal security management issues and system configurations. We present a security management framework of an arbitrary information system (IS) which builds upon knowledge-based resources, such as security ontology (SO) providing reusable security knowledge interoperability, aggregation and reasoning exploiting security knowledge from diverse sources; in addition, the separation of security requirements from their technical implementations facilitates the security management. We provide a feasible framework, which links the high-level policy statements and deployable security controls and facilitates the security expert's work.*

Keywords: Security Management, Security Knowledge, Security Ontology, Risk Assessment.

1. Introduction

The dynamic nature of modern IS underlines the paramount importance of sound security management. The establishment of such a framework which effectively supports the Information System (IS) security management as defined with PDCA cycle (Plan, Do, Check, Act) [1], is not an easy task; security experts deal with a variety of diverse security-related knowledge sources such as *high-level statements* (e.g. output of Risk Analysis (RA) tools, policy statements expressed in a managerial level, Service Level Agreements), *IS technical information* and *managerial decisions*, which formulate the *organizational security objectives*. The ultimate output of this process is to: a) transform the security objectives into an effective set of *security controls* (applicable, low-level technical countermeasures, which can be applied directly to the IS devices), b) deploy and manage the security controls over the IS and, c) establish a risk management process over the effectiveness and efficiency of the security controls in place. This is often an effort-consuming intervention – especially for large organizations – which has not yet been properly assisted by automated processes.

We argue that we may employ a structured approach to support the process leading from informal, high-level statements found in policy and RA documents to deployable technical controls. The outcome of this process will be a knowledge-based, ontology-centric security management system, eventually bridging the IS risk assessment and organizational security policies with security management; the critical aspect for a successful framework realization, is to *separate the security requirements ("What") from their technical implementations ("How"), which are eventually matched in order to formulate the necessary actions ("Do")* to be deployed in the information assets.
This paper further is organized as follows; in section 2 we describe the background and our standards-based security ontology, while in section 3 we present our ontology-based security framework; related work is discussed in section 4, and finally, conclusions and further research directions are given in section 5.

2. Security Ontology

The basis of our framework is the SO which builds upon well-known paradigms and standards in information, knowledge and security management; first, we extend the DMTF Common Information Model (CIM) [2] standard in order to use it as a container for IS security-related information, based on wide accepted security management standards. Next, we enrich this CIM extension with ontological semantics [3] in order to support knowledge sharing and reuse defining a generic Security Ontology (SO), which is "*an ontology that elaborates on the security aspects of an information system*". CIM is advantageous for our approach in that the model can be mapped, under certain limitations, to structured Semantic Web specifications such as OWL [4].
For the modelling of the ontology concepts and relations, we have used the wide-accepted standards ISO-/IEC 17799 [5], British Standard 7799 Part 2 [1], Australian Standard Handbook of Information Security Risk Management (AS/NZS 4360) [6], as well as the CCTA Risk Analysis and Management Method (CRAMM) [7]. The SO is formulated as a CIM extension schema enriched with ontological semantics, modelling the security management information; in addition, it is linked with the legacy CIM concepts in order to access other CIM-OWL ontologies. The SO acts as a container for the IS security requirements ("*What*" part), as they are extracted from the available information sources. In order to achieve this goal, we have followed a three-phase approach: a) modelling in a conceptual level, b) linking with CIM as an extension, and c) implementing the SO in OWL. While there is no standard method for ontology development [8], we followed the collaborative approach

[1] *Corresponding author.*

for ontology design described in [9]. A detailed description of the SO development process is given in [10], whereas the conceptual model of our SO is

