An Ontology-based Framework for Modelling Security Requirements

Joaquín Lasheras, Rafael Valencia-García
Jesualdo Tomás Fernández-Breis and Ambrosio Toval

Department of Informatics and Systems. University of Murcia
30071 Campus de Espinardo, Murcia, Spain

Abstract. In the last years, security in Information Systems (IS) has become an important issue, so that it has to be taken into account in all the stages of IS development, including the early phase of Requirements Engineering (RE). One of the most helpful RE strategies for improving the productivity and quality of software process and products is the reuse of requirements, and this can be facilitated by Semantic Web technologies. In this work, we describe a novel ontology-based framework for representing and reusing security requirements based on risk analysis. A risk analysis ontology and a requirement ontology have been developed and combined to represent formally reusable security requirements and improve security in IS, detecting incompleteness and inconsistency in requirements and achieving semantic processing in requirements analysis. These ontologies have been developed according to a formal method to build and compare ontologies and with a standard language, OWL. This framework will be the basis to elaborate a "lightweight" method to elicit security requirements.

1 Introduction

Information confidentiality, security or privacy, issues of interest for Information System designers, are nowadays critical and vital issues for the society [1]. Hence, security has to be taken into account in all the stages of the software development process [2]. These include the early phases related to *Requirements Engineering* (RE) [3], where, security has been identified as a research hotspot [4].

Security requirements include the types and levels of protection necessary for equipment, data, information, applications, and facilities to meet security policy. Specifically, security requirements are identified by *risk analysis* - "*the systematic use of information to identify sources and to estimate the risk"* [5]. Risk analysis is one of the three sources identified by the security standard ISO 27002 – "*Code of Practice for Information Security Management*" [5] – to identify security requirements. The others two sources are related to legal, regulatory and contractual requirements of an organization and with the principles, objectives and business requirements for information processing that an organization have developed to support its operations.

In this context, *reuse* comes out as an important factor to achieve efficient requirements management. There is a consensus [6] on the many benefits of reuse, becoming more important as the abstraction level augments; not only code, but also

Lasheras J., Valencia-García R., Tomás Fernández-Breis J. and Toval A. (2008).
An Ontology-based Framework for Modelling Security Requirements.
In *Proceedings of the 6th International Workshop on Security in Information Systems*, pages 78-88
DOI: 10.5220/0001743400780088

designs and specifications, are reused [7]. So, methods and technologies for facilitating requirements reuse are needed, including security requirements [8].

The *Semantic Web* community has experience in the design and development of reusable components. *Ontologies* are its backbone technology [9] and have become widely used due to their advantages (*reusability* and *shareability*) [10]. An ontology represents a common, reusable and sharable - since it captures knowledge which has the consensus of the community [11]- view of a particular application domain. The benefits of using the ontological technology in terms of information systems' security is stated in [12] according to three main properties: (1) the ontology organizes and makes it systematic any phenomenon at any detail level and reduces the diversity of items to a properties list; (2) many approaches take advantage of the modularity it induces, for instance, to establish relations among measurements to detect some properties; and (3) an ontological approach provides mechanisms to forecast security problems. Several authors [13, 14] consider the definition of a security ontology a challenge within the community of security engineering.

In this work an ontology-based framework for representing, storing and reusing security requirements is presented. This framework is based on risk analysis, permitting a formal representation (intelligible by a machine) of the *requirements*, their *metainformation,* their *relationships* and the *constraints*, *axioms* and *rules* derived of their use (*semantic relationships*). This framework combines a risk analysis ontology, based on methods of risk analysis and security standards, and a requirements ontology (based on our RE method SIREN [15]).

This paper has been structured as follows: Section 2 presents the ontology-based framework for modelling security requirements. First the risk analysis ontology (Section 2.1) and the requirements ontology (Section 2.2) are described, and then its combination (Section 2.3) and its application (Section 2.4) are showed. Section 3 presents related work and, finally, Section 4 shows the conclusions and further work.

