Modelling Reusable Security Requirements based on an Ontology Framework

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.
{jolave,valencia,jfernand,atoval}@um.es

In recent years, security in Information Systems (IS) has become an important issue, and needs to be taken into account in all stages of IS development, including the early phase of Requirements Engineering (RE). Reuse of requirements improves the productivity and quality of software process and products. This can be facilitated by Semantic Web technologies. We describe an 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 reusable security requirements formally and to improve security in IS by detecting incompleteness and inconsistency and achieving semantic processing in requirements analysis. This extensible framework is the basis on which to elaborate a "lightweight" method to elicit and specify security requirements, based on security standards.

Keywords: Security Requirements, Requirements Reuse, Risk Analysis, Ontologies

ACM Computing Classification System: D.2.1 Requirements/Specifications, D.2.13 Reusable Software, I.2.4 Knowledge Representation Formalisms and Methods

1. INTRODUCTION

Information confidentiality, security or privacy, issues of interest for Information System designers, are critical and vital matters for today´s society (Smith and Spafford, 2004). Hence, security has to be taken into account in all stages of the software development process (Devanbu and Stubblebine, 2000). These include the early phases related to *Requirements Engineering* (RE) (Jürjens, 2005), where, security has been identified as a research hotspot (Cheng and Atlee, 2007).

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*" ISO 27002 (2005). Risk analysis is one of the three sources identified by the security standard ISO 27002 – "*Code of Practice for Information Security Management*" (ISO27002, 2005) – to identify security requirements. The other two sources are related to the legal, regulatory and contractual requirements of an organization and to the principles, objectives and business requirements for information processing that an organization has developed to support its operations.

Manuscript received: 23 April 2008
Communicating Editor: Eduardo Fernandez-Medina Paton

In this context, reuse comes out as an important factor in achieving efficient requirements management, since negative consequences related to the cost and functionality of an inappropriate requirements management in the system development are well-known (Glass, 2002). There is a consensus (Prieto-Díaz, 1993; Rothenberger, Dooley, Kulkarni and Nada, 2003) on the many benefits of reuse, and these become more important as the abstraction level augments. Not only code, but also designs and specifications, are reused (Cybulsky and Reed, 2000; Sommerville, 2006). Rine and Nada (2000) have proved empirically that the level of reuse determines the effectiveness of the improvements in productivity, quality and time-to-market, and they concluded that greater benefits are obtained when reuse is considered during the early phases of the software development lifecycle. Moreover, Cheng and Atlee (2007) said that "requirements reuse has been pointed out as being one of the most pressing needs and grand challenges in RE research, whose solutions are likely to have the greatest impact on Software Engineering research and practice". So, methods and technologies for facilitating requirements reuse are needed, including security requirements (Firesmith, 2004).

The *Semantic Web* community has experience in the design and development of reusable components. *Ontologies* are its backbone technology (Berners-Lee, Hendler and Lassila, 2001) and have become widely used on account of their advantages (*reusability* and *shareability*) (Brewster, O'Hara, Fuller, Wilks, Franconi, Musen, Ellman and Buckingham, 2004). An ontology represents a common, reusable and sharable – since it captures knowledge which has the consensus of the community (Gruber, 1995) – view of a particular application domain. The benefits of using ontological technology in terms of information systems' security are stated in Raskin, Hempelmann, Triezenberg and Nirenburg (2001) according to three main properties: (1) the ontology organizes and makes any phenomenon systematic 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 between measurements to detect some properties; and (3) an ontological approach provides mechanisms to forecast security problems. Several authors (Tsoumas and Gritzalis, 2006; Mouratidis and Giorgini, 2007a) 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 for a machine) of the *requirements*, their *metainformation*, their *relationships* and the *constraints*, *axioms* and *rules* derived from 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 (Toval, Olmos and Piattini, 2002b)).

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 consistency (Section 2.4). The application of the framework is showed in Section 3. Section 4 presents related work and, finally, Section 5 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 study, 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. 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. The combination of both ontologies is shown in Section 2.3 which will enable the specification of security requirements with all their metainformation, relationships and semantic constraints. Finally, Section 2.4 describes the consistency of the ontologies.

2.1 Risk Analysis Ontology

The risk analysis ontology is based on MAGERIT (MAGERIT, 2006), the information systems risk analysis and management method of the Spanish public administration. It conforms to the ISO/IEC 15408-1999 (ISO15408, 2005) and is based on 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, 2006) 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 (MAGERIT, 2006):

- 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 comparison and even integration of the analyses made by different teams.

The MAGERIT method is 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. Furthermore, in MAGERIT, the relationships between the elements are explained by means of tables and natural language text. In this paper, this information has been formalized in an ontology (see Figure 1 and Figure 2 for a partial representation of its structure). The current version of this ontology is available at http://dis.um.es/~jolave/RiskElements.owl.