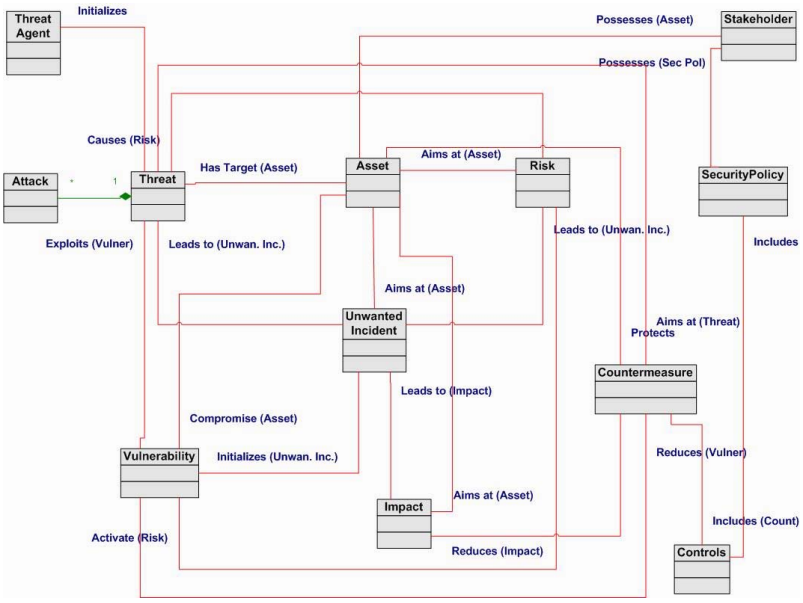


Figure 1: Conceptual model of RA SO

depicted in Figure 1.

Among the others, we have used *Asset, Stakeholder, Vulnerability, Countermeasure and Threat* concepts in the construction of the SO. More specifically, a *Stakeholder* (i.e. an entity with a valid interest) possesses an *Asset*, which in turn can be compromised by a *Vulnerability*. In addition, a *Threat* initiated by a *Threat Agent* targets an *Asset* and exploits a *Vulnerability* of the *asset* in order to achieve its goal. Exploitation of a *Vulnerability* leads to the realization of an *Unwanted Incident,* which has a certain *Impact*. Furthermore, *Countermeasure*s reduce the impact of the *threat* with the use of *Control*s. Finally, *Security Policy* formulates the *Control*s into a manageable security framework possessed by *Stakeholder*s. CIM uses UML to define classes and the relations between them. UML, though, limits the ability of the model to "*provide globally understood constructs for expressing semantics and to impose a common interpretation of the meta-data contained within the model*" [11]. In addition, interoperability, aggregation and reasoning are necessary features for representing knowledge in a domain, providing a formal vocabulary as well as a formal representation of the relations between the concepts of the domain. In order to union the SO with the CIM metamodel, we transform the latter to a formal ontology based on the methodology described in [11] (in this paper, we transform only the CIM_ManagedElement concept, which is necessary for the demonstration of our approach).

Having imprint the conceptual model of the Security Ontology, we proceed to the implementation of the SO itself, based on the OWL plug-in of *Protégé* tool [12], [13]. We have used the Description-Logic versi-

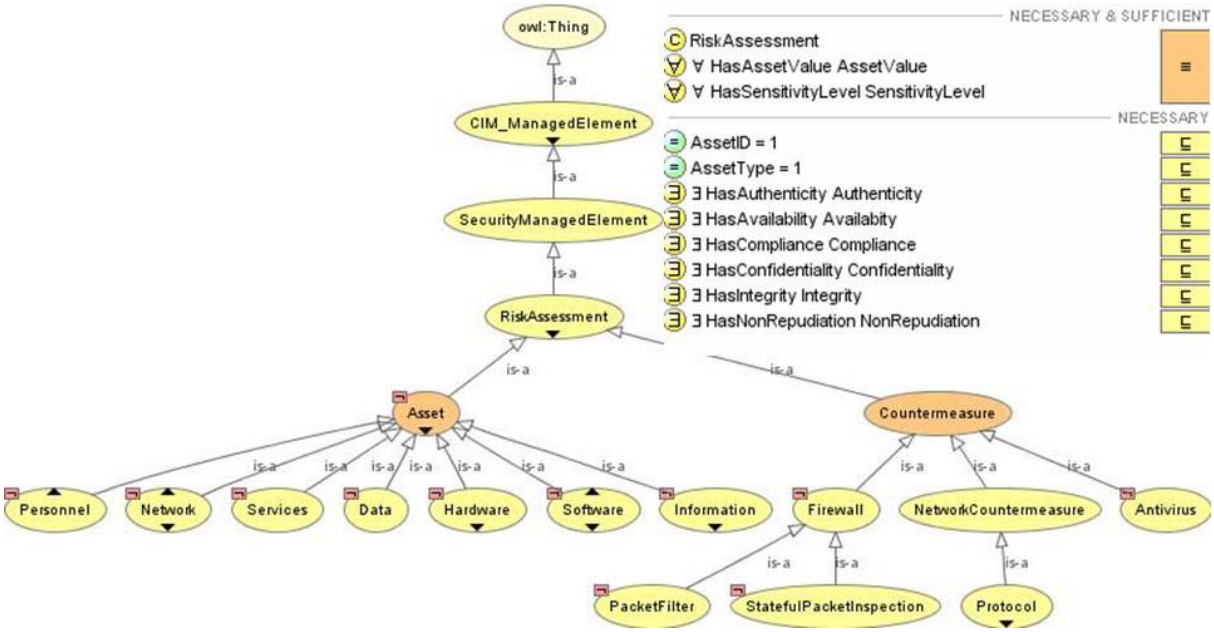


Figure 2: RA Security Ontology in Protégé

on of OWL (OWL-DL), which has great expression capabilities and is used in the majority of the ontolo-