2 Ontology-based Framework for Modelling Security Requirements in Risk Analysis

In this framework, knowledge is represented through ontologies. In this work, the ontologies have been implemented by using the Ontology Web Language (OWL), which is the current W3C recommendation for exchanging semantic content on the Web. OWL has different flavors, being OWL-DL the chosen one since it has the expressivity needed and allows for complete reasoning. Our framework for modelling security requirements is based on two ontologies: the risk analysis ontology (Section 2.1) and the requirements ontology (Section 2.2). The first one conceptualizes the risk analysis domain including concepts such as assets, or threats, and is based on risk analysis methods and standards of security. On the other hand, the requirements ontology models reusable requirements, with their metainformation and relationships. In Section 2.3, the combination of both ontologies is showed, which will permit to specify security requirements with all its metainformation, relationship and semantic properties (constraints, axioms and rules). In Section 2.4, the application of the framework is presented.

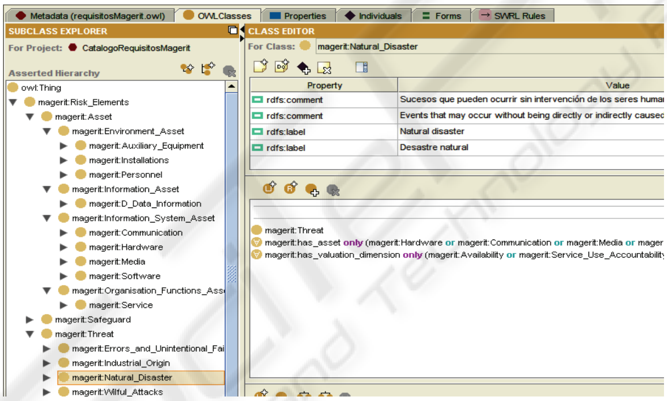
2.1 Risk Analysis Ontology

The risk analysis ontology is based on MAGERIT [16], the information systems risk analysis and management method of the Spanish public administration. It conforms to the ISO/IEC 15408-1999 [17] and is based in 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 [16] for details).

MAGERIT defines a document with the elements that must appear in a risk analysis project. This document, named "*catalogue of elements*", has two purposes in a risk analysis and management project [16]:

- To offer a standard item for quick consultation, centred on the specifics of the system being analysed, to facilitate the work of the people involved in the project.
- To provide uniform results of the analysis, promoting terminology and criteria that allow to compare and even integrate the analyses made by different teams.

In MAGERIT, the relationships between the elements are explained by means of tables and natural language text. In this work, this information has been formalized in an ontology (see Fig. 1 for a partial representation of its structure). The current version of this ontology is available in http://dis.um.es/~jolave/RiskElements.owl.



For Project: CatalogoRequisitosMagerit

```
owl:Thing
▼ magerit:Risk_Elements
  ▼ magerit:Asset
    ▼ magerit:Environment_Asset
      ▶ magerit:Auxiliary_Equipment
      ▶ magerit:Installations
      ▶ magerit:Personnel
    ▼ magerit:Information_Asset
      ▶ magerit:D_Data_Information
    ▼ magerit:Information_System_Asset
      ▶ magerit:Communication
      ▶ magerit:Hardware
      ▶ magerit:Media
      ▶ magerit:Software
    ▼ magerit:Organisation_Functions_Ass
      ▶ magerit:Service
  ▶ magerit:Safeguard
  ▼ magerit:Threat
    ▶ magerit:Errors_and_Unintentional_Fai
    ▶ magerit:Industrial_Origin
    ▶ magerit:Natural_Disaster
    ▶ magerit:Wilful_Attacks
```

For Class: magerit:Natural_Disaster

Property	Value
rdfs:comment	Sucesos que pueden ocurrir sin intervención de los seres huma
rdfs:comment	Events that may occur without being directly or indirectly caused
rdfs:label	Natural disaster
rdfs:label	Desastre natural

```
magerit:Threat
magerit:has_asset only (magerit:Hardware or magerit:Communication or magerit:Media or mager
magerit:has_valuation_dimension only (magerit:Availability or magerit:Service_Use_Accountabilit
```

Fig. 1. Taxonomy and axioms of the elements of the risk analysis ontology in the Protégé Editor.

This ontology identifies five groups of *elements*:

- Asset: anything that provides value to the organization. It is classified within a hierarchy of types of assets.
- Valuation dimension: the features or attributes that make an asset valuable. It is the measurement of the loss caused by damages in an asset in a certain dimension: availability, integrity, confidentiality, authenticity and accountability.
- Threat: the possible threats to the assets in an information system.
- Safeguard: the safeguards that allow threats to be faced.

