Computer Standards & Interfaces 33 (2011) 372–388

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Basis for an integrated security ontology according to a systematic review of existing proposals

Carlos Blanco [a], Joaquín Lasheras [b], Eduardo Fernández-Medina [c,*], Rafael Valencia-García [b], Ambrosio Toval [b]

[a] Department of Mathematics, Statistical and Computation, Facultad de Ciencias, University of Cantabria, Av. De los Castros, s/n — 39071, Santander, Spain
[b] Software Engineering Research Group, Computer and Systems Department, University of Murcia, Campus de Espinardo, 30071, Murcia, Spain
[c] Department of Information Technologies and Systems, Escuela Superior de Informática, GSyA Research Group, University of Castilla-La Mancha, Paseo de la Universidad, 4 — 13071, Ciudad Real, Spain

ARTICLE INFO

Article history:
Received 22 July 2009
Received in revised form 21 December 2010
Accepted 26 December 2010
Available online 8 January 2011

Keywords:
Ontologies
Security
Systematic review

ABSTRACT

The use of ontologies to represent knowledge provides us with organization, communication and reusability. The concepts and relations managed by any scientific community need to be formally defined. Since security in information technologies has evolved as a critical aspect and many related topics have been developed, this paper applies the method of systematic review for identifying, extracting and analyzing the principal proposals for security ontologies. The most mature proposals have been selected and compared by using a formal framework, extracting the key requirements that an integrated and unified security ontology should have, and providing the first steps towards its definition.

1. Introduction

An ontology is a specification of a conceptualization [1]. It is a simplified abstract view of reality which contains the objects, concepts and relations that interest us. It represents knowledge in a formal and structured form, and provides better communication, reusability and organization of knowledge along with a better computational inference [2–4].

The principal objective of ontologies is to establish ontological agreements that will serve as a basis for communication between either human or software agents, thus the decreasing language ambiguity and knowledge differences between these agents which may lead to errors, a lack of understanding and unproductive efforts. The use of ontologies also provides other advantages such as the filtering and inference of knowledge or the validation of consistency. In other words, models and meta-models are abstractions of a part of the reality, and ontologies are a basic component of these meta-models that support their construction by expressing what can or cannot be included, and permit the definition of axioms which support the inference of knowledge or the validation of consistency.

Moreover, any research community manages a great deal of concepts and relations which need to be formally defined, and ontologies help us to organize our knowledge, to report incidents in an effective manner and to share information with other organizations. For example, sharing explicit knowledge in Software Engineering is often extremely difficult, and may lead to a waste of time and effort. Researchers are, therefore, currently working on knowledge integration and its support with software tools [5,6], and ontologies are a good approach by which to support this.

We can thus conclude that each community should profit from the advantages provided by the use of ontologies and should attempt to reach a consensus regarding the main concepts that they manage. In fact, in most scientific communities it is frequently possible to find standards which can serve as a basis for the construction of an ontology [7,8]. Section 4.1 provides a more in-depth discussion regarding the obtaining of general ontologies for different scientific fields [9] and the main difficulties involved in achieving this goal [10].

Information security is also a vital aspect for the development of information systems [11] and the survival of enterprises [12], and a security community manages a great deal of concepts and relations which need to be supported by an ontology that provides a support through which to develop methods, processes and methodologies [7,13].

IT security has undergone a spectacular evolution in the scientific community, and the number of events and journals focused on security has increased dramatically. Security is also identified in digital bibliography databases (such as DBLP) as being the most popular keyword in computer science journals and proceedings, thus signifying that security is one of the most active scientific disciplines, and making the existence of an ontology which clearly defines, classifies and links the related concepts is highly important.

[*] Corresponding author.
E-mail addresses: carlos.blanco@unican.es (C. Blanco), jolave@um.es (J. Lasheras), Eduardo.FdezMedina@uclm.es (E. Fernández-Medina), valencia@um.es (R. Valencia-García), atoval@um.es (A. Toval).

0920-5489/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2010.12.002

Information assurance, security and privacy have thus moved from being considered by information systems designers as narrow topics of interest to becoming critical issues of fundamental importance in our society [14]. These are serious requirements which must be carefully considered, not as isolated aspects, but as elements that must be present in all stages of the development lifecycle, from requirements analysis to implementation and maintenance [15–21].

The application of ontological engineering to IT security provides us with better knowledge organization and mechanisms for the prediction of security problems [22]. Many examples of this have thus been applied to the semantic web [2,14], privacy preservation in trust negotiations [23], security risk management [24], computer attacks [25], etc. The need for a complete security ontology in the security community has been identified as an important challenge and research branch [21].

We have therefore carried out a literature review by using the systematic review approach proposed by Kitchenham [26,27] with the objective of obtaining a solid background with regard to security ontologies. This work is the evolution of a preliminary review which was published in [28]. A significant amount of this paper is therefore devoted to formally defining the review and its execution in order to discover, analyze and compare the most relevant proposals concerning ontological engineering when applied to security. As a conclusion of this study, we have identified that although these security ontologies make important contributions to the security community, they only offer partial solutions to the integration of their knowledge into an integrated security ontology. What is more, we have identified that the accomplishment of this general integrated ontology is a difficult and complex task that requires broad discussion and consensus within the scientific and professional community. This paper is therefore intended to be as a starting point by which to attain this goal. We have therefore identified a set of key requirements which must be fulfilled by an integrated security ontology, starting from a formal and contrasted ontology comparison framework, which we have identified as the best means to achieve this aim in the ontological field. We have then analyzed whether the current approaches fulfill these key requirements, and we have made certain proposals concerning how an integrated security ontology could be defined in the form of guidelines and recommendations regarding aspects of content and relations of the ontologies, along with other technological aspects related to the formal characteristics of the ontologies and their support tools. This set of recommendations may also serve to define new specific research in this line of work.

The remainder of the paper is organized as follows: Section 2: shows how the review was planned by defining the research question; describes how the review was carried out; defines the data that we wished to extract and synthesizes the information obtained from the most relevant studies. Section 3 compares the most mature ontologies of these studies, considering a formal ontology comparison framework, which compares their components and dimensions (principally the content dimensions, which include concept, relation, taxonomy and axiom factors) and analyzes the results obtained. In Section 4 we discuss the difficulties involved in defining an integrated ontology and identify a set of key requirements for an integrated security ontology we compare the current approaches, analyzing how they fulfill these key requirements, and finally we offer a set of ideas and recommendations for the definition of this ontology. Finally, Section 5 sets out our conclusions.

2. Systematic review

The realization of a literature review through the use of a systematic review permits all the significant studies (primary studies) related to a given research question to be identified, evaluated and analyzed. A systematic review provides several advantages: it allows the existing evidence related to a treatment or technology to be summarized, any gaps in present-day investigation to be identified, thus suggesting areas of future investigation, and a framework to be provided in which new research activities can be appropriately positioned.

A systematic review is based on a defined research strategy that attempts to detect all relevant literature. Explicit inclusion and exclusion criteria are needed to evaluate each potential primary study and to specify the information of each primary study, including quality criteria. This systematic review has been performed on the basis of the guidelines proposed by Kitchenham [27], which are appropriate for software engineering researchers. We have also used a review protocol template developed by Biolchini [29] which facilitates systematic review planning and execution in software engineering.

Our systematic review consisted of several stages. We begin by showing how the review was planned, along with identifying its needs and defining its protocol. We then describe how the planned review was applied in order to obtain the set of primary studies and how the relevant information was extracted from them. The main proposals are compared by using a formal framework, and we conclude by stating their early state of development and the need for additional research efforts.

2.1. Review planning

In this phase, we define the research objectives and the way in which the review will be carried out. This includes both the formulation of research questions and the planning of how the selection of sources and studies will take place.

2.1.1. Question formularization

In this section, the research objectives are clearly defined. The question focus is that of identifying the most relevant works centered on the development of ontologies that deal with security issues. The research question that was addressed by our review is, therefore, as follows: What initiatives have been carried out to develop security ontologies in the field of ontological engineering? Table 1 shows the keywords and related concepts that were used to formulate this question and which were used during the review. These keywords have been considered to be those which were most representative in discovering those papers that assisted us in answering our research question.

In the context of the planned systematic review, the proposals concerning security ontologies were observed, analyzed and compared. The population group that was observed thus consisted of publications in the selected data sources.

The expected result at the end of this systematic review was the identification of initiatives related to security ontologies. The outcome measures are the initiatives that were identified grouped by area and the comparison of the main proposals. The main application area that will benefit from the results of this systematic review is that of

Table 1
Keywords.

Area	Keywords	Related concepts
Ontologies	Ontology OWL RDF DAML	Ontological Engineering
Security	Security	Security engineering Privacy Confidentiality Integrity Availability Authentication Non repudiation

ontological engineering when applied to security, and the specific people who will benefit are academics, researchers and professionals.

The meta-analysis of the review is focused on analyzing the contributions of the security ontologies proposed in the primary studies by, on the one hand, discovering the current contributions and analyzing areas of interest and, on the other hand, comparing the most significant ontologies identified through the use of a formal comparison framework [30]. This will allow us to obtain a vision of the current situation and thus detect deficiencies and situate future research.

2.1.2. Source selection

The objective of this phase was to select the sources in which to carry out searches for primary studies. The selection criteria used to evaluate the study sources was based on the research experience of the authors of this work, and these sources were selected by considering certain constraints: studies included in the selected sources had to be written in English and these sources had to be Web available and possess search engines that would allow us to execute advanced search queries. The following list of sources was considered: Science Direct, ACM digital library, IEEE digital library, SCOPUS, Scholar Google, DBLP. The experts considered that this list of digital sources indexes the most relevant research from journals, conferences and book chapters. Nevertheless, the results obtained after executing the review was refined by a manual process, in which other sources (such as web pages of communities and research groups, book chapters not indexed in the previously listed sources, etc.) were queried in order to incorporate relevant works that it was not possible to identify during the execution of the review.

2.1.3. Studies selection

Having defined the sources, it is now necessary to describe the process and the criteria used for study selection and evaluation. Firstly, as is shown in Table 2, we combined the selected keywords with AND and OR connectors to obtain our search chain.

The procedure for study selection began with the adaptation of our search chain to the syntax of each search engine, and its execution. We then obtained a set of results to which the inclusion criteria were applied in order to make an initial selection of studies which were potential candidates for primary studies. This criterion was focused on analyzing titles, keywords and abstracts of the studies in order to discover how the concepts of ontological engineering when applied to security are related. We thus discovered relevant proposals and discarded most of the irrelevant studies. Security standards were not considered to be security ontologies in the review because they are plain taxonomies of concepts and are used in the development of these ontologies.

The exclusion criterion was then applied to the set of relevant studies in order to obtain a set of primary studies. In this stage, the set of candidate studies was analyzed in depth, focusing on abstracts, conclusions and other sections, in order to detect which proposals are genuinely important contributions in the field of ontological engineering when applied to security, i.e. works which define security ontologies for a general purpose or for a specific domain and theoretical works which discuss the importance of an integrated security ontology and provide proposals through which to achieve this.

Table 2
Search chain.

(Ontology OR (Ontological and engineering) OR OWL OR RDF OR DAML And (Security OR (security and engineering) OR privacy or confidentiality or integrity or availability or authentication or (non and repudiation))

2.2. Review execution

During this phase, it is necessary to execute the search in the defined sources and to evaluate the obtained studies according to the established criteria. After executing the search chain on the selected sources we obtained a set of about 300 results which were filtered with the inclusion criteria to give a set of about 100 relevant studies. This set of relevant studies was again filtered with the exclusion criteria to give a set of studies consisting of 31 primary proposals.

The obtained studies which completely fitted all the previously defined inclusion and exclusion criteria are shown in Table 3, and are classified into three kinds of proposals: (1) contributions which attempt to define security ontologies with a general purpose; (2) contributions focused on security ontologies for a certain application domain; and (3) theoretical works which deal with the importance of an integrated ontology in the security community and how to approach the problem. In the following section, the information relevant to the research question is extracted from the selected studies.

2.3. Information extraction

Once the primary studies have been selected, relevant information is extracted and results are described. In order to standardize the way in which this information will be presented, we have created a means by which to collect data from the selected studies. The forms of information defined for this review are made up of three components: basic information (authors, title, publication, and reference in APA

Table 3
Primary studies classified by area.