gical applications. The ontology concepts have been populated with certain attributes and relevant semantic constraints, and the concepts have been defined as mutually disjoint where necessary according to the OWL specification. Furthermore, the ontology concepts are linked with other, arbitrary concepts of CIM-OWL domains representing information resources. This linkage can be performed during run-time and facilitates the reuse of existing CIM-OWL implementations. Figure 2 depicts the security ontology related with RA - for spatial reasons only a small portion of the full ontology is depicted; at the top right the semantic constraints of the concept *Asset* are presented. Furthermore, in *Asset* concept we have defined a multi-dimensional array named Threats-CMs field, representing the threats that may put the asset in risk.

2.1. Information Asset Countermeasure Semantics

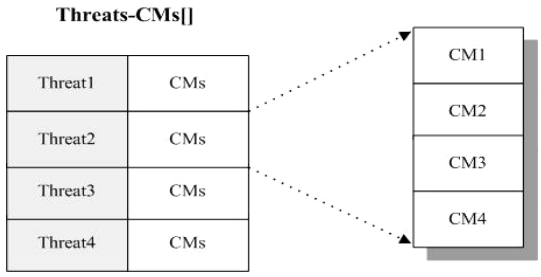
Every information asset is associated with certain threats, which can be mitigated by a set of countermeasures. At this point, we focus on countermeasures' characteristics that can be obtained from the RA information sources. Our first task is to define the structure of a countermeasure – in other words we make a preliminary approach to answer the question "*what attributes are necessary in order to define a countermeasure*?" The basic countermeasure characteristics are further depicted in Table 1. The Group/-Subgroup attributes follow the CRAMM [7] taxonomy scheme for countermeasures.

Countermeasure Structure	
Countermeasure_Identifier (CM_ID)	Unique identifier
Target	The IS resource that this countermeasure is going to be applied
Subject	The entity that is going to apply the countermeasure to the Target
Countermeasure Group	Categorizes the countermeasure in a group
Countermeasure Subgroup	Categorizes the countermeasure in a subgroup (further)
Action	Action(s) to be taken for the countermeasure to be applied
Constraints []	Time, place, and Subject constrains
Type	[Managerial | Procedural | Technical]
SecurityAttributes2Preserve	[Confidentiality | Integrity | Availability]
Type Of Control	[Protective | Detective | Corrective]
Risk Mitigation Factor	[High | Medium | Low]
Control Purpose	[Security | Audit]

Table 1. Countermeasure Definition

Our implementation for extracting the countermeasure attributes is an information asset-based one; during the instantiation of the ontology classes, each identified asset is associated with an instance of the relevant concept (e.g. "Server"); in the sequel, the concept instance is associated with the relevant threats which populate the *Threats-CMs[]* array, with each row representing a single threat for the specific asset, along with an array of countermeasures that mitigate the specific threat, shown in Figure 3. Due to OWL limitations to express arrays, we have chosen to implement this multi-dimensional array with character arrays (strings) using certain delimiters for the inner structure of each countermeasure (CM*i* elements of every CMs item).

Figure 3: Decomposition of Threats-CMs array for a given Asset



3. Framework Description

The security expert has a handful of security-related knowledge information sources, which influence him in a direct or indirect way in order to implement the security controls. Direct sources are bound to the specific IS and include organization policies and Service Level Agreements (SLA), RA outputs and IS infrastructure information. The indirect sources are implicitly associated with the given IS and include security and risk management standards [1] [14], technical best practices [15], security advisories from vendors [16] and security portals [17], security mailing lists [18] and vulnerability catalogues such as CVE [19]. An indirect source of security information, usually neglected by the experts is the *business decisions* made by the organization stakeholders (e.g. "*Company's IT systems should support the Sales process*"). This may raise certain IS security considerations (e.g. "*the sales application must be accessible by the salesmen with wireless laptops during business hours*").