The risk analysis ontology contains relations, constraints, axioms and rules. For example, there is a binary relationship between the assets and the threats to represent which *threat* can affect to which *asset*. The semantics of this generic relation is completed at each concept (of the *assets* and *threats* taxonomies) by adding the corresponding range and domain constraints in OWL (see Fig. 1). For example, *software (asset)* cannot be affected by *natural disasters (threat)*, and the *errors and unintentional failures (threat)* affect to the s*oftware (asset)* but not to the *personnel (asset)*. Moreover, other constraints such as disjointness or cardinality can also be defined in OWL and have been very useful for the construction of this ontology (and the followings described above). However, not every type of rule can be directly formalized in OWL. In this case, rules languages, such as the *Semantic Web Rule Language* (SWRL), play an important role in combination with semantic reasoners.

2.2 Requirements Ontology

The concepts, metainformation and relationships included in the requirements ontology have been mostly taken from our experience in the context of requirements reuse, specifically in the reuse-based RE method called SIREN [15, 18]. SIREN could be considered both a document-based and a repository-based approach since it builds upon a reusable requirements repository organized by catalogs based on RE standards, specifically IEEE [19, 20]. To date, these requirements are represented textually (organized in the IEEE documents [19, 20]), with their metainformation associated to them. Fig. 2 shows a part of the taxonomy of the requirements ontology. There, requirements are classified according to the IEEE documents standards. Next, the main metainformation elements (*attributes*) identified for each requirement, and their OWL modelling, are described:

- Requirements are characterized by a **unique identifier** and a **textual description**. Both have been defined in OWL as *Datatype properties*.
- **Priority**: this value must be established by the analyst and shows the order of development *Datatype property* {“*high*”, “*medium*”, “*low*”}
- **Rationale:** why the requirements are considered. *Datatype property*.
- **State:** 9 states are possible for a requirement. *Datatype property {To be Determined, Determined, To be Revised, To Rule out, Approved, Modelled in Analysis, Modelled in Design, Implemented or Verify}*
- **Traceability**: in a requirements document the requirements can appear related to each other by means of traces. The traceability model includes three types of relationships: *inclusive relationships, parent-children relationships* and *exclusive relationships* which have been modelled using the following *Object properties*:
    - **Inclusive**: the *inclusive traceability relationships* are defined between two requirements A and B, which means that to satisfy A, B also needs to be satisfied and, therefore, the reuse of A will imply the reuse of B. This is a dependence relation and satisfies a set of properties (*reflexivity, asymmetry* and *transitivity*). This relationship has been modeled in OWL using 2 inverse object properties (*trace_to* and *trace_from*), being both transitive.
    - **Parent**-**Child**: They are relationships through which the children requirements refine the meaning of the parent ones. This relation has been modeled as the previous relationship.
    - **Exclusive**: the *exclusive traceability relationships* mean that the requirements implied are mutually exclusive. This relationship is directly related to reuse since it indicates those requirements that cannot be selected from the repository for the same project. This relation has been modeled by an *Object property* with the *symmetry* property of OWL.

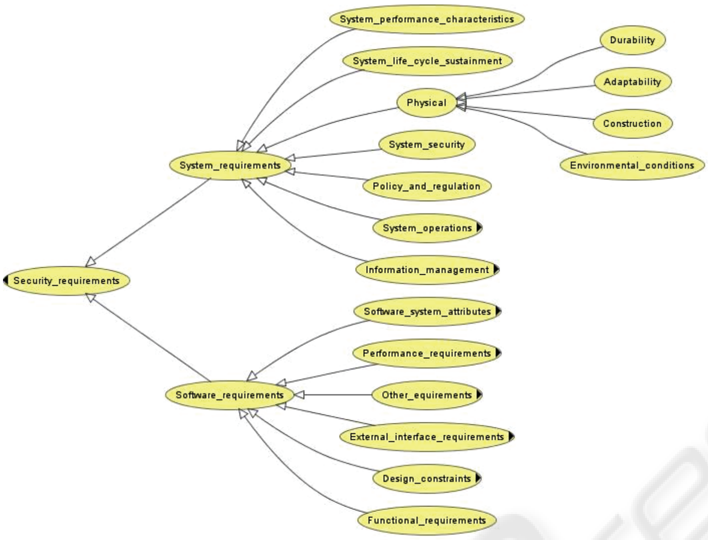


Fig. 2. Requirements Taxonomy (an extract).