Proposal	Security ontologies general	Security ontologies specific domain	Theoretical works
Basile et al. [31]		Policy-based network management	
Beji and Kadhi [32]		Mobile applications	
Denker et. al. [14,34,35]	X		
Dobson et al. [2,36]		QoS constraints	
Donner [7]			X
Fenz et al. [24,37–43]	X		
García-Crespo et al. [46]		Access control	
Geneitakis and Lambrinoudakis [92]		VoIP services	
Herzog et al. [33]	X		
Kagal and Finin [93]		Conversation policies	
Karyda et al. [47]	X		
Kim et al. [48,49]	X		
Kwon and Moon [94]		RBAC	
Lee et al. [50]	X		
Li and Wang [51]		Auditing	
Liu and Lee [52]		Network	
Maamar et al. [95]		Web services	
McGibney et al. [96]		Intrusions	
Mouratidis et al. [53–56]		Social aspects	
Mouratidis and Giorgini [21]			X
Parkin et al. [58]		Human-oriented security issues	
Raskin et al. [22]			X
Tan and Poslad [97]		Security reconfiguration	
Thuraisingham [98]		Security standards	
Tsoumas et al. [13,59,60]	X		
Undercoffer et al. [25]		Vulnerabilities	
Vorobiev et al. [63,64]		Vulnerabilities	
Wang et al. [65]		Vulnerabilities	
Yu et al. [57]		Social aspects	
Zhou et al. [66]		Reliability	
Zhou et al. [68]	X		
Total	8	20	3

format), general description (study area and summary) and our general impressions and comments (details related to the composition of the proposed ontology, information regarding security standards, language and methodology used to define the ontology, etc.). The selected areas which were used to classify the studies are as follows: security ontologies (general), security ontologies applied to specific domains and theoretical works.

We shall now present a brief outline of each of the studies selected in the previous section according to the extracted information obtained from the forms. We have limited the scope of this study and we focus on security ontology proposals.

2.3.1. Basile et al., "Ontology-based Security Policy Translation" [31]

The main goal of this ontology is to model concepts within the field of policy-based network management and to provide reasoning in order to generate configurations for security controls by using Access Control Lists (ACL) and secure channels.

This ontology has been organized into three levels. The first includes the main instances which are obtained from processing external files. The instances in the other levels are obtained by applying a reasoning process. The second level classifies computers into workstations or servers, and the third level classifies them into shared or personal. Properties regarding concepts have also been included in order to allow deduction.

2.3.2. Beji and Kadhi, "Security Ontology Proposal for Mobile Applications" [32]

The field of mobile applications' development requires the consideration of security constraints, but standards have not been established.

The authors analyze the most relevant concepts related to this field (Vulnerabilities, Threats, Assets, Constraints, Actors, Mechanisms, Resources, Services and Value Types) and define an ontology in OWL-DL to conceptualize the main concepts related to mobile applications and the relations between actors and security goals.

This ontology is based on the security for information systems ontology defined by Herzog et al. in [33] and is composed of three sub-ontologies: an Asset-Vulnerability-Threat ontology (AVTo); a Mobile-Profile ontology (MPo); and a DefenseMechanism ontology (DMo).

2.3.3. Denker et al., security ontology to annotate web services [14,34,35]

These authors develop an ontology with which to annotate web services by including well-known security concepts that enable security standards to be interconnected. This work was first developed by using DAML [34] and later by using OWL-S (http://www.daml.org/services/owl-s/) [14,35]. OWL-S uses agent technology to enable the automation of services on the Semantic Web, and it is supported by tools that permit editing (plug-in for Protégé, etc.), matching, validating, visualizing, etc.

OWL-S is made up of three main parts: a service profile with which to advertise and discover services; the process model, which provides a detailed description of a service's operation; and the grounding, which provides details on how to interoperate with a service via messages. This ontology, denominated as "OWL-S Security and Privacy", contains two sub-ontologies related to security: "security mechanisms", which captures high-level security notations, and "credential", which defines authentication methods.

2.3.4. Dobson et al., Ontology for non-functional requirements "QoSOnt" [2,36]

In these works, the authors review ontologies for Requirements Engineering and propose an ontology for non-functional requirements which can be reused in different domains. This ontology, called "QoSOnt", provides a conceptual model with which to describe quality of service (QoS) constraints by using non-functional attributes and metrics, and also includes semantic rules with which to, for example, automatically convert units and metrics.

As a part of their main goal, the authors use OWL to define an ontology focused on the field of dependability requirements which is compliant with the IFIP Working Group 10.4 taxonomy and includes security issues such as "dependability", "reliability", "availability", "integrity", "confidentiality" or "safety".

2.3.5. Fenz et al., Ontology for IT-Security based on risk management "SecurityOntology" [24,37–43]

In [37,39], the authors study security in small and medium size enterprises (SMEs) and propose a holistic solution based on a security ontology ("SecurityOntology") that includes low-cost risk management and threat analysis.

This security ontology is based on Landwehr's security and dependability classification [44]. It is used to allow organizations to capture the business knowledge required, and it is created during a security risk analysis in which instances of concepts are added to the ontology to permit the simulation of various attack and disaster scenarios, always focusing on the Infrastructure asset. Each scenario can be replayed with a different protection profile so as to evaluate the effectiveness and the cost/benefit ratio of individual safeguards.

This security ontology is available at (http://securityontology.securityresearch.at/) and can be browsed and edited on-line by using web protégé (http://sec.sba-research.org/webprotege/). The ontology consists of five sub-ontologies. The main sub-ontology is "threat", and includes appropriate countermeasures, threatened infrastructures and suitable evaluation methods. The "Security Attribute" sub-ontology models the impact of threats, "Infrastructure" describes infrastructure elements, "Role" maps enterprise hierarchies and "Person" represents the natural people who are relevant to security issue modelling. In [40] the threat simulation approach is extended with risk analysis methods in order to improve quantitative risk analysis, and in [24] the authors develop a tool called AURUM (Automated Risk and Utility Management) in order to support decision makers in selecting security measures according to technical and economic requirements. Furthermore, in [38,43] the authors propose a methodology based on their Security Ontology and the Security Standard ISO/IEC 27001 [45], which allows security metrics to be generated automatically, and organizations to evaluate their compliance with security standards, thus enhancing their overall IT security level.

2.3.6. García-Crespo et al., "SecurOntology: a semantic web access control framework" [46]

This work focuses on the specification of access control security policies by using an ontology defined in OWL-DL. It formally describes RBAC policies and is complemented with an architecture which permits its application to typical web systems.

SecurityOntology is composed of: classes which represent the main concepts (resources, owners, roles, permissions, permissions associated with specific resources, and consults); properties which establish relations between concepts (hasRole, isOwnerOf, itsOwnerIs, hasPermission, hasChild, isChildOf, resource and permission); and rules defined in SWRL which allow new knowledge to be inferred.

2.3.7. Herzog et al., "An Ontology of Information Security" [33]

The authors of this proposal use OWL to develop an ontology for information security which is available on-line (http://www.ida.liu.se/~iislab/projects/secont/). This ontology is mainly focused on assets, threats and countermeasures. Nevertheless, it also includes concepts concerning access control models (RBAC, MAC, DAC, etc.), cryptographic models, security goals (authentication, confidentiality and integrity goals), vulnerabilities, products (databases, Operating Systems, etc.), strategies and so on.

The main classes, “Asset”, “Threat” and “Countermeasure”, are defined in greater detail. For example, “Asset” considers kinds of Networks, electronic tokens (encryption keys, passwords, cookies, etc.), biometric tokens (fingerprint, retina, voice, etc.), physical tokens, kinds of hardware, hosts, etc.

Moreover, relationships between the ontology's concepts are defined, thus providing us with reasoning capabilities such as the establishment of the countermeasures associated with specific threats.

This ontology can be extended with new elements; the authors therefore provide several examples of how further extensions can be achieved: an ontology to define sorted views for threats and countermeasures; an ontology which includes tools used to analyze C source code; and an extension of the memory protection countermeasure achieved by adding several tools and methods.

2.3.8. Karyda et al., “An ontology for secure e-government applications” [47]

In this work, the authors use OWL to propose a security ontology with which to develop secure applications. It captures the security knowledge of experts in order to support communication between the security experts, users and developers who use it to both include security requirements and to support design choices. The authors develop their ontology by applying four stages in an iterative process: they determine questions of competency, enumerate important terms, define classes and hierarchies, and instantiate. The proposed ontology is formed of “assets” (data asset, hardware data, etc.), “countermeasures” (identification and authentication, network management, auditing services, physical protection, etc.), “objectives”, “persons” (insider stakeholder, attacker, etc.) and “threats” (errors, attacks, technical failures, etc.). They validate the defined ontology by using nRQL queries, and demonstrate that their ontology can be used in various contexts by applying it to e-government scenarios: e-tax and e-voting.

2.3.9. Kim et al., security ontology for annotating resources “NRL” [48,49]

These authors use OWL to develop the “NRL” security ontology which focuses on the annotation of functional aspects of resources. This ontology is capable of representing security statements such as mechanisms, protocols, algorithms and credentials, and can be applied to any electronic resource.

“NRL” presents an architecture which is easy to use and easy to extend, and is made up of seven sub-ontologies. Three of them are based on existing ontologies in DAML: firstly, the “Service security ontology” describes the security annotation of semantic web services; secondly, the “Agent security ontology” enables the querying of security information; and finally the “Information object ontology” describes the security of Web service input and output parameters.

The four remaining ontologies are as follows: the “Main security ontology”, which describes security protocols, mechanisms and policies; the “Credentials ontology”, which specifies authentication credentials; the “Security algorithms ontology”, which describes various security algorithms; and the “Security assurance ontology”, which specifies different assurance standards. In [49] this ontology is applied to a Service Oriented Architecture to annotate security aspects of Web service descriptions and queries.

2.3.10. Lee et al., “Building Problem Domain Ontology from Security Requirements in Regulatory Documents” [50]

In this paper, the authors identify security requirements for certification and accreditation activities which are expressed in regulatory documents. These requirements are of a non-functional nature which imposes complex constraints on the behavior of software systems and makes them difficult to understand, predict and control. The authors present a framework which includes techniques extracted from software requirements engineering and knowledge engineering, and they propose a common language with which to extract concepts from regulatory documents. This methodology is applied to the construction of a problem domain ontology from regulatory documents enforced by the DITSCAP — Department of Defense Information Technology Security Certification and Accreditation Process.

2.3.11. Li and Wang, “Applications of Ontology in Management of Information Asset” [51]

The authors develop an ontology with which to manage information assets, focused on auditing. It combines auditing regulations in SWRL and adds inference capabilities that help auditors to know the status of information assets and discover problems.

2.3.12. Liu and Lee, “Constructing Enterprise Information Network Security Risk Management Mechanism by Ontology” [52]

This ontology defines information security risk management based on the ISO 27001 security standard. It is composed of three sub-ontologies: a task ontology which defines the problems to be solved; a resolution ontology which specifies problem solving methods; and a domain ontology which includes information security and domain knowledge concepts. Security management rules have been additionally defined over relations between concepts, thus allowing inferences to be attained.

2.3.13. Mouratidis et al., an Ontology for Modelling Security with Tropos [53–56]

The Tropos methodology considers two approaches in software development: a security-oriented process and a management of a trust-oriented process. This methodology is based on social hierarchies and adapts components of the i* framework [57], which uses the concepts of actors, goals, tasks, resources and social dependencies to define the obligations of actors (dependees) towards other actors (dependers). The authors improve the social ontology created for the i* framework with new security concepts: constraints, secure entities (secure goals, tasks, resources, ownership) and secure dependences between actors (such as trust of execution, trust of permission, delegation of permission and delegation of execution). The result is a methodology which considers the issues of security and trust as a part of its development process [53–55]. A case study from the health domain is employed to illustrate the approach. The authors have recently proposed a framework which attempts to align the security concepts used in laws and regulations with the terminology used in requirements engineering [56].

2.3.14. Parkin et al., “An information security ontology incorporating human-behavioural implications” [58]

This ontology represents human-oriented security issues and can be used by security managers to assist in information security decisions. It is based on the ISO 27002 security standard, which is related to human behavior, and permits the identification of their effects on information security. The ontology has been developed by using OWL and is composed of the following main concepts: objects from the ISO 27002 security standard (chapter, section, guideline and guideline step); information assets identified as being critical for the security management process; roles responsible for security assets' maintenance; vulnerabilities with assets' weakness; threats which exploit vulnerabilities; and behavior controls which mitigate vulnerabilities.

2.3.15. Tsoumas et al., an Ontology for SPIT management “OntoSPIT” [13,59,60]

In this proposal, the authors describe a security framework of an arbitrary information system which provides security acquisition and knowledge management. This framework is based on a security ontology which extends the DMTF Common Information Model (CIM) (www.dmtf.org) with ontological semantics in order to use it as a container for IS security-related information. This Security ontology is based on security and risk management practices such as CRAMM [61] or COBIT [62]. This proposal describes 4 phases with which to establish the IS security management framework, the first being the "building of the security ontology". As further work they envisage the development of a standards-based, best practices database with implicit security knowledge to support information extraction and the decision making process which will consider semantic rules and the ontologies' properties of reusability and interoperability. The authors have recently developed an ontology called OntoSPIT [60] which they hope to incorporate into more general security ontologies, despite its being focused on the modelling and management of spam attacks on VoIP communications.