In the following we present a brief description of the necessary steps in order to establish the IS security management framework under discussion (Figure 4). Four major phases and six steps can be identified throughout the process, namely: a) *Building of Security Ontology,* b) *Security Requirements Collection,*

c) *Security Actions Definition, and* d) *Security Actions Deployment and Monitoring*. The steps in each phase are as follows:

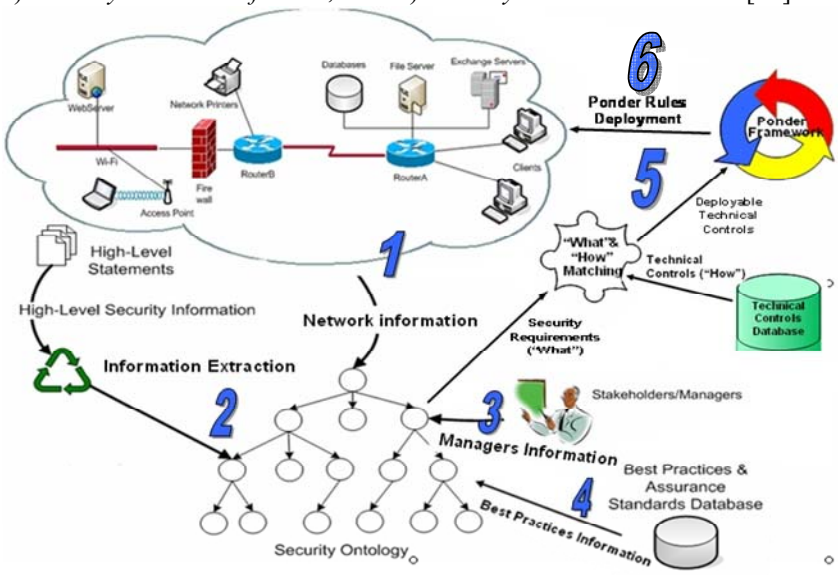


Figure 4: An ontology-centric architecture for IS security management

Phase 1: Building of Security Ontology (Step 1)

I. *Get IS infrastructure data*; in this initial step, technical data concerning the network topology, technologies used, servers, wireless access points, services and active ports are located through the use of network scanning tools such as Nmap [20] and NetStumbler [21];

II. *Justify with organization managers and discuss business decisions*; management input entered into the knowledge system via dialog-based interfaces may influence dramatically the security of the IS, since it might affect network topologies, active services and open ports.

III. *Generate ontology concepts' instances from infrastructure data*; in this step ontology instances are generated from the correct concepts of the SO. Populate the instances with information from step I. The management of concepts' instances and population is performed via ontology environments and tools, such as Protégé [12] with API support.

Phase 2: Security Requirements Collection (Steps 2,3,4)

IV. *Extract security knowledge from the IS policy document;* perform information extraction work from the policy statements and populate the ontology concept instances with the extracted information, using tools such as GATE [22]. Eventually fill the missing information (if possible) in the instances from step II.

V. *Present the security requirements to management and security expert(s) for evaluation*; if necessary, perform adjustments and/or corrections to security requirements. The database of security and assurance standards may be used for enriching the security requirements, in case the information contained in the policy documents is deemed insufficient.

Phase 3: Security Actions Definition (Step 5)

VI. *Associate the security requirements with specific security controls;* this step performs the linking of requirements with deployable security controls (Database of Technical Controls) into *Actions*, customized for the concept instance under question. The Database of Technical Controls is an enabling repository of deployable security measures. The process is facilitated from the infrastructure data collected during step I.

VII. *Transform the controls identified into a Ponder-compatible input;* this step involves the transformation of the actions (controls) specified in step VI into a form that can be piped into Ponder rules [23]. The transformation to Ponder can be realized through an appropriate interface. The CIM-Ponder transformation/mapping is already discussed in [24] [25] [26].

Phase 4: Security Actions Deployment and Monitoring (Step 6)

VIII. *Deploy the Ponder rules over the IS infrastructure;* employ Ponder management framework in order to realize the security requirements (enforcing the policy statements that apply to technical controls) over the IS devices.

IX. *Iterate from step I in a timely basis;* in order to keep up with the changes in the IS environment and policy modifications, the whole process should be employed over certain periods of time.