- **Source**: It is the source of the requirement. Although the client needs are the main source, others requirements could be derived from the technical solution, current legislation or standards (i.e., security standards). References, URL, the name of standards or even the source catalog, if the requirement has been reused, must be collected. *Datatype property*.
- **Verification Method**: This is the method used to verify that the requirement is satisfied in the final product. *Datatype property* {*"inspection", "analysis", "demonstration", "test"*}
- **Section**: The document section in which the requirement is situated, if any. *Datatype property*.

2.3 Combining Ontologies: The Security Requirements Ontology

As described in the introduction, this work is focused on *security requirements* identified by *risk analysis,* which is one source to elicit security requirements. identified by the security standard ISO 27002 [5]: "*One source is derived from assessing risks to the organization, taking into account the organization's overall business strategy and objectives. Through a risk assessment, threats to assets are identified …"*. These requirements, which meet security policy, can be functional or non-functional (software or system requirements), supporting some security issues in the system. Consequently, the requirements ontology described in section above 2.2 has been used to classify the security requirements (Fig.2), extending it with concepts from the risk analysis ontology (Section 2.1). Then, the metainformation related to risk analysis (such as assets or threats) and the constraint, axioms and rules that help to maintain consistency in the security requirements have been modelled.

Although the main basis to identify this new metainformation has been MAGERIT (Section 2.1), other instructions from the family of security standards ISO 27000, specifically by the ISO 27002 [5], have been considered, so the results could be adapted and certified under these standards in the future.

The new properties, and their OWL modelling, are described next:

- **has_asset**: Every security requirement has to be related to one asset. So, an *Object property* has been added to the concept security requirement to represent its asset associated. The range of the *Object property* is the class *Asset* defined in the risk analysis ontology. This *Object property* is inherited and restricted along the hierarchy. For example a *physical system requirement* can only be associated to *hardware assets*. Furthermore storing the *owner*, the *person and the unit responsible for* the asset are relevant. This information is an objective in ISO 27002 (*Section 7 - Asset Management)*.
- **has_threats:** It represents possible threats associated to the non-fulfilment of the requirements. The risk analysis ontology has constraints of which threat can be occurred to which asset, so the system can infer if a threat associated to a requirement can be inconsistent with the asset associated to this requirement. For example the user cannot relate a *physical system requirement* (which has as asset "*hardware system"*) to the threat "*repudiation*", associated to the asset of "*Services*". This property is represented by an *Object property* over the hierarchy of threats of the risk analysis ontology. In [16] are described the different kind of threats. Furthermore, information about the *effect of the threat*, about its *probability of occurrence* and *previous record* must be stored.
- **valuation_criteria:** The purpose is to provide relative values of the assets in their various valuation dimensions. It is a number *Datatype property* (1-10) that values the importance of the requirement for the system. MAGERIT describes such possible values [16]. Besides, the *Datatype property* "*rationale_of_valuation_criteria*" includes the rationale for selecting this value.
- **has_valuation_dimensions:** the features that make an asset valuable. There exist five valuation dimensions modelled using an *Object property*: "*Availability*", "*Integrity*", "*Confidentiality*", "*Accountability*" and "*Authenticity*".
- **has_safeguards:** This *Object property* associates to the safeguard its related requirement. Information about the *efficacy to confront a threat* and its *state of implantation* must be stored.

With this combination, the **Source** of the requirement becomes essential. It specifies the current legislation or security standards a requirement has been derived from. This is useful for the application of the framework shown in the next section.

2.4 Application of the Framework

In Security Information, a basic (baseline) protection must be implemented in all systems except for particular situations. This type of reasoning is frequently applied and leads to the deployment of a minimum of "purely common sense" safeguards [16]. There are numerous sources to identify these safeguards, including international standards (such as ISO 27002 [5] or CCF [17]), national standards or regulations (such as protection data laws), and sector standards.