2.3.16. Undercoffer et al., "Modelling Computer Attacks: An Ontology for Intrusion Detection" [25]

Here, the authors first analyze approximately 4000 vulnerabilities and their exploit strategies, and they then use DAML + OIL and DAMLJessKB to create an "IDS Ontology" with which to specify computer attacks. In this paper, the authors also summarize the main languages for specifying computer attacks: P-Best, STATL, LogWeaver, CISL, BRO, Snort Rules and IDMEF and present several use case scenarios with common attacks: "Denial of Service — Syn Flood", "The Classic Mitnick Type Attack" and "Buffer Overflow Attack".

2.3.17. Vorobiev et al., "Security Attack Ontology for Web Services" [63,64]

The authors develop several ontologies related to vulnerabilities by considering previously defined ontologies such as "OWL-S Security and Privacy" and "NRL".

This work presents a high level ontology for security assets-vulnerabilities (SAVO) which is oriented towards non-security professionals. They also define ontologies for: security algorithms-standards (SASO) with concepts such as security algorithms, standards, credentials, assurance levels, etc.; security functions (SFO) related to SASO elements; security attacks (SAO); and defences (SDO) with specific countermeasures to avoid them.

2.3.18. Wang et al., "An Ontological Approach to Computer System Security" [65]

The authors present an ontology focused on vulnerability management which includes all the vulnerabilities detected by NVD (http://nvd.nist.gov/scap.cfm). It also includes inference and knowledge discovery capabilities.

2.3.19. Yu et al., "A Social Ontology for Integrating Security and Software Engineering" [57]

The authors integrate security into a requirement driven development process. This proposal deals with the social context of security by using a social ontology based on i*, an agent framework oriented towards model systems which considers intentional aspects such as goals, softgoals, tasks, resources and beliefs. It also includes models with which to describe relations between actors and to support the reasoning of each actor in his/her relations with other actors.

2.3.20. Zhou et al., an Ontology Approach focused on Reliability "OntoArch" [66,67]

The authors propose an ontology-based method for software reliability modelling called OntoArch, which includes a software reliability ontology developed in OWL together with an ontology-based software modelling system. They describe reliability engineering as a series of interrelated processes by which reliability knowledge is reorganized with the support of methods, tools, models, organization, and the specifications of input and output. The authors have additionally validated OntoArch by applying it to the design of a system architecture for a Personal Information Repository.

2.3.21. Zhou et al., "An Integrated QoS-Aware Service Development and Management Framework" [68]

This work proposes a method for management and service quality assurance (QoS-aware) which consists of a QoS-aware service management infrastructure, a QoS ontology and a QoS property ontology. The QoS ontology provides us with a knowledge mapping with QoS concepts and relations that can be used for QoS-aware service communication and exchange. The QoS property ontology has two sub-ontologies: "Technical QoS property", which defines concepts and relations related to software development and "Managerial QoS property" which focuses on service provision.

2.4. Conclusions of the systematic review

After carrying out the systematic review, the results were summarized and analyzed by using the methods defined during the planning phase. Table 3 presents the primary studies classified according to the purpose of the work, which was: to define a general purpose security ontology, to define a security ontology focused on a certain domain (in this case the domain has been indicated in Table 3), or theoretical works which, despite making interesting contributions, do not formally define an ontology. As can be seen in Table 3, although security ontologies with a general purpose are being developed, the majority of security ontologies found in this review are focused on formalizing a concrete domain that is necessary to solve a specific problem, since formalizing all the concepts in the security domain has been identified as a difficult task.

Table 4 shows a detailed summary of the works selected in this review. The following aspects are analyzed: whether the ontology has been conceived with a general purpose or only attempts to represent a specific domain; aspects related to the design of the ontology, particularly whether development models or standards have been used; whether security standards (such as ISO/IEC 27001 [45], ISO/IEC 15408-1999 [83], etc.) or best practices (such as MAGERIT [74], CRAMM [61], OCTAVE [84], COBIT [62]) have been considered; the language used to formalize the ontology; whether the ontology is available on-line or is under development; and the main contributions provided by the ontology.

Although the majority of the security ontologies found are focused on specific domains, we can observe that most of them have been developed over a general approach, which considers the importance of a general purpose security ontology and allows the further extension of the ontology. Furthermore, whereas many ontologies have considered standards or best practices in security (ISO/IEC 27001 [45], CRAMM [61], COBIT [62], IFIP WG 10.4, etc.), very few works clearly identify the use of a model or standard for the ontology's development. Since security standards are plain taxonomies of concepts, ontologies can establish a semantic layer into which these standards can be integrated and improved with more features such as knowledge inferences. Finally, the majority of general purpose security ontologies have been defined by using OWL and are available on-line, thus allowing an active participation and feedback from the security community. A good example is the Security Ontology defined by Fenz [24,37–43] which can be edited on-line, thus supporting the participation of the community in their definition.

The systematic review has led us to observe that obtaining a general security ontology has been identified as a necessity. However, despite the fact that current ontology proposals make important contributions, they do not solve the problem of obtaining an integrated security ontology. This is not an easy task (there is no previous work dealing with this subject). Nevertheless, we consider that existing security ontologies can be used as a general basis for reuse thanks to their properties of shareability and reusability,

Table 4
Security ontology proposals: summary of contributions.

Proposal	General	Model or standard of development	Integration of security standards and best practices	Ontology language	Availability	Main contributions
Denker et. al. [14,34,35]	Yes	Web service descriptions in OWL-S (WSDL)	Xml signature, SAML, WS-security	OWL	Y	Security annotations of agents and web services
Dobson et al. [2,36]	No	IFIP dependability model and UMD	IFIP working group 10.4 taxonomy	OWL	Y	Ontology for dependability requirements engineering
Fenz et al. [24,37–43]	Yes	ROPE for business processes	COBIT, ISO 17799 ISO 27001	OWL-DL	Y	Ontology focuses on low-cost risk management and threat analysis
García-Crespo et al. [46]	No	Not identified	RBAC, SWRL	OWL-DL	N	Access control framework based on RBAC
Geneitakis and Lambrinoudakis [92]	No	Not identified	None	DAML + OIL	N	Describes vulnerabilities for VoIP services based on the SIP architecture
Herzog et al. [33]	Yes	Not identified	None	OWL-DL	Y	Ontology for information security focused on assets, threats and countermeasures
Karyda et al. [47]	Yes	Top–down approach	None	OWL y NRQL	N	Ontology for developing security critical applications
Kim et al. [48,49]	No	Not identified	Security standards in web services	OWL	Y	NRL security ontology for annotating resources
Lee et al. [50]	Yes	Not identified	DITSCAP	OWL (GenOM toolkit)	N	Ontology for security requirements in regulatory documents
Liu and Lee [52]	No	Not identified	ISO 27001	OWL	N	Network
Mouratidis et al. [53–56]	No	TROPOS methodology (i* framework)	None	Formal TROPOS grammar	N	Modelling security with the TROPOS methodology
Parkin et al. [58]	No	Not identified	ISO 27002	OWL	N	Human-oriented security issues
Tsoumas et al. [13,59,60]	Yes	Extends the standard DMTF	CRAMM COBIT	OWL	N	Framework for security acquisition and knowledge management
Undercoffer et al. [25]	No	Not identified	Languages for computer attacks	DAML + OIL	Y	Security ontology to specify computer attacks
Yu et al. [57]	No	Business process modelling and redesign and software process modelling	None	Not implemented	N	i* framework for modelling and reasoning requirements about organizational environments
Zhou et al. [66,67]	No	Not identified	None	OWL	N	Software reliability ontology
Zhou et al. [68]	Yes	Not identified	None	OWL	N	QoS-aware method based on an ontology

although it is first necessary to identify whether these proposals are adequate, and, to extract the key requirements that must satisfied in order to obtain an integrated and unified security ontology.

We must emphasize that in order to identify these key requirements, in the ontological field, is necessary to start from a formal and contrasted ontology comparison framework (in our case OntoMetric, the choice of which is justified in next Section 3) in order to formally discover the features of the proposals. What is more, we have also detected that, specifically in the field of security, criteria with which to compare an integrated ontology does not exist because the only things that exist are standards and best practices which do not deal with security from a global point of view and, in any case, help as taxonomies of concepts, but do not contribute with other aspects that ontologies may contain (concepts, relations, axioms, ontological agreements …).

The following section therefore compares the set of identified proposals, by using a specific framework for ontologies, which allows us to discover how well these ontologies are defined, and how they could be integrated and reused. However, owing to the lack of detailed information regarding these proposals, only those which are most mature (and which are available on-line) have been selected for the comparison. Finally, in the following sections an initial proposal for an integrated security ontology will be presented.

3. Comparison of the ontologies

Various approaches for the measurement and evaluation of ontologies have been considered in literature, and these are distributed in two major categories [69]. On the one hand, a few approaches exist which are based on the manual assessment of a set of ontology design criteria (e.g. OntoMetric [70,71]). On the other hand, there are many automatic approaches (e.g. the approaches compared in [72]), which evaluate different aspects of an ontology (e.g., vocabulary, conceptual structure) by relying on different views of what constitutes a good "quality" ontology. In this work, we intend to perform a manual comparison of ontologies. Various approaches exist, depending on what kinds of ontologies are being evaluated and for what purpose. For example, in [73] a classification of the most important current approaches for evaluating and comparing ontologies is carried out by using six evaluation levels: lexical, vocabulary; hierarchy, taxonomy; other semantic relations; context, application level; syntactic; and structure, architecture, design. The OntoMetric method [70,71] is a specific method with which to evaluate ontologies once they have been developed, which accomplishes all these evaluation factors by using a detailed set of 117 criteria.

In this section we compare the most mature proposals identified in the systematic review, using the OntoMetric based framework presented in [30]. This framework adds a comparison and measurements concerning the basic ontological elements (concepts, relations, attributes, etc.) to the OntoMetric method.

A comparison has been made of the most mature proposals whose ontologies are available in a specific ontology language (see Table 4, field "Availability"), and are consequently not yet under construction. In addition, although several ontologies identified in this work were developed with different aims, and different specific domains (always with the security aim in mind), it would be interesting to measure them in order to discover how well these ontologies are defined, how complete they are within the field of security and software engineering, and whether they could be integrated and reused, whilst simultaneously identifying the basis of the key requirements that these ontologies should satisfy in order to obtain an integrated and unified security ontology.

The comparison results and conclusions that were obtained after analyzing these most mature ontologies are presented in the following subsections (3.1, 3.2). These conclusions are extracted by comparing similar ontologies: "OWL-S Security and Privacy" (proposed by Denker et al. [14,34,35]) versus “NRL” (proposed by Kim el al. [48,49]), which are general ontologies describing secure mechanisms, and “QoSOnt” (proposed by Dobson et al. [2,36]) versus “IDS Ontology” (proposed by Undercoffer et al. [25]), which focus on specific domains. Although it is considered to be general, the “SecurityOntology” (proposed by Fenz et al. [24,37–43]) is applied to the environment of risk analysis, and will therefore be dealt with in a specific manner.

3.1. General Comparison

Table 5 shows the general measures with regard to the basic elements of the available ontologies which have been obtained using an OWL ontology editor, SWOOP. These basic elements are: the concepts of the domain that the ontologies represent; their metainformation associated through the attributes and their relationship; their instantiation by using concrete elements (instances); and the association of their elements through the use of inheritance (taxonomic relations, root concepts and subclasses).

At first sight, we can observe that “OWL-S Security and Privacy” has a greater number of concepts and instances than “NRL”. This points to the fact that “NRL” is more general and does not provide details of any concrete area since both describe secure mechanisms; “NRL” is composed of seven sub-ontologies (see Section 2 for details) and that of “OWL-S Security and Privacy” is mainly composed of two. This conclusion is enhanced by the fact that both define an ontology which is focused on authentication methods, but “OWL-S Security and Privacy” defines it in greater depth. However, although “OWL-S Security and Privacy” carries out a greater conceptualization of the domain, it uses fewer attributes to define concepts and it should assign more properties (related to inheritance), as “NRL” does (details in following section).

On the other hand, the “QoSOnt” and “IDS Ontology” proposals present a greater number of concepts since they attempt to model specific issues (dependability and computer attacks). However, “IDS Ontology” does not identify attributes, and attributes are not used to define concepts, which is necessary if we are to understand the concepts of the domain represented in the ontology.

Finally, although “SecurityOntology” is a general ontology, it is applied to the scope of risk analysis, and therefore has many concepts related to this (for example, the kind of threats or security controls specified for standards [45], although the latter has 133 concepts). This explains the tremendous difference between the number of concepts in the other ontologies analyzed. However, we must take into consideration that there are several concepts in “SecurityOntology” which do not model the risk analysis domain, such as the conceptualization of an organization's components and a classification of possible software in a company which is used as an example.