This ontology identifies five types of risk elements (see Figure 1) according to (MAGERIT, 2006):

- **Asset:** anything that provides value to the organization. It is classified within a hierarchy of types of assets. MAGERIT distinguishes nine disjoint types of assets: *Service*, *Media*, *Communication*, *Software*, *Hardware*, *Data Information*, *Auxiliary Equipment*, *Installations* and *Personnel*. These assets have been grouped in four disjoint classifications: *Organisation Functions*, *Information System*, *Information*, and *Environment* assets.
- **Threat:** the possible threats associated to the assets in an information system. Four main disjoint types of threats have been identified: *Natural disasters*, *Industrial Origin*, *Errors* and *Unintentional Failures* (unintentional failures caused by people) and *Wilful Attacks* (deliberate failures caused by people).
- **Safeguard:** the safeguards that allow threats to be faced. For example, *access control*, *record of actions* or *back-up copies* (see MAGERIT, 2006 for details).
- **Valuation dimension:** the features or attributes that make an asset valuable. This is the measurement of the loss caused by damages in an asset in a certain dimension: *availability*, *integrity*, *confidentiality*, *authenticity* and *accountability*. For example the availability dimension of an asset means "How important would it be if the asset was not available?"

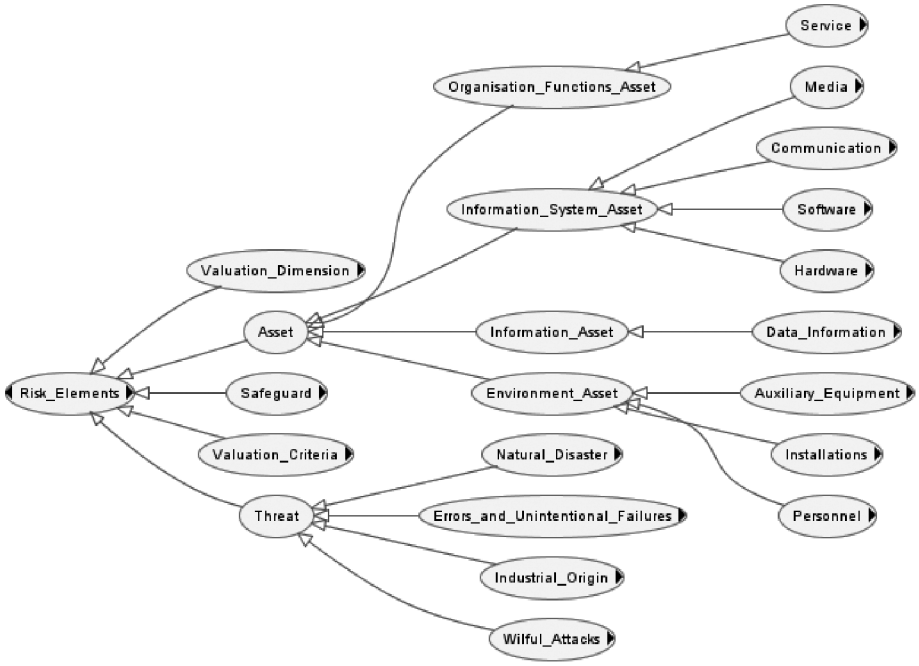


Figure 1: Taxonomy of the elements of the risk analysis ontology

- **Valuation criteria:** The criteria which made an asset valuable for an organization, in other words, how interesting the asset is for the system. We must protect the most valuable asset. It includes a numerical value (1-10) of their importance, and a rationale for this (see MAGERIT, 2006 for details).

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 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. For example, the *Errors and Unintentional Failures (threat)* affect the *Software (asset)* but not the *Personnel (asset)*, and *Traffic Analysis (threat)* only affects *Communication (asset)*. Moreover, other constraints such as disjointedness or cardinality can also be defined in OWL and have been very useful for the construction of this ontology (and the following ones, described in the next sections). An example of these constraints, captured on the OWL Protégé Editor Program, are shown in Figure 2.

In Figure 2 the class *Natural Disaster* (which represents a *threat*) is selected in the taxonomy shown in the left of the image. We can see in the bottom of Figure 2 that the concept *Natural Disaster* is disjoint with the threats *Industrial Origin*, *Willful Attacks* and *Errors and Unintentional Failures*. Besides, there exist two binary relations "has_asset" that represents which *threat* can affect which *asset* and "has_valuation_dimension" that represents the possible *valuation dimensions* in which the *assets* can be affected by this type of *threat*. A set of constraints has been introduced in these relations over the hierarchy. These constraints are shown in the "Properties and Restriction" square in Figure 2. There exist two types of constraints: inherited or specific. In this case, the cardinality constraints of both relations are inherited constraints from