In that way, our framework can be used as a source to specify a baseline protection, by using the security requirements ontology and the properties that it models. This ontology represents a "catalog" of security requirements that can be a useful starting point for further refinement in the system. Furthermore, protection by catalog can be refined somewhat by considering the value of the assets or quantifying the threats [16]. This is achieved thanks to all the properties, relationships and semantic properties, of the ontologies, permitting to elicit and specify requirements, for instance, by assets or by probably threats to the system.

On the other hand, our framework covers a very wide spectrum of interests, accounting for all types of situations in security. In practice, the user may face situations in which the analysis is more restricted, for instance, files affected by legislation regarding personal data or communications security. In this case, the metainformation related to the requirement through the "attribute" source must be considered as a layer to search and identify requirements.

The advantages of protection by catalogue are quickness, need for little effort (once the catalogue has been developed), and standardization, providing uniform results with other similar organisations. An additional advantage of using this framework is the possibility of identifying the measurement of percentage of satisfied requirements. In [21], a catalog of reusable requirements related to Personal Data Protection was applied to the process of auditing a case study in a Health Information System. Here, the % of the satisfied requirements from the catalog was presented as a measurement of security.

Application to a Reusable Requirements Repository. In previous work [18], a reusable requirements catalog related to security with about 350 requirements (with their metainformation and traceability relationships) was defined. This new framework allows for verifying the consistency of the associated metainformation, using the properties - constraint, axioms and rules - identified in Section 2.3 (*has_asset*, *has_threats* …). For example, some inconsistencies were identified by the proper instantiation process, such as checking that the traceability of the requirements is free of cycles, that a requirement associated to a determined asset has to be related to a possible threat, or that a safeguard is applicable to this asset. Specifically 27 inconsistencies have been detected: 8 were related to the traceability of the requirements, and 19 were identified by constraint of risk analysis. In table 1, some details about the ontologies are showed.

Table 1. Ontology details.

	Risk analysis ontology	**Security Requirements ontology**
Classes	266	46
Individuals	280	350
Number of datatype properties	6	21
Number of object properties	6	9
Restrictions	140	52
Disjoint Axioms	30	15

An Example of Security Requirement. Let us suppose that the requirement to model is a high-level one (closed to the stakeholders and the problem-realm), and it is defined in natural language:

"*The installations will be built earthquake-proof*"

First, the taxonomic category of the requirement has to be identified. This requirement has to be inserted into the class "*System requirement- Physical - Construction*" (Fig. 2). Each property of this new individual has to be filled in (Fig. 3). The system allows for associating the requirement only with instances of the asset

"*Installations*", due to the constraints defined in the ontology. Besides, due to the asset of this requirement (“*Installation”)*, the user cannot relate to it any *threat* of the hierarchies (Fig. 1) associated to "*Wilful attacks*" (deliberate failures caused by people) or “*Error and unintentional failure*” (unintentional failures caused by people).

has_asset: *Installations*

  Owner: *the company*

  Responsible for: *the building of the company*

has_valuation_dimensions: *availability*

valuation_criteria: 7

  Rationale: *is likely to cause damage to the operational effectiveness or security of the Operations / Logistics mission*

has_threats: *Natural Disaster*

  Probability of occurrence: *probability of earthquake in the city of the company.*

  Previous record: *information about this threat in other branches of the company.*

has_safeguards: *Certification*

  State of implantation: *we get a certification that the installations are earthquake-proof*

  Efficacy to confront a threat: *% of buildings certified that have recovered from the threat.*

Fig. 3. Properties of the requirement, an extract.

The **source** attribute allows for identifying that this requirement is associated to the ISO 27002 control objective - 9.1.4 *Protecting against external and environmental threats*. However, this requirement has not a direct correspondence to requirements extracted by Protection Data Laws. Nevertheless, thanks to the relationships of traceability between requirements, and the relation between threats-assets-requirements, it has been detected that, if data regulated by protection laws is stored in these installations, then the requirement must be considered.

3 Related Work

Several authors consider the definition of security ontology an important investigation area and a challenge within the community of security engineering [13, 14]. In our previous works, a systematic review and comparison (based on the formal method [22]) of Security Ontologies was made [23], which has been the basis for identifying the contribution of this work.