We can infer that a high number of instances is a clue to the application of the ontologies to case studies (real or invented). However, this is not important since we are interested in evaluating the ontologies with regard to the knowledge represented in the conceptual model of the ontology, supposing that several sets of instances may or may not exist.

Table 5
General comparison.

	OWL-S security and privacy	QoSOnt	Security ontology	NRL	IDS ontology
Number of concepts	87	92	453	82	106
Root concepts	45	32	18	20	41
Instances	136	61	601	81	22
Avg depth of inheritance	1,9	2,26	3,28	2,19	1,8
Avg of rel concepts	0,57	0,62	0,99	0,37	0,55
Avg of attributes	0,11	1,18	2,45	0,42	0
Avg of subclasses	0,44	0,65	0,95	0,65	0,61
N of taxonomic relations	42	60	434	62	65
N of no taxonomic relations	24	25	47	25	75

The remainder of the measures shown in Table 5 complement the OntoMetric [70,71] comparison, which is carried out in the following subsection.

3.2. OntoMetric

OntoMetric [70,71] is based on comparing the importance of the objectives and features of ontologies in order to measure whether these ontologies can be reused in new projects. This method is used to compare ontologies, and is composed of factors which are grouped into five dimensions: the content represented in the ontology, the language in which the ontology is implemented, the methodology followed to develop it, the software environments used to build it, and the cost of using the ontology in the system.

In our comparison, only the content dimension (as described below) was studied in depth since we consider that when a user finds an ontology, s/he should first analyze the identified concepts (and how they are represented) and check whether they satisfy the needs of the system to be modelled. We have, however, considered some aspects from the other dimensions:

- The *language* dimension is particularly important for integrating the ontology into the system. We have decided that the ontology should be described by using OWL (Web Ontology Language), which is accepted as a standard by the World Wide Web Consortium (W3C), and which has sufficient properties to permit the integration and combination of the ontologies analyzed.
- In the *software environments* used for building the ontology — such as Ontolingua, WebOnto, Protégé, etc. [70] the visualization, edition, user-friendliness and interaction with other tools of the software environment are important factors. We have detected that the use of Protégé and OWL covers all the necessities for the integration of ontologies, but in this case we have not restricted the software used for the comparison.
- The *methodology* and *cost* dimensions have not been considered for the comparison, the former because in the field of security the normal means by which to describe an ontology is based on security standards or best practices (a classification accepted by the community), and the latter because we have not obtained the data to show the estimation of costs which has led to the development of the compared ontologies in their respective projects.

Having studied the *content* dimension in depth, the following four factors and their related characteristics have been considered for the comparison: concepts, relations, taxonomy and axioms:

- The *concepts* factor focuses on analyzing the degree of coincidence of the domain concepts which are modelled, and how they are specified. We first analyze whether the essential concepts are collected, which implies that an ontology must include the fundamental concepts of what is modelled, also bearing in mind that the people who use this ontology will be able to find these concepts easily and without ambiguity. So, for example, possible synonyms for the concepts have to be modelled and the names for the concepts must be well described in natural language, as is the case of the attributes chosen for the concepts. We therefore consider whether they contain the appropriate attributes; whether these concepts are described in the upper levels of the taxonomy in order to make them more reusable and, whether they are conveniently described in natural language, along with their description such as the attributes, relationships and axioms described; the suitability of the attributes defined to describe the concepts; and finally whether the number of concepts represented (the size of the ontology) is adequate, in relation to the domain that they represent. The size of an appropriate ontology is a subjective factor, and depends on the domain that is intended to be modelled (for example, whether or not all the possible synonyms are considered) and the use that it will be put to. In fact, the most important aspect of this issue is that if the number of concepts is too large it could signify that an ontology is not easy to use, and this could perhaps imply that a process is needed to select the concepts. On the contrary, if the number of concepts is too small, this could imply that the ontology does not collect all the concepts (which is a deficiency).

- In the *relations* factor, as in the case of concepts, the degree of compliment with the domain necessities must be established, along with whether the way in which they are specified is appropriate. We have described whether the essential relations have been collected and are associated with suitable concepts; whether they have been described in natural language and whether their arity is appropriate; whether they are reflected in a suitable number; and finally, it is important to discover whether they collect the formal properties of relationships such as: Symmetry, Asymmetry, Antisimmetry, Reflexivity, Irreflexivity, Transitivity and Intransitivity (see [70] for details).
- With regard to the *taxonomy*, we assess whether the concepts of the ontology are suitably organized (the taxonomy is suitable for the ontology); whether there has been a classification of concepts in various perspectives (i.e., does a concept have a number of "subclass_of" type relationships in the same concept?); whether the set of classes that form the partition is comprehensively defined to the parent class (exhaustive partition), whether the partitions are disjoints (no instances have common or sub classes); and we finally assess the maximum depth in the hierarchy of concepts and the average number of children per concept (subclasses).
- The *Axioms* factor shows us how these axioms can be used to restrict the values of the attributes of the instances and instances of relationships, to maintain the consistency of the ontology and to make deductions. We finally consider whether these axioms are defined independently of the ontology and describe whether the amount of axioms can provide us with an idea of the potential deduction specified in the ontology and the ability to maintain consistency.

Table 6
OntoMetric comparison: concepts factor.

Concepts factor	OWL-S security and privacy	NRL	Security ontology	QoSOnt	IDS ontology
Essential security concepts	3	4	4	4	3
Concepts concerning Vulnerabilities, Threats, Attacks	–	–	4	4	3
Concepts concerning Controls, Countermeasures	–	–	4	–	–
Concepts concerning Security Protocols, Mechanisms, Policies	3	4	-	–	–
Essential concepts in superior levels	4	5	3	5	4
Concepts properly described in NL	3	2	4	3	1
Formal specification coincides with NL	4	3	4	5	1
Attributes describe concepts	2	4	4	4	1
Number of concepts	4	3	3	5	4

* degree of acomplishment 1: very low, 2: low, 3: medium, 4: high and 5: very high.

Table 7
OntoMetric comparison: relations factor.

Relations factor	OWL-S security and privacy	NRL	Security ontology	QoSOnt	IDS ontology
Essential relations	4	4	4	4	4
Relations relate appropriate concepts	5	5	5	5	5
Relations property described in NL	2	1	4	3	1
Arity specified	4	4	4	4	4
Formal properties of relations	1	1	3	2	2
Number of relations	4	4	4	4	3

* degree of acomplishment 1: very low, 2: low, 3: medium, 4: high and 5: very high.

For the purposes of this comparison, each factor has a group of measurable characteristics which are scored from 1 to 5 according to their low or high degree of accomplishment, specifically very low, low, medium, high or very high depending on the suitability of the ontology for the feature we are evaluating, as is suggested in [70]. The values considered for each characteristic are shown in Tables 6–9.

3.2.1. Concepts factor

Table 6 shows the values for the concepts factor. This table shows how the ontologies accomplish this factor through a group of six features measured in each row of the table. The first feature represents whether the ontology contains the essential concepts of the domain. The level of accomplishment of this feature for the ontologies "NRL", "SecurityOntology" and "QoSOnt" is high, that is, these ontologies contain the essential concepts of the domain they are attempting to model. In this case, in order to analyze this factor properly, it is first necessary to clearly identify the domain that the ontology is intended to model. We therefore compare the ontologies by grouping them according to the domains identified in Table 6: "Vulnerabilities, Threats and Attacks", "Controls and Countermeasures" and "Security Protocols, Mechanisms and Policies". As mentioned later (Section 4.2.1) the essential concepts in the field of security are extracted from the main security standards and best practices.

"NRL" and "OWL-S Security and Privacy" ontologies focus principally on the concepts of "Security Protocols, Mechanisms and Policies". We have detected that the essential concepts are better detected in "NRL", since it includes more essential concepts than "OWL-S Security and Privacy", although it is not possible to identify all of them because the mechanism field and security protocols alter drastically in a short period of time. For example, for the concept of 'security credential', "NRL" considers essential concepts as being "physical tokens", "electronic tokens" or "biometric token", whereas "OWL-S Security and Privacy" solely considers "electronic token".

"IDS Ontology" and "QoSOnt" must be considered in the domain of "Vulnerabilities, Threats and Attacks". In this respect, "QoSOnt" collects the essential concepts in a better manner, since, for example, unlike "IDS ontology", it considers concepts such as safety and

Table 8
OntoMetric comparison: taxonomy factor.

Taxonomy factor	OWL-S security and privacy	NRL	Security ontology	QoSOnt	IDS ontology
Several perspectives	2	2	2	4	2
Appropriate exhaustive partitions	2	4	3	3	4
Appropriate disjoint partitions	2	4	1	4	1
Maximum depth	3	4	2	4	3
Average of subclasses	2	3	4	3	3

* degree of acomplishment 1: very low, 2: low, 3: medium, 4: high and 5: very high.

Table 9
OntoMetric comparison: axioms factor.

Axioms factor	OWL-S security and privacy	NRL	Security ontology	QoSOnt	IDS ontology
Solves queries	2	2	5	3	3
Infers knowledge	2	3	4	3	4
Verifies consistency	3	3	5	3	3
Not linked to concepts	2	3	4	1	1
Number of axioms	2	2	4	3	3

* degree of acomplishment 1: very low, 2: low, 3: medium, 4: high and 5: very high.

dependability. What is more, "IDS Ontology" also focuses on a continuously changing domain: 'attack intrusion on the network'.

Finally, "SecurityOntology" is that which covers most security concepts, considering not only "Vulnerabilities, Threats and Attacks", but also "Controls and Countermeasures", and therefore collecting more essential concepts than the previous proposals. But it still does not take into account all the possible kinds of essential Vulnerability and Control concepts (again because innovations appear quickly in security), and what is more important, it should consider terms which are synonyms in other risk analysis methods, not only those on which it is based [44], such as for example CRAMM [61] or MAGERIT [74].

The second feature represents whether the superior levels of the taxonomy contain the essential concepts of the domain that the ontology models. As we can see in Table 6, both "NRL" and "QoSOnt" obtain the best score. Although "SecurityOntology" has more concepts than the other ontologies, the essential concepts of the security domain are not included in the superior levels of the ontology. The third row of the table states that even when "NRL" models almost all the essential concepts of the domain, these concepts are not properly described in natural language. On the other hand, "SecurityOntology" describes the concepts in natural language in a precise manner, thus facilitating the reuse of the ontology. The next feature represents whether the formal specification of the concepts, that is, the properties and axioms that characterize each concept, corresponds with the natural language description of this concept. Here, "QoSOnto" obtains the highest score because the concepts of the ontology are correctly specified with attributes and axioms. On the other hand, "IDS ontology" obtains the worst score because it does not contain any attributes, as is shown in the sixth row of Table 5. The fifth row indicates how the attributes describe the concepts and whether they are well defined in the ontology. For this feature, "IDS ontology" again obtains the lowest score because it does not contain any attributes. The concepts of "NRL", "SecurityOntology" and "QoSOnt" are well defined by their attributes. Finally, the last feature represents whether the size of the ontology is appropriate in terms of concepts. Here we can see that, although "SecurityOntology" contains more concepts than the other ontologies, they are not sufficient to represent the risk management domain, and this ontology obtains worse results than the "QoSOnt" ontology.

By examining all the characteristics of this factor we can note that, despite having stated in the section above that "OWL-S Security and Privacy" performs a greater conceptualization of the domain, some concepts in this ontology ('security mechanism') are useful concepts in another domain, but are not particularly relevant for describing security-related information (such as 'syntax' and 'data transfer protocols'). Therefore, although "NRL" describes more security concerns (and identifies fewer concepts than "OWL-S Security and Privacy"), we consider that both identify the same essential concepts (as Table 6 shows). However, "NRL" makes their reuse difficult because the concepts are not properly described in natural language.

Furthermore, "OWL-S Security and Privacy" should assign more properties to each concept if it is to correctly define the attributes that the concept instances have. Moreover, we have identified that the properties that "OWL-S Security and Privacy" defines are determined for the top class. However, these properties do not apply to most of the subclasses (inadequate use of inheritance). For example, no instance under the 'Syntax' subclass has a property or a need for the 'enc' (relative to Encryption) or 'reqCredential' (Required Credential) properties, yet they are all inherited because these properties are defined in the top class. "IDS Ontology" describes concepts in a schematic manner and does not make use of attributes to describe them, thus making the concepts difficult to understand. Nevertheless, "QoSOnt" widely describes concepts in natural language and includes attributes for defining them. "QoSOnt" is therefore more reusable. Finally, the "SecurityOntology" collects the concepts related to risk analysis in an efficient manner, with the exception of some concepts related to the system's assets since only the infrastructure assets have been considered (i.e. the service assets are not considered). For example, we can see other kinds of assets in risk analysis methods [74] or [61].

3.2.2. Relations factor

Table 7 specifies the values for the relations factor. The first row of this table indicates whether the ontology contains the essential non taxonomic relations between the domain concepts. The level of accomplishment of this feature for all the ontologies is high, that is, these ontologies cover the most important relations between the concepts of the domains they are modelling. The second feature represents whether the relations are associated with the right concepts in the ontology. For this feature, all the ontologies obtain the highest score because the concepts that participate in the relations are appropriate. The next row of the table indicates that although "OWL-S Security and Privacy", "NRL" and "IDS ontology" include almost all the essential relations, these concepts are not properly described in natural language. On the other hand, "SecurityOntology" describes the concepts in natural language in a precise manner, thus facilitating the reuse of the ontology. The fourth feature represents whether the number of arguments in the relation, that is, the number of concepts associated with each relation, is appropriate. All the ontologies obtain a high score for this measure, indicating that the arity of the relations is well defined. The fifth feature states whether the relations are defined through formal properties such as reflexivity, transitivity, asymmetry and symmetry, which are extremely useful in verifying the consistency of the ontologies. The "OWL-S Security and Privacy" and "NRL" ontologies obtain the lowest score for this measure, again because they do not specify any formal property in the relations. On the other hand, the relations in "SecurityOntology" are defined by the functional, symmetry and transitive properties. Finally, the last feature represents whether the size of the ontology is appropriate in terms of relations. The correct number of relations depends on the number of concepts in the ontology. A small number of relations could signify that the essential relations have not been modelled. On the other hand, a large number of relations could imply a less manageable ontology, as is the case of "IDS Ontology" which has 75 non-taxonomic relations for 106 concepts.

In general, although the relationships have been properly defined in the ontologies, they are not properly specified in natural language and not all of the formal properties of the relations are identified. In fact, some authors such as those of "OWL-S Security and Privacy" and "NRL" do not take them into account. The relations should be specified in a formal manner to obtain both reusability and to detect incongruence.

3.2.3. Taxonomy factor

The values for the taxonomy factor are presented in Table 8. The "Several perspectives" feature examines the correctness of the subclass of relations and whether the ontology contains more than one subclass of relations for the same concept and this classification is necessary in the domain. In this case, "QoSOnt" defines the taxonomy by considering this aspect. The second measure identifies whether the set of subclasses is comprehensively defined to the parent class, that is, it evaluates the completeness of the partition. The level of accomplishment of this feature is high for all the ontologies, given that they include the appropriate concepts in the partitions. On the other hand, in the “OWL-S Security and Privacy” ontolog,y some of the concepts in the hierarchy are missing. The fourth feature evaluates whether the disjointness of the classes of the ontology has been defined. This guarantees that an individual that is a member of one class cannot simultaneously be an instance of another specified class. For this feature, “SecurityOntology” and “IDS Ontology” do not specify those disjoint classes in the ontology which are used in automatic classification processes, and thus obtain the lowest score. Finally, the last feature calculates the average number of subclasses in the ontology, which helps to determine the detail of the taxonomy in the ontology. This measure has been calculated in Table 5, As we can see, “SecurityOntology” obtains the highest score with an average of 0.95 subclasses of relations.

In general, the concepts in the ontologies analyzed are not classified under different perspectives and the ‘no_subclass_of’ relation has not been used. We have identified that obtaining the exhaustive partition is a difficult task, because new security issues always appear, and can be included. Therefore, we have detected that all the ontologies improve the use of appropriate exhaustive partitions if they review all the possible decomposition classes in the domain. For example in “SecurityOntology” we have detected that the hierarchy or kind of asset identified is not complete, because it focuses solely on Infrastructure assets. In this respect, exhaustive partitions are easier to define in a concrete domain (thus “IDS Ontology” and “QoSOnt” proposals have higher values). On the other hand, we have identified that some concepts in “SecurityOntology”, such as the kind of threat, should be hierarchically categorized in order to facilitate their reuse, since some threats are closely related. Another weakness of “SecurityOntology” is that the disjoint partitions, which are important in the automatic classification of ontologies, are hardly ever used. A further conclusion is that “NRL” pays more attention to this factor than does “OWL-S Security and Privacy”, since the former is more exhaustive than the latter, and an attempt should be made to obtain the general concepts in the first layer of the ontologies, in order to make them more reusable.

3.2.4. Axioms factor

Finally, in Table 9, we show the axiom factor and its features. The first feature identifies whether any of the axioms in the ontology are defined to query non explicit knowledge in the knowledge base (instances) of the ontology. In OWL, these axioms are usually represented by the allValuesFrom, someValuesFrom and hasValue restrictions. “SecurityOntology” comprises these restrictions in the ontology, thus assisting in the inference of new knowledge. The next feature studies whether the axioms help to infer either some of the values of the attributes of the new instances inserted into the ontology or some instances of the relations of the ontology. The cardinality and range restrictions are used to model this kind of axioms in OWL. The level of accomplishment of this feature for “SecurityOntology” and “IDS ontology” is high, since they implement these restrictions on the ontology. On the other hand, “OWL-S Security and Privacy” should define the range of the attributes and relations defined in the ontology. The third feature states the usefulness of the axioms for checking the consistency of the ontology. More specifically, this kind of axioms allows the permitted values of the attributes and the relations between concepts to be verified and restricted. For this feature, “SecurityOntology” implements this type of axioms in an effective way. The “Not linked to concepts” measure identifies the use of external axioms that do not depend on the terms of the ontology. In this case, “QoSOnt” and “IDS Ontology” obtain the lowest score because they should specify this kind of axioms in order to facilitate their understanding and modification. Finally, the number of axioms in the ontology is studied in the last feature. This measure indicates the potential for making inferences in the ontology. Here, “SecurityOntology” includes several axioms in the ontology. On the other hand, “OWL-S Security and Privacy” and “NRL” ontologies should add new axioms in order to exploit the reasoning capability provided by the ontologies.

By examining all the features of this factor we can conclude that “NRL” and “OWL-S Security and Privacy” define few axioms and cannot, therefore, infer knowledge, only some restrictions with regard to the value of the attributes of the concepts, while “QoSOnt” and “IDS Ontology” use more constraints and can infer knowledge and verify consistency, but these are related to the concept of the ontology (they are not independent). The “SecurityOntology” is worthy of note because it takes the axioms into account in order to verify consistency and to solve questions (independent of the ontology). An example of application is shown in [38] in which the JESS reasoner is used [75]. In this case, axioms are defined independently of the concepts of the ontology, thus providing a greater ease of comprehension and modification, since their definition does not depend on the changes made to other concepts in the ontology.

3.3. Conclusions of the comparison

We have first used the framework presented in [30] to focus on basic elements (Section 3.1), and we have then used a formal framework [70] to compare the most mature proposals of the security ontologies identified in Section 2.3, after carrying out our systematic review. The aim of this was to obtain a vision of the current situation which will allow us to discover how well these ontologies are defined, and how they could be integrated and reused, whilst simultaneously allowing us to identify (in the following sections) the key requirements for an integrated security ontology.

As a conclusion of this comparison, we have discovered that not only do the ontologies include few attributes with which to define concepts but also that the natural language expressions used to describe them are not appropriate, because they are too schematic and, therefore, difficult to understand. The ontologies are not exhaustive because they do not define all the possibilities of the studied domain. An example of this can be seen in “SecurtityOntology” which does not define all the types of assets (see previous section). These ontologies use few axioms and formal properties in their relationships to infer knowledge such as reflexivity, transitivity, symmetry and asymmetry. What is more, certain ontologies (such as that of “OWL-S Security and Privacy”) describe as instances elements which should be considered as domain concepts, or the taxonomic classification is not sufficiently generalized (as in the “SecurityOntology” proposal).

However, we should highlight “SecurityOntology”, which can infer knowledge by using axioms, although it does not have an appropriate taxonomy classification. Moreover, “NRL” has a good taxonomy classification of several security concepts, and collects the essential concepts (by following security standards for Web Services). This is also true of “QoSOnt”, although it is focused on the dependability domain. Moreover, all the ontologies have collected the essential relations between the concepts of their domains.

4. Towards an integrated general security ontology

Defining an integrated security ontology is a highly complex challenge which the scientific community has not yet completely fulfilled. In this section, we offer a discussion on this, which is organized into three subsections. Section 4.1 describes the problems of integrating ontologies. In Section 4.2 we describe the key requirements that we believe a unified and integrated ontology should consider, extracted from the formal comparison described in Section 3. The level of achievement of the previously studied proposals with respect to these key requirements is also discussed,

Table 10
The key requirements, and how they are satisfied by the proposals.

	Static knowledge					Dynamic knowledge		Reusability		
Key Requirements Proposal	Essential concepts	Follows standards	Updated	Attributes are described	Relations are described	Detects inconsistency	Infers knowledge	Taxonomy (Exhaustive partition and generalized)	Commented in NL	Standard language
OWL-S security and privacy	p	n	N	p	p	p	n	p	y	y
NRL	y	y	Y	y	p	y	p	y	p	y
Security ontology	y	y	Y	y	p	y	y	p	y	y
QoSOnt	y	n	N	y	p	y	n	y	y	y
IDS ontology	p	y	N	n	p	y	p	y	n	y

* key requiremetns are: y = yes; n = no; or p = partially satisfied.

again through a comparison (summarized in Table 10). Finally, in Section 4.3 we present the basis for obtaining a first proposal of a general security ontology, which should satisfy the key requirements identified, considering the problems defined in Section 4.1.

4.1. Ontology integration

There is an increasing interest both in ontologies and in the idea of reusing existing (domain) ontologies [9]. Moreover, a high level of consensus has been reached that both costly human resources and intense organizational work are normally required to construct ontologies. Thus, on the one hand, experts and knowledge engineers must work together in order to create an ontology. On the other hand, it is necessary to co-ordinate experts' work, and to manage the information flow between the human agents involved in the ontology's construction. In this respect, a complex elicitation process between experts and knowledge engineers is usually required. This situation creates various problems, such as the need to have meetings between experts and knowledge engineers and to establish a consensus on the task schedule that must be adhered to in order to construct an ontology [76]. Fortunately, experts' availability can to some extent be overcome by exploiting the possibilities of a communication network to which a set of user nodes is connected. (Expert) user nodes can therefore be distributed in both space and time while all of them are solving a certain task.

Given the difficulties involved in resolving this problem, we believe that the construction of corporate ontologies can be enhanced by reusing already existing ontologies. Constructing reusable domain ontologies is not straightforward: there is a trade-off between specific use and reuse. That is, the more committed an ontology is to a specific domain and task, the less its terminological elements can be generalized and reused in other domains and tasks. Many authors have defended the concept of reusable ontologies as a worthwhile and attainable goal in knowledge representation. Several of them have attempted to show that domain ontologies can be constructed for specific systems in which the ontology is reused in alternative tasks or applications.

Hence, ontologies would be reused by applying ontology integration processes. In [77] integration processes are considered beneficial for promoting knowledge sharing and reuse. According to Reimer [78], Knowledge Integration can be seen from two points of view: integration of different knowledge bases and integration of different representations of the same knowledge at different formalization levels.

There is no consensus in literature on how to define ontology integration processes, although authors consider ontology integration to be a complete process rather than a single ontology. However, these authors provide different definitions of this:

(1) In [79], the integration process is divided into the following steps:aggregation
(2) combination
(3) assembling different source ontologies together in order to form the resulting ontology, possibly after reused ontologies have undergone some changes.

On the other hand, for [80], ontology integration consists of three iterative steps:

(1) finding overlapping areas within the ontologies;
(2) relating concepts;
(3) checking for all consistency, coherency and non-redundancy of the result.

Nevertheless, the Semantic Web community would appear to agree on the objectives of such a process, that is, reusing ontologies to build more complete domain models to, for example, enhance the semantics of Web content. Work on collaborative construction of reusable knowledge components (i.e., ontologies) can be split into two categories: (1) frameworks, algorithms and tools for integration, aligning and merging; and (2) collaborative development of a global ontology. However, working cooperatively can give rise to several problems [4]. Some of these are:

- Redundant information. Two different experts might attempt to describe the same part of the domain knowledge. Given this eventuality, it would be desirable for the system to be capable of managing this possible situation so that redundancies could be avoided.
- Use of synonymous terms for a concept. Apart from dealing with redundant information, different experts may employ different terminologies for the same concept. In other words, there might be correspondence between different terms employed for a given concept. During the ontology construction process, information concerning the use of synonymous terms for a concept must be stored and managed, since a particular terminology should not be imposed on any expert during the Knowledge Acquisition process. However, an ontology would strive towards 'consensual knowledge', that is, a fixed terminology. Synonyms are possible but, ideally, everybody should agree on the terminology.