Furthermore, reporting facilities should be in place so as to be able to monitor every step of the process. Additional capabilities, such as storage of the ontology in a suitable manner so as to be able to perform

queries upon the ontology, are highly preferable. The representation of the ontology in OWL [27] promotes reusability and exchange of security knowledge.

4. Related work

Although the need for a security ontology has been recognized by the research community [28] [29] [30], only partial attention has been drawn for a common solution. A research loosely-related to our work [29] [30] deal mainly with access control issues; Standards discussed include XML Signatures and integration with Security Assertions Markup Language (SAML) [31] and XACML [32] constructs. Work on KAON [33] focuses mostly on the managing infrastructure of generic ontologies and metadata, whereas in [24] Rei language authors present a policy ontology based on deontic logic, elaborating, among others, on delegation of actions.

Raskin et al. presented an ontology-driven approach to information security [34], arguing that a security ontology could organize and systematize all the security phenomena such as computer attacks and support attack prediction.

The design of the KAoS [35] [36] policy ontology suggests the use of a description logic inference engine to analyze policy rules. The policy analysis mechanism in the e-Wallet system [37] exploits the XSLT technology to translate policy rules from RDF to JESS rules and uses a JESS rule engine to compute policy restrictions. Furthermore, the SOUPA [38] policy language is similar to Rei in modeling a policy as a set of rules that defines restrictions on actions with limited support for meta-policy reasoning and speech-acts.

The legacy DMTF approach (i.e. the root of our SO), lacks a) the security management aspect (which we define as an Extension Schema), b) the centralized management of security management information, and c) the domain knowledge perspective, which we incorporate into our model enriching the Extension Schema with ontological support. The modelling of CIM with OWL has also been proposed by Clemente et al. in [4]. Finally, Tsoumas et. al. in [3] defined a generic knowledge-centric, ontology-based framework for IS security management with combination of security knowledge from a variety of sources.

In addition, most of these approaches are related with specific aspects of security and particularly to specific application domains; our approach is generic enough to be applied in every information system, incorporating security knowledge from various sources. Furthermore, all aforementioned approaches lack the security standards support, which we use for modeling the security requirements.

The work presented in this paper, however, is distinguished from these previous efforts in that this work:

a) extends the legacy CIM with risk assessment support that is directly linked with the modeled information resources,
b) concentrates the security requirements in a centrally managed location, therefore facilitating the overall management,
c) facilitates the security-related reuse of already OWL-modeled CIM ontologies without the need to redefine them,
d) abstracts the security management requirements of a CIM-based domain from the actual implementation, therefore reducing the complexity of countermeasures management,
e) uses NLP techniques for extracting the security requirements from high-level statements combining with network audit information,
f) takes advantage of security knowledge bases in order to reuse and extend previously gathered knowledge.

5. Conclusions and Further Research

In this work we described a framework for security knowledge acquisition and management; we defined a standards-based knowledge container (Security Ontology) which extends the CIM model with ontological semantics and can be used for reusable security knowledge interoperability, aggregation and reasoning, using security knowledge from diverse sources. We fully implemented a SO that relates to Risk Assessment and demonstrated that the security information extraction from high-level statements (RA countermeasures) is feasible.

Regarding future work we envisage the development of a standards-based, best practices database with implicit security knowledge in order to support the information extraction and decision making process; in the same direction is the enhancement of heuristic rules so as to produce more concrete and accurate results. Evaluation of the results is one more interesting area, which will be assisted by the further enrichment of the ontology with - more - semantic rules. Finally, we plan to experiment with more complicated countermeasures in order to simulate real-world complexity and to integrate our approach with network management tools and risk assessment software.

References

[1] British Standard 7799, Part 2 (1999), Information Technology - Specification for Information Security Management System, BSI.

[2] DMTF CIM Policy Model v. 2.81, available at www.dmtf.org (Feb. 2005).

[3] Gruber T., "Toward principles for the design of ontologies used for knowledge sharing". In Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic Publishers, 1993.

[4] Clemente F., Perez G., Blaya J., Skarmeta A., "Representing Security Policies in Web Information Systems", Policy Management for the Web Workshop, 14th International World Wide Web Conference, May 2005, Japan.

[5] ISO/IEC17799, Information technology – Code of practice for information security management, ISO.

[6] Standards Australia and Standards New Zealand, Australian/New Zealand Standard for Risk Management 4360 (1999).

[7] United Kingdom Central Computer and Telecommunication Agency. CCTA Risk Analysis and Management Method: User Manual, ver. 3.0. HMSO.