In security, ontologies concerning dependability domain [24], trust domain [25] and to other functional issues about security (algorithms Security or Policy security) [26] have been used to provide an unified conceptualization, but none of them are associated to risk analysis. In [13] the authors present an ontology of risk analysis based on standards used in a security management framework for information systems, and in [27] is centred in small and medium companies. Nevertheless, none of them has integrated it with RE techniques in order to take care of their benefits and the used of reuse requirements catalogs. [28] proposes a methodology to elicit and certify security requirements, although it is particularly adapted to the Department of Defense, *Information Technology Security Certification and Accreditation Process* (DITSCAP). The ontology proposed there is not related to RE techniques, and catalogs of requirements are not used nor semantics is managed.

Consequently, although several related work has been identified, none of them combines the use of method of RE, reuse, ontologies and pre-existing catalogs, as the basis of a “lightweight” method to elicit security requirements following security standards, as the ISO 27002 [5], with the aim that the results obtain could be adapted and certified in future under these standards. Besides, none of them have followed a formal method to the development of the ontologies.

4 Conclusions and Further Work

This work, proposes an ontological representation for reusable requirements, which allows for detecting *incompleteness* and *inconsistency* in requirements and achieving semantic processing in requirements analysis without rigorous NLP (*Natural Language Processing*) techniques. The ontologies have been developed according to a formal method to build and compare ontologies [22] and implemented in a standard language, OWL. Besides, their definition is based on requirements [19, 20] and security standards and regulations [5, 16]. So the results obtained could be adapted and certified in future under these standards.

This framework will be the basis to elaborate a *“lightweight” method* to elicit security requirements which permits us to reduce significantly the impact of risks without making large investment in software, by arranging the management of the security. We permit users or developers (without being security experts) to identify security requirements through reuse, with the advantages of using a method based on “catalogs” (see Section 2.4). A measurement of security is also offered by identifying the % of satisfied requirements with respect to standards or regulation.

Furthermore, thanks to the property of *shareability* and *reuse* of the ontologies, the community can benefit from having a shared set of reusable security requirements. Besides, the framework could be extended with other ontologies related to security. Work in progress is focused on such issue, considering the ontologies identified [23]. In that way, we are also planning to extend the risk analysis top level ontology using others wide-accepted standards, such as the family ISO/IEC 27000, the CCTA Risk Analysis and Management Method (CRAMM) or OCTAVE (*Operationally Critical Threat, Asset and Vulnerability Evaluation* by the Carnegie-Mellon). These methods are used once the architectural design has been defined, allowing only an ”*a posteriori*” approach of IT security, resulting in a gap between security requirements and business security needs. On the other hand, as further work, our ontologies might be used to perform semantic searches by using Semantic Web oriented languages such as SPARQL, and we expect to improve the selection of the requirements with the use expert system to prioritize the requirements.

Acknowledgements

This work has been partially financed by the Spanish Ministry of Science and Technology, projects TIC2006-15175-C05-03 and TSI2007-66575-C02-02, by the Government of Murcia under project TIC-INF 06/01-0002; by the Junta de Castilla-

La Mancha (Spain), project PBC-05-012-3, and by the FEDER and the Junta de Castilla-La Mancha (Spain), project PBC-05-012-1.