In the case of a framework for the merging of ontologies, a third collaborative problem must be highlighted; that of inconsistencies among knowledge descriptions.

4.2. Key Requirements of an Integrated Security Ontology

As a consequence of the formal comparison made in Section 3, we believe that the key requirements that a unified and integrated ontology should consider can be grouped into three aspects: i) STATIC KNOWLEDGE, which will allow the concepts collected in the ontology to be properly identified ii) DYNAMIC KNOWLEDGE, in order to ensure that the knowledge collected in the ontology can be used to infer other knowledge and, finally iii) REUSABILITY, enhancing the fact that the ontology is developed by taking into consideration aspects that permit its reuse and shareability. These key requirements are presented in Table 10, and are divided into several sub-aspects, showing how the proposals identified in the previous section satisfy each one: completely (y — yes), not at all (n — no) or partially (p). We shall now explain all these sub-aspects in detail, comparing the proposals through these key requirements.

4.2.1. Static knowledge

STATIC KNOWLEDGE refers to the fact that an integrated ontology should satisfactorily describe in natural language the essential concepts, along with their associated properties (relations and attributes) for the domain it models. An ontology must include the fundamental concepts of what is modelled, also bearing in mind that the people who use this ontology will be able to find these concepts easily and without ambiguity. So for example, possible synonyms for the concepts have to be modelled. Specifically, in the field of security, the essential concept of the ontology should be principally collected from security standards or best practices (for example, a taxonomy accepted by the security community). Security standards (such as ISO/IEC 27001 [45], ISO/IEC 15408-1999 [83], etc.) and best practices (such as MAGERIT [74], CRAMM [61], OCTAVE [84], COBIT [62]), must therefore be considered, along with those terms which are synonyms and the correlations between them. In relation to this is the problem of the size of the ontology, which is a subjective feature that depends on what is intended to be modelled, and its use. We must bear in mind that if the number of concepts grows considerably then this will imply the use of a precise process to select the concepts. In the field of security this growth is normal because concepts must be updated, since the security environment is unsettled and new standards (or new versions) and terms appear quickly, signifying that concepts are soon out of date.

Furthermore, the relations should be specified with their formal properties (Symmetry, Asymmetry, Antisymmetry, Reflexivity, Irreflexivity, Transitivity and Intransitivity) [70] which verify consistency. Finally, instances of concepts should be used only as examples, since the suitable valuation is made with regard to the represented knowledge in the conceptual model of the ontology, supposing that several sets of instances may or may not exist.

By examining the first aspect in the studied ontologies we can conclude that only “NRL” and “SecurityOntology” satisfy all the key requirements for the STATIC KNOWLEDGE aspect. Although the “QoSOnt” and “OWL-S Security and Privacy” ontologies consider the essential concepts (see Section 3.2.1, ‘Concepts factor’, for further details) and both the attributes and relations are well described, they do not follow the standards, and some of the concepts that they contain are out of date. The “IDS Ontology”, partially includes (see Section 3.2.1, ‘Concept factor’, for details) the essential concepts and the relations are well described, but it should describe the attributes properly. Although this ontology follows the standards, it should be updated. Finally, the “SecurityOntology” collects the concepts related to risk analysis in an efficient manner, with the exception of some concepts related to the system's assets, since only the infrastructure assets have been considered (i.e. the service assets are not considered). We must therefore consider other kinds of assets in risk analysis methods [74] or [61], and detect other concepts or assets that are synonyms in this other risk analysis method.

4.2.2. Dynamic knowledge

The use of axioms (equivalent to DYNAMIC KNOWLEDGE in Table 10) should also be defined in order to restrict the values of the attributes of the instances and instances of relationships, to maintain the consistency of the ontology and to make deductions or infer knowledge. Another important factor is that axioms should be defined independently of the concepts of the ontology in order to provide us with a greater ease of comprehension and modification, since their definition does not depend upon the changes made to other concepts in the ontology. This issue is extremely important in security. For example, in risk analysis we can identify relationships between the threats to a system and the asset affected, independently of the system modelled. In conclusion, this is outside the sphere of Lightweight ontologies which include concepts, concept taxonomies, relationships between concepts, and properties that describe concepts, but not within the sphere of Heavyweight ontologies which add axioms and constraints to lightweight ontologies [81].

Regarding the DYNAMIC KNOWLEDGE factor in the selected ontologies, although “NRL” and “QoSOnt” permit consistency to be checked, some axioms for inferring new knowledge should be included in the ontology. Both “OWL-S Security and Privacy” and “IDS Ontology” lack axioms for detecting inconsistencies and inferring knowledge. Finally, “SecurityOntology” implements the axioms successfully.

4.2.3. Reusability

The taxonomy of the concepts should be adequate, and should be conscious of the difficulty of representing an exhaustive partition in a domain (particularly in security), and should therefore be prepared for reuse (REUSABILITY in Table 10). The generalization of concepts should thus be taken into account, since if the essential concepts are in the top level of the hierarchy they facilitate the reuse of the concepts. Furthermore, disjoint partitions should be considered, since they are important in the process of automatic classification of the ontologies, and different perspectives of the concept being classified should sometimes be given (for example whether the people involved have several roles). This is an important factor in the field of security, since different roles will normally be involved (security designers, developers, users, administrators or managers), and it is not usual for all of these people to be experts in the domain. This reinforces the fact that the concepts should be described properly in natural language. Another important issue is the language used to implement the ontology. Its expressive and reasoning capacity should be analyzed, since the language is particularly important for integrating the ontology into a system or with other ontologies. In fact, ontologies help us to specify a conceptualization (describe the semantics of information) in a particular context, and therefore allow information (reasoning) to be extracted from the concepts, and their relationships, which are included in the ontology, and new knowledge to be created. As we have stated in the section above, the language we have selected is OWL, which is accepted as a standard by the World Wide Web Consortium (W3C), and which has sufficient properties to permit the integration and combination of the ontologies analyzed. This language should be combined with the software environments used for building it, in which its visualization, edition, user friendliness and interaction with other tools are important factors. As regards the methodology [82], we have not identified one which is ideal but we consider that the most important aspect in security is that the basis through which to develop the ontology should be security standards (e.g. ISO/IEC 27001 [45], ISO/IEC 15408-1999 [83], etc.) and best practices (such as MAGERIT [74], CRAMM [61], OCTAVE [84], COBIT [62]).

With regard to the REUSABILITY factor in the proposals studied, only “QoSOnt” completely accomplishes all the key requirements. Both “NRL” and “IDS Ontology” should improve their natural language descriptions of the ontological elements (concepts, relations and attributes) to facilitate their understanding. Finally, the taxonomy of “OWL-S Security and Privacy” and that of “SecurityOntology” should be revised.

Additionally, the cost of developing the integrated ontology should be as low as possible, and always within the bounds of the budget of the project. These costs include the purchase and exploitation of ontology licenses training, installations and maintenance. The cost of

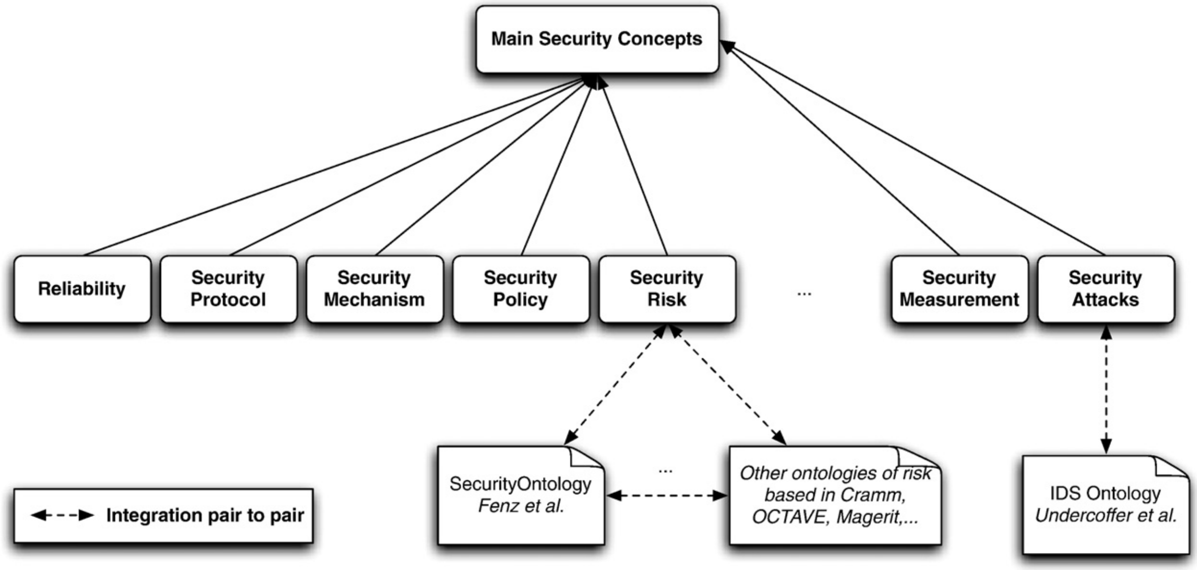

Fig. 1. Overlapping between security domains for integrated security ontology.

the methodology and software environments chosen to develop the ontology should therefore be considered.

As regards the proposals analyzed, although they make important contributions to the security community, they offer only partial solutions to the integration of knowledge into a general security ontology, and we have identified some aspects that could be improved (see Table 10). The security community consequently still needs an integrated security ontology to solve these aspects in order to improve and provide reusability, communication and knowledge sharing. The first steps towards the definition of such an ontology are presented in the following section.

4.3. An integrated general security ontology

As was stated in Section 4.1, there is no consensus on how to define the ontology integration process, and it is a difficult and time consuming task. The integrated ontology to be obtained should achieve all the key requirements described above (Section 4.1). We have identified that the best manner in which to obtain this integrated ontology is that of studying current security ontologies and attempting to combine them by following these steps.

The integration of ontologies must be performed in pairs. First, the ontologies that are going to be integrated have to be sorted using one criterion, and they will then be inserted into the transformed ontology in the order established by that sorting process. Different criteria can be used for this. For example in [4] the ontologies are sorted according to the number of concepts they have. This criterion is used to minimize the number of modifications to be made to the transformed ontology, since it is more probable that the knowledge of the remaining ontologies has already been included in the transformed ontology. We believe that the size of the ontology should be considered in the integration process, and this may be a good starting point for the automatic integration of ontologies, but other factors would have to be taken into account in a manual integration process such as those identified in the section above (STATIC KNOWLEDGE, DYNAMIC KNOWLEDGE and REUSABILITY). Specifically, when we attempt to integrate and unify ontologies we must take special care of the growth of the size of these ontologies, because this could imply having to use a precise process to select the concepts once the concepts of these ontologies have been integrated.

In this respect, we would use the main security concepts of the "NRL" ontology (such as Reliability, Security protocol, Security Mechanism, Security Policy or Security Risk) as a starting point and we would extend this ontology with new concepts such as security assurance, security credentials, security algorithms, risk analysis, etc. in order to obtain this integrated security ontology (Fig. 1). These new security concepts will therefore use other ontologies described in other proposals to specify the specific concepts. More specifically, the "SecurityOntology" could be considered as the second ontology to be integrated, since it considers the elements identified in the section above as part of an integrated ontology, with the exception of the aspect of taxonomy (REUSABILITY), i.e., several perspectives are not considered, and neither is the disjoint relation.

We could then consider the "IDS Ontology", the "QoSOnt" and the "SecurityOntology", and fix all the deficiencies identified in the comparison (see Section above). This general security ontology could also be extended through the addition of new ontologies such as, for example, a security measurement ontology which is highly important in security issues, and is not considered by the ontologies analyzed. All this work must be carried out by taking into consideration the relationships between the concepts of the different ontologies identified.

Once the ontologies had been sorted, the first would be selected as the skeleton and would have to be integrated with the second. The result of this integration would then have to be integrated then with the third, and so on. The ontology integration process that we suggest is that proposed by [80] and, as we have stated previously, consists of three iterative steps: (1) finding overlapping areas within the ontologies (see Fig. 1); (2) relating concepts; and (3) checking for all consistency, coherency and non-redundancy of the result.

Obviously, when integrating ontologies we observe that the concepts defined for each of the ontologies are different, so it is necessary to carry out an initial filtering of conflicts and inconsistencies. This is related to the problem of modelling essential concepts in ontologies, considering for example all the possible synonyms. More specifically, two main kinds of conflicts can occur [85]: conflicts at the instance-level and conflicts at the concept-level. The latter are the most difficult to solve, and more attention must be paid to them. At the concept-level, Nguyen [86] assumes that two ontologies contain the same concept but that its structure is different in both ontologies. Some of the conflicts that can occur at this level are the following:

- The same concept is defined with different terms. For example, "NRL" specifies the terms CryptograhicKey and BiometricToken, while "OWL-S Security and Privacy" uses Key and Biometric to refer to the same concepts.

- The same term is used to represent different concepts. For example, "OWL-S Security and Privacy" describes the term SecurityProtocol as a subclass of SecurityMechanism, but "NRL" describes these two concepts as being in the same level as the Security Concepts subclasses.
- Definition of the same concept using different attributes. Some inconsistencies have been identified in the attributes associated with the ontology and in the use of some of the expressions in natural language which are employed to describe concepts.

Despite the fact that the all authors of this paper are domain experts, combining ontologies will be a difficult task unless one of them is the creator of the ontologies and knows exactly what each term means.

Once the ontologies have been integrated, the consistency of the resulting ontology will have to be checked. As stated previously, the language we have selected is OWL, and more concretely the OWL-DL version based on Description Logics. Its formal model allows a set of Description Logic inference services to be performed automatically, and this can be supported by DL reasoners including HermiT, Pellet2, Fact++ or Racer [87], such as:

- Consistency checking, which ensures that an ontology does not contain any contradictory facts.
- Concept satisfiability, which checks whether it is possible for a class to have any instances. If a class is unsatisfiable, then defining an instance of the class will cause the whole ontology to be inconsistent.
- Classification, which computes the subclass relations between every named class to create the complete class hierarchy. The class hierarchy can be used to answer queries such as obtaining all or only the direct sub-classes of a class.
- Realization, which finds the most specific classes to which an individual belongs; or, in other words, computes the direct types for each of the individuals.

For example, in the case of a correct Classification, we have identified that some concepts in "SecurityOntology", such as the kind of threat, should be categorized hierarchically in order to facilitate their reuse. What is more, the concept of asset can be extended with more concepts related to risk analysis since only infrastructure assets have been considered (for example, with data or service asset). It does, however, take axioms into account to solve queries related to the domain or to restrict the values of their properties or attributes.

Finally, we must consider that one of the STATIC KNOWLEDGE key requirements is "follow security standards and best practices". We are therefore working to develop a risk analysis ontology, by extending "SecurityOntology" to represent security issues following that criteria. We have consequently used our previous work with security requirements [88,89] as a starting point, and we now focus on privacy (an issue not considered by "SecurityOntology"). The work is based on MAGERIT [74]. MAGERIT is the information systems risk analysis and management method used by the Spanish public administration. It conforms to the ISO/IEC 15408-1999 (Evaluation Criteria for Information Technology Security Standard, also known as the Common Criteria Framework — CCF [90]). MAGERIT is based on both international and national legal regulations, which are relevant in the analysis and management of risk: administrative procedure, protection data, electronic signature, classified information and network and information security (see Appendix 3 MAGERIT [74] for details of these regulations). Thanks to this work we have identified certain relationships between threats and assets, which are not identified in "SecurityOntology". What is more, at the moment of integration we have extended the threats hierarchy, which had been detected as a necessity for "SecurityOntology".

5. Conclusions

After the planning and execution of the systematic review, and once the most mature proposals had been studied and compared, we observed that the majority of the identified works focus on specific domains, thus signifying that the scientific community has not accomplished an integrated security ontology, although this has been identified as a branch of research. In the field of security it is difficult to formalize all existing concepts, which should be adapted to security standards. The definition of an integrated security ontology has not, therefore, been considered as an isolated task, and the community should make greater efforts to combine and improve the ontologies developed. However, we have used a formal comparison to identify that the analyzed proposals, despite making important contributions to the security community, offer only partial solutions to the integration of knowledge into an integrated security ontology. We have therefore presented the key requirements that ontologies should take into consideration if they are to obtain an integrated and unified security ontology. This integrated security ontology should at least identify the essential and updated concepts (STATIC KNOWLEDGE), should allow us to infer knowledge (DYNAMIC KNOWLEDGE), and should be reusable and shareable (REUSABILITY).

We have, moreover, identified that the best way in which to obtain this integrated ontology is to study the current state of the art (as has been done in this paper), comparing the most mature proposals through a formal comparison (through the factors of contents, taxonomy, relation and axioms) in order to obtain a vision of the current situation, and to detect aspects which should be improved to integrate and combine them, thus reducing the cost of developing new ones from scratch without taking those which already exist into consideration. We have also identified that the combination of the Protege software and the OWL language, which is accepted as a standard by the World Wide Web Consortium (W3C), is the most appropriate tool through which to do this. What is more, we have proposed a first schema of this integrated security ontology, taking the ontologies analyzed as a basis, indicating the aspects to be improved and the new issues to be considered in order to combine them, and also identifying new sources of possible concepts.

Finally, as further work, we plan to extend the risk analysis ontology with works in progress such as those described in Section 2 (for example [13]), or other methods or practices accepted by the risk analysis community such as the CCTA Risk Analysis and Management Method (CRAMM) [61], OCTAVE [84] or MAGERIT [74] — Fig. 1). In addition, we wish to apply this integrated security ontology to a framework in order to identify security requirements that we are currently developing [91], with the possibility of adapting it to other security requirements frameworks.

Acknowledgments

This research is part of the following projects: BUSINESS (PET2008-0136), PEGASO-MAGO (TIN2009-13718-C02-01) and PEGASO-PANGEA (TIN2009-13718-C02-02), projects financed by the Spanish Ministry of Science and Innovation (Spain) and the FEDER, and SISTEMAS (PII2I09-0150-3135) and MELISA (PAC08-0142-335), financed by the FEDER and the Regional Science and Technology Ministry of Castilla-La Mancha (Spain).

References

[1] T. Gruber, Towards principles for the design of ontologies used for knowledge sharing, International Journal of Human Computer Studies 43 (5/6) (1995) 907–928.

[2] G. Dobson, P. Sawyer, Revisiting ontology-based requirements engineering in the age of the semantic web, International Seminar on "Dependable Requirements Engineering of Computerised Systems at NPPs", Institute for Energy Technology (IFE), Halden, 2006.

[3] M. Gruninger, J. Lee, Ontology applications and design, Communications of the ACM 45 (2) (2002) 39–41.

[4] J.T. Fernández-Breis, R. Martínez-Béjar, A cooperative framework for integrating ontologies, International Journal of Human Computer Studies 56 (2002) 665–720.

[5] G. Colombo, A. Mosca, F. Sartori, Towards the design of intelligent CAD systems: an ontological approach, Advanced Engineering Informatics 21 (2) (2007) 153–168.

[6] M.-Á. Sicilia, Ontology of systems and software engineering, Advanced Engineering Informatics 21 (2) (2007) 117–118.

[7] M. Donner, Toward a security ontology, IEEE Security and Privacy 1 (3) (2003) 6–7.

[8] S. Durbeck, et al., A Semantic Security Architecture for Web Services — the Access-eGov Solution, International conference on availability, reliability and security (ARES), IEEE Computer Society, 2010, pp. 222–227.

[9] L. Crow, N. Shadbolt, Extracting focused knowledge from the semantic web, International Journal of Human Computer Studies 54 (1) (2001) 155–184.

[10] A. Hameed, D.H. Sleeman, A. Preece, A detecting mismatches in experts' ontologies acquired through knowledge elicitation, Research and Development in Intelligent Systems XVIII, Springer, 2001, pp. 9–22.

[11] R. Sandhu, K. Ranganathan, X. Zhang, Secure information sharing enabled by trusted computing and PEI models, ASIACCS, 2006, pp. 2–12.

[12] G. Dhillon, Information security management: global challenges in the new millennium, Idea Group Publishing, 2001.

[13] B. Tsoumas, D. Gritzalis, Towards an ontology-based security management, Proceedings of the 20th International Conference on Advanced Information Networking and Applications, IEEE Computer Society, 2006, Volume 1 (AINA"06) — Volume 01 AINA "06.

[14] G. Denker, L. Kagal, T. Finin, Security in the semantic web using OWL, Information Security Technical Report 10 (1) (2005) 51–58.

[15] R. Anderson, Security engineering: a guide to building dependable distributed systems, Jonh Wiley & Sons, Inc., 2001.

[16] R. Crook, D. Ince, B. Nuseibeh, Modelling access policies using roles in requirements engineering, Information and Software Technology 45 (14) (2003) 979–991.

[17] P. Devanbu, S. Stubblebine, Software engineering for security: a roadmap, Future of Software Engineering, ACM Press, 2000, pp. 227–239.

[18] E. Ferrari, B. Thuraisingham, Secure database systems, in: M. Piattini, O. Díaz (Eds.), Advanced databases: technology design, Artech House, London, 2000.

[19] P. Giorgini, F. Massacci, J. Mylopoulos, Requirements engineering meets security: a case study on modelling secure electronic trasactions by VISA and Mastercard, Proceedings of the International Conference on Conceptual Modelling (ER), Springer-Verlag, 2003, LNCS 2813.

[20] J. McDermott, C. Fox, Using abuse care models for security requirements analysis, Proceedings of the 15th Annual Computer Security Applications Conference, 1999.

[21] H. Mouratidis, P. Giorgini, Integrating security and software engineering: advances and future vision, Idea Group Publishing, 2006.

[22] V. Raskin, et al., Ontology in information security: a userful theoretical foundation, Proceedings of the 2001 workshop on New security paradigms NSPW'01, ACM Press, 2001.

[23] A. Squicciarini, et al., Achieving privacy in trust negotiations with an ontology-based approach, IEEE Transaction on Dependable and Secure Computing (TDSC) 3 (1) (2006) 13–30.

[24] A. Ekelhart, S. Fenz, T. Neubauer, AURUM: a framework for supporting information security risk management, 42nd Hawaii International Conference on System Sciences (HICSS), IEEE Computer Society, 2009.

[25] J. Undercoffer, A. Joshi, J. Pinkston, Modeling computer attacks: an ontology for intrusion detection, The Sixth International Symposium on Recent Advances in Intrusion Detection, Springer, 2003.

[26] B. Kitchenham, Procedures for performing systematic reviews (Joint Technical Report), TR/SE-0401, Software Engineering Group. Department of Computer Science: Keele University, 2004, p. 33.

[27] B. Kitchenham, Guideline for performing Systematic Literature Reviews in Software Engineering. Version 2.3, University of Keele (Software Engineering Group, School of Computer Science and Mathematics) and Durham (Department of Conputer Science), 2007.

[28] C. Blanco, et al., A systematic review and comparison of security ontologies, Proccedings of The Third International Symposium on Frontiers in Availability, Reliability and Security (FARES), IEEE Computer Society, Barcelona, Spain, 2008.

[29] J. Biolchini, P. Gomes, Systematic Review in Software Engineering, Systems Engineering and Computer Science Department, UFRJ, Río de Janeiro, Brazil, 2005.

[30] E. Blomqvist, A. Öhgren, K. Sandkuhl, Ontology construction in an enterprise context: comparing and evaluating two approaches, Proceedings of the Eighth International Conference on Enterprise Information Systems: Databases and Information Systems Integration, Paphos, Cyprus, 2006.

[31] C. Basile, et al., Ontology-based security policy translation, Journal of Information Assurance and Security 5 (2010) 437–445.

[32] S. Beji, N. Kadhi, Security ontology proposal for mobile applications, International Conference on Mobile Data Management: Systems, Services and Middleware, IEEE Computer Society, 2009, pp. 580–587.

[33] A. Herzog, N. Shahmehri, C. Duma, An ontology of information security, International Journal of Information Security and Privacy 1 (4) (2007) 1–23.

[34] G. Denker, et al., Security for DAML web services: annotation and matchmaking, The SemanticWeb — ISWC 2003, Springer Berlin, Heidelberg, 2003, pp. 335–350.

[35] D. Martin, et al., Bringing semantics to web services with OWL-S, World Wide Web Journal 10 (3) (2007) 243–277.

[36] G. Dobson, S. Hall, G. Kotonya, A domain-independent ontology for non-functional requirements, International Conference on e-Business Engineering (ICEBE), 2007, pp. 563–566.

[37] S. Fenz, E. Weippl, Ontology based IT-security planning, Proceedings of the 12th Pacific Rim International Symposium on Dependable Computing PRDC '06, IEEE Computer Society, 2006, pp. 389–390.

[38] S. Fenz, et al., Information Security Fortification by Ontological Mapping of the ISO/IEC 27001 Standard. in accepted for the Proceedings of the 13th IEEE Pacific Rim International Symposium on Dependable Computing, 2007. 2007.

[39] A. Ekelhart, et al., Security ontology: simulating threats to corporate assets, Information Systems Security (ICISS'06), 2006, pp. 249–259.

[40] A. Ekelhart, et al., Security ontology: improving quantitative risk analysis, 40th Hawaii International Conference on System Sciences (HICSS'07), IEEE Computer Society, Los Alamitos, CA, USA, 2007.

[41] G. Goluch, et al., Integration of an ontological information security concept in risk-aware business process management, 41st Annual Hawaii International Conference on System Sciences, 2008, p. 377.

[42] S. Fenz, A. Ekelhart, Formalizing information security knowledge, ACM symposium on information, computer and communications security (ASIACCS), 2009.

[43] S. Fenz, Ontology-based generation of IT-security metrics, ACM Symposium on Applied Computing (SAC), ACM, Sierre, Switzerland, 2010, pp. 1833–1839.

[44] A. Avizienis, et al., Basic concepts and taxonomy of dependable and secure computing, IEEE Transactions on Dependable Secure Computing 1 (1) (2004) 11–33.

[45] ISO/IEC, ISO/IEC 27001, Information technology — Security techniques — Information security management systems — Requirements, 2005.

[46] A. García-Crespo, et al., SecurOntology: A semantic web access control framework, Computer Standard and Interfaces 33 (1) (2011) 42–49.

[47] M. Karyda, et al., An ontology for secure e-government applications, First International Conference on Availability, Reliability and Security (ARES'06), IEEE Computer Society, 2006, pp. 1033–1037.

[48] A. Kim, J. Luo, M. Kang, Security ontology for annotating resources, 4th International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE'05), Agia Napa, Cyprus, 2005.

[49] A. Kim, J. Luo, M.H. Kang, Security ontology to facilitate web services description and discovery, Journal on Data Semantics 9 (2007) 167–195.

[50] S.-W. Lee, et al., Building problem domain ontology from security requirements in regulatory documents, Proceedings of the 2006 international workshop on Software engineering for secure systems, ACM Press, Shanghai, China, 2006.

[51] S.-H. Li, K.-C. Wang, Applications of ontology in management of information asset, International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), IEEE Computer Society, 2009, pp. 230–233.

[52] F.-H. Liu, W.-T. Lee, Constructing enterprise information network security risk management mechanism by ontology, Tamkang Journal of Science and Engineering 13 (1) (2010) 79–87.

[53] H. Mouratidis, P. Giorgini, G. Manson, An ontology for modelling security: the tropos approach, Knowledge-Based Intelligent Information and Engineering Systems, Springer Berlin, Heidelberg, 2003, pp. 1387–1394.

[54] P. Giorgini, H. Mouratidis, N. Zannone, Modelling security and trust with secure tropos, Integrating Security and Software Engineering: Advances and Future Visions, Idea Group Publishing, 2006.

[55] K. Bimrah, H. Mouratidis, D. Preston, Information systems development: a trust ontology, OTM Workshops, 2007, pp. 25–26.

[56] S. Islam, H. Mouratidis, S. Wagner, Towards a framework to elicit and manage security and privacy requirements from laws and regulations, Requirements Engineering: Foundation for Software Quality, Springer Berling, Heidelberg, 2010, pp. 255–261.

[57] E. Yu, L. Liu, Mylopoulos, A social ontology for integrating security and software engineering, Integrating Security and Software Engineering: Advances and Future Visions, Idea Group Publishing, 2006.

[58] S. Parkin, A. Moorsel, R. Coles, An information security ontology incorporating human-behavioural implications, 2nd International Conference on Security of Information and Networks, Famagusta, North Cyprus, 2009, pp. 46–55.

[59] B. Tsoumas, et al., Security-by-ontology: A knowledge-centric approach, in: S. Boston (Ed.), Security and Privacy in Dynamic Environments, 2006, pp. 99–110.

[60] S. Dritsas, et al., OntoSPIT: SPIT management through ontologies, Computer Communications 32 (1) (2009) 203–212.

[61] CRAMM, CRAMM, United Kingdom Central Computer and Telecommunication Agency, CCTA Risk Analysis and Management Method: User Manual, ver. 5.1, HMSO, 2005.

[62] COBIT, COBIT, IT Governance Institute, Control Objectives for Information and related Technology (COBIT 4.0), 2005.

[63] A. Vorobiev, J. Han, Security attack ontology for web services, Proccedings of the Second International Conference on Semantics, Knowledge, and Grid SKG '06, IEEE Computer Society, 2006, p. 42.

[64] A. Vorobiev, N. Bekmamedova, An ontology-driven approach applied to information security, Journal of Research and Practice in Information Technology 42 (1) (2010) 61–76.

[65] J. Wang, M. Guo, J. Camargo, An ontological approach to computer system security, Information Security Journal: A Global Perspective 19 (2) (2010) 61–73.

[66] J. Zhou, E. Niemelä, A. Evesti, Ontology-based software reliability modelling, Proceedings of Software and Services Variability Management Workshop — Concepts, Models and Tools. Helsinki, Finland, 2007, pp. 17–31.

[67] J. Zhou, et al., OntoArch: approach for reliability-aware software architecture development, 32nd Annual IEEE International Computer Software and Applications Conference (COMPSAC), IEEE Computer Society, Turku, Finland, 2008, pp. 1228–1233.

[68] J. Zhou, E. Niemela, P. Savolainen, An Integrated QoS-Aware Service Development and Management Framework, WICSA, 2007, p. 13.

[69] M. Sabou, et al., Evaluating the semantic web: a task-based approach, Proc. of ASWC/ISWC, 2007.

[70] A. Lozano-Tello, A. Gómez-Pérez, ONTOMETRIC: a method to choose the appropriate ontology, Journal of Database Management. Special Issue on Ontological analysis, Evaluation, and Engineering of Business Systems Analysis Methods 15 (2) (2004).

[71] A. Lozano-Tello, Métrica de idoneidad de ontologías. Ph.D. thesis, in Departamento de Informática. 2002, Universidad de Extremadura.

[72] M. Sabou, et al., Ontology selection: ontology evaluation on the real semantic web, Proc. of the EON Workshop, 2006.

[73] J. Brank, M. Grobelnik, D. Mladenic, A survey of ontology evaluation techniques, Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005), Citeseer, Ljubljana, Slovenia, 2005.

[74] MAGERIT, MAGERIT, Methodology for Information Systems Risk Analysis and Management, Available from: http://www.csi.map.es/csi/pg5m20.htm, 2006.

[75] E.F. Hill, Jess in Action: Java Rule-Bases Systems, Managing Publications co, Greenwich, CT,USA, 2003.

[76] A. Hameed, D.H. Sleeman, A. Preece, Detecting mismatches in experts' ontologies acquired through knowledge elicitation, Research and Development in Intelligent Systems XVIII, Springer, 2001, pp. 9–22.

[77] N.F. Noy, M.A. Musen, The PROMPT suite: interactive tools for ontology merging and mapping, International Journal of Human Computer Studies 59 (2003) 983–1024.

[78] U. Reimer, Knowledge integration for building organizational memories, Proceedings of the 11th Banff Knowledge Acquisition for Knowledge Based Systems Workshop, 2, KM-6.1, KM-6.20, 1998.

[79] H.S. Pinto, J.P. Martins, Ontology integration: how to perform the process, International Joint Conference on Artificial Intelligence, 2001.

[80] D.L. McGuiness, et al., An environment for merging and testing large ontologies, in: A. Cohn, F. Giunchiglia, B. Selman (Eds.), KR2000: Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, San Francisco, USA, 2000, pp. 483–493.

[81] A. Gómez-Pérez, M. Fernández-López, O. Corcho, A. Gómez-Pérez, M. Fernández-López, O. Corcho, Ontological engineering, 1st edSpringer, London, 2004.

[82] M. Fernández, A. Gómez-Pérez, J. Pazos, Building a chemical ontology using METHONTOLOGY and the ontology design environment, IEEE Intelligent Systems 14 (1) (1999) 37–46.

[83] ISO/IEC, ISO/IEC 15408-1, Information technology. security techniques. evaluation criteria for ti security. Part I: introduction and general model, ISO/IEC, Switzerland, 1999.

[84] C. Alberts, A. Dorofee, Managing information security risks: The OCTAVE (SM) approach, Addison Wesley, Boston, 2002.

[85] T.H. Duong, et al., Complexity analysis of ontology integration methodologies: a comparative study, Journal of Universal Computer Science 15 (4) (2009) 877–897.

[86] N.T. Nguyen, Advanced Methods for Inconsistent Knowledge Management, ed. L. Springer-Verlag. 2008: Springer-Verlag, London.

[87] E. Sirin, B. Parsia, Pellet: An OWL DL reasoner, Proc. of the 2004 Description Logic Workshop (DL 2004), 2004, pp. 212–213.

[88] A. Toval, et al., Requirements reuse for improving information systems security: a practicioner's approach, Requirements Engineering Journal 6 (4) (2002) 205–219, Springer.

[89] A. Toval, A. Olmos, M. Piattini, Legal requirements reuse: a critical success factor for requirements quality and personal data protection, IEEE Joint International Conference on Requirements Engineering (ICRE'02 and RE'02), Essen, Germany, 2002.

[90] ISO/IEC, ISO/IEC 15408 (Common Criteria v3.0), Information technology security techniques-evaluation criteria for IT security, 2005.

[91] J. Lasheras, et al., An ontology-based framework for modelling security requirements, The 6th International Workshop on Security In Information Systems (WOSIS-2008), INSTICC Press, Barcelona (Spain), 2008.

[92] D. Geneiatakis, C. Lambrinoudakis, An ontology description for SIP security flaws, Computer Communications 30 (6) (2007) 1367–1374.

[93] L. Kagal, T. Finin, Modeling conversation policies using permissions and obligations, AAMAS workshop on Agent communication, LNCS, Springer-Verlag, 2005.

[94] J. Kwon, C.-J. Moon, Visual modeling and formal specification of constraints of RBAC using semantic web technology, Knowledge-Based Systems 20 (4) (2007) 350–356.

[95] Z. Maamar, N.C. Narendra, S. Sattanathan, Towards an ontology-based approach for specifying and securing Web services, Information and Software Technology 48 (7) (2006) 441–455.

[96] J. McGibney, N. Schmidt, A. Patel, A service-centric model for intrusion detection in next-generation networks, Computer Standards & Interfaces 27 (5) (2005) 513–520.

[97] J.J. Tan, S. Poslad, Dynamic security reconfiguration for the semantic web, Engineering Applications of Artificial Intelligence 17 (7) (2004) 783–797.

[98] B. Thuraisingham, Security standards for the semantic web, Computer Standards & Interfaces 27 (3) (2005) 257–268.

Carlos Blanco has an MSc in Computer Science from the University of Castilla-La Mancha. He is currently a PhD student and a member of the GSyA Research Group at the School of Computer Science at the University of Castilla-La Mancha (Spain). His research activity is in the field of security in Information Systems focused on Data Warehouses, OLAP tools, MDD and Ontologies. He is the author of several papers on these topics.

Joaquín Lasheras is a PhD student at the University of Murcia, in Spain. He received a degree in computer science from the University of Murcia. He is a member of the Software Engineering research group of the Department of Informatics and System (www.um.es/giisw) whose research manager is Professor José Ambrosio Toval Álvarez. His current research interests include requirements engineering, reuse, ontologies and security. He is involved in a variety of applied research and development projects with industry and networks related to security and quality.

Dr. Eduardo Fernández-Medina holds a PhD in Computer Science from the University of Castilla-La Mancha. He leads the GSyA Research Group of the Department of Computer Science at the University of Castilla-La Mancha. His research activity is in the field of security in databases, data warehouses, web services and information systems, and also in security metrics. Fernández-Medina is a coeditor of several books and book chapters on these subjects and has presented several dozens of papers at national and international conferences (DEXA, CAISE, UML, ER, etc.). He is the author of several manuscripts in national and international journals (DSS, ACM Sigmod Record, IS, IST, C&S, ISS, etc.) and belongs to various professional and research associations (AEC, ISO, IFIP WG11.3, etc.).

Dr. Rafael Valencia-García received his BA, MSc and PhD degrees in Computer Science from the University of Murcia. He is a Lecturer at the Department of Informatics and Systems, University of Murcia. His main research interests are Natural Language Processing and the application of Knowledge Technologies such as ontologies. He has published over 25 articles in journals, conferences and book chapters. He is the author or coauthor of several books.

Dr Ambrosio Toval Álvarez is a full professor at the University of Murcia, in Spain. He holds a BS degree in Mathematics from the University Complutense of Madrid, and received a Ph.D. in Computer Science (cum laude) from the Technical University of Valencia (both in Spain). He is involved in a variety of applied research and development projects with industry and conducts research in the design and implementation of conceptual UML model verification, requirements engineering processes and computer-aided requirement engineering tools, and security requirements. Dr. Toval is currently the Head of the Software Engineering Research Group, at the University of Murcia.