[8] Noy N., McGuiness D., "Ontology Development 101: A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory Technical Report KSL-01-05.

[9] Holsapple C., Joshi K., "A collaborative approach to ontology design", Com. of the ACM, 45(2):42-47, 2002.

[10] Tsoumas B., Dritsas S., Gritzalis D. "*An ontology-based approach to information system security management*", 3rd International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security, September 2005, Russia.

[11] Quirolgico S., Assis A., Westerinen A., Baskey M., Stokes E., "Toward a Formal Common Information Model Ontology", in Proc. of WISE, pp 11-21, 2004.

[12] Protégé Ontology Development Environment, available at http://protege.stanford.edu/ (Sep. 2005)

[13] Protégé OWL Plugin, available at http://protege.stanford.edu/plugins/owl/ (Sep. 2005)

[14] COBIT 3rd Edition Control Objectives, IT Governance Institute, 2000.

[15] BSI, IT Baseline Protection Manual, Germany available at www.bsi.bund.de/ (Mar. 2005).

[16] Cisco Security Advisories, available at www.cisco.com/go/psirt/ (Mar. 2005).

[17] SecurityFocus security portal, available at www.securityfocus.com (Mar. 2005).

[18] Org Security Mailing List Archive, available at www.seclists.org (Mar. 2005).

[19] Common Vulnerabilities and Exposures, available at www.cve.mitre.org/ (Mar. 2005).

[20] Nmap scanner, available at www.insecure.org/nmap (Mar. 2005).

[21] Netstumbler 802.11 network scanner, available at www.stumbler.net (Mar. 2005).

[22] Cunningham H., et al., "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications". Proc. of the 40th Conference of the Association for Computational Linguistics. USA, July 2002.

[23] Damianou N., et al., "The Ponder Policy Specification Language". In Workshop on Policies for Distributed Systems and Networks, Springer-Verlag (LNCS 1995), 2001, pp. 18-39.

[24] Kagal L., et al., "A policy language for a pervasive computing environment". In 4th IEEE International Workshop on Policies for Distributed Systems and Networks, 2003.

[25] Lymberopoulos L., Lupu E., Sloman M., "Ponder Policy Implementation and Validation in a CIM and Differentiated Services Framework". NOMS 2004, Seoul, 2004.

[26] Alcantara O., Sloman M., "QoS policy specification - A mapping from Ponder to the IETF", Dept. of Computing, Imperial College, UK.

[27] Dean M., et al., OWL Web Ontology Language Reference W3C Recommendation, http://www.w3.org/TR/owl-ref/ (Mar. 2005)

[28] Donner M., "Toward a Security Ontology", In IEEE Security and Privacy, Vol. 1, No. 3, pp. 6-7, May 2003.

[29] Denker G., Access Control and Data Integrity for DAML+OIL and DAML-S, SRI International, USA, 2002.

[30] Denker G., Security Mark-up and Rules, SRI International, CAIn: Dagstuhl Seminar on Rule Markup Techniques, 2002.

[31] OASIS Security Service TC. Security Assertion Markup Language (SAML), www.oasis-open.org/committees/security/ (Mar. 2005)

[32] XACML Specification (2003), eXtensible Access Control Markup Language, v. 1.1 available at www.oasis-open.org (Mar. 2005).

[33] Bozsak E., Ehrig M., Handschub S., Hotho J., "KAON – Towards a Large Scale Semantic Web", in: Bauknecht, K., et al. (Eds.): Proc. of the 3rd International Conference on e-Commerce and Web Technologies, 2002, pp. 304-313.

[34] Raskin V., et al., "Ontology in Information Security: A Useful Theoretical Foundation and Methodological Tool". In V. Raskin, et al. (Eds.), Proc. of the New Security Paradigms Workshop, ACM, USA, 2001.

[35] Uszok A., et al., "KAoS: A Policy and Domain Services Framework for Grid Computing and Semantic Web Services", Proc. of the 2nd International Conference on Trust Management, 2004.

[36] Tonti G., et al., "Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei and Ponder", Proc. of the 2nd International Semantic Web Conference, 2003.

[37] Gandon F., Sadeh N., "Semantic web technologies to reconcile privacy and context awareness". Web Semantics Journal, 1(3), 2004.

[38] Chen H., et al., "SOUPA: Standard ontology for ubiquitous and pervasive applications", Proc. of the 1st International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004.