References

1. Smith, S.W.Spafford, E.H., *Grand Challenges in Information Security: Process and Output.* IEEE Security & Privacy, 2(1): (2004). p. 69-71.
2. Devanbu, P.Stubblebine, S., *Software engineering for security: a roadmap.* ACM Press. Future of Software Engineering: (2000). p. 227-239.
3. Jürjens, J., *Secure Systems Development with UML*: Springer (2005).
4. Cheng, B.Atlee, M. *Research Directions in Requirements Engineering*. in *Future of Software Engineering 2007 (FOSE 2007)* Minneapolis, Minnesota (2007).
5. ISO27002, *ISO/IEC 17799-27002 Code of Practice for Information Security Managament.* (2005).
6. Rothenberger, M.A., Dooley, K.J., Kulkarni, U.R., Nada, N., *Strategies for Software Reuse: A Principal Component Analysis of Reuse Practices.* IEEE Trans. on Soft. Eng., 29(9): (2003). p. 825-837.
7. Sommerville, I., *Software Engineering (7th edition)*: Pearson Education Limited (2004).
8. Firesmith, D., *Specifying Reusable Security Requirements.* Journal of Object Technology, 3(1): (2004). p. 61-75.
9. Berners-Lee, T., Hendler, J., Lassila, O., *The Semantic Web*, in *Scientific American*.(2001): http://www.scientificamerican.com.
10. Brewster, C.O'Hara, K., *Knowledge Representation with Ontologies: The Present and Future.* IEEE Intelligent Systems, 19:1: (2004). p. 72-73.
11. Gruber, T., *Towards Principles for the Design of Ontologies used for Knowledge Sharing.* International Journal of Human-Computer Studies, 43(5/6): (1995). p. 907-928.
12. Raskin, V., Hempelmann, C.F., Triezenberg, K.E., Nirenburg, S. *Ontology in Information Security: A Useful Theoretical Foundation and Methodological Tool*. in *New Paradigms Security Workshop NSPW'01. ACM Press* Clouford, New Mexico, USA (2001).
13. Tsoumas, B.Gritzalis, D., *Towards an Ontology-based Security Management.* Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06). IEEE Computer Society, 1: (2006).
14. Mouratidis, H.Giorgini, P., *Integrating Security and Software Engineering: Advances and Future Visions*: Idea Group Publishing (2007a).
15. Toval, A., Olmos, A., Piattini, M. *Legal Requirements Reuse: A Critical Success Factor for Requirements Quality and Personal Data Protection*. in *IEEE Joint International Conference on Requirements Engineering (ICRE'02 and RE'02)*. Essen, Alemania (2002b).
16. MAGERIT, *Methodology for Information Systems Risk Analysis and Management*: http://www.csi.map.es/csi/pg5m20.htm. (2006)
17. ISO15408, *ISO/IEC 15408 (Common Criteria v3.0) "Information Technology Security Techniques-Evaluation Criteria for IT Security"*.(2005).
18. Toval, A., Nicolás, J., Moros, B., García, F.,*Requirements Reuse for Improving Information Systems Security: A Practicioner's Approach.*Requirements Engineering Journal.Springer,6(4):(2002a).p.205-219.
19. IEEE, *Std 830-1998 Guide to Software Requirements Specifications* in *Volume 4: Resource and Technique Standards*. The Institute of Electrical and Electronics Engineers, Inc. IEEE Software Engineering Standards Collection.(1999).
20. IEEE, *Std 1233-1998 Guide for Developing System Requirements Specifications*, in *Volume 1: Customer and Terminology Standards*. The Institute of Electrical and Electronics Engineers, Inc. IEEE Software Engineering Standards Collection(1999).

21. Martínez, M.A., Lasheras, J., Toval, A., Piattini, M. *An Audit Method of Personal Data Based on Requirements Engineering*. in *The 4th International Workshop on Security In Information Systems (WOSIS-2006)*. Paphos, Chipre (2006).
22. Lozano-Tello, A.Gómez-Pérez, A., *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).
23. Blanco, C., Lasheras, J., Valencia-García, R., Fernández-Media, E., Toval, A., Piattini, M. *A Systematic Review and Comparison of Security Ontologies*. in *International Workshop on Frontiers in Availability, Reliability and Security (FARES) in conjunction with ARES.* Barcelona (2007).
24. Dobson, G.Sawyer, P., *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).
25. Mouratidis, H., Giorgini, P., Manson, G., *An Ontology for Modelling Security: The Tropos Approach*, in *Knowledge-Based Intelligent Information and Engineering Systems*. Springer Berlin / Heidelberg. (2003) p. 1387-1394.
26. Kim, A., Luo, J., Kang, M. *Security Ontology for Annotating Resources* in *4th International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE'05)*. Agia Napa, Cyprus (2005).
27. Fenz, S.Weippl, E.*Ontology based IT-security planning.* Proceedings of 12th Pacific Rim International Symposium on Dependable Computing PRDC '06. IEEE Computer Society: (2006). p. 389-390.
28. Lee, S.W., Gandhi. R.A., *Ontology-based Active Requirements Engineering Framework*. *APSEC* (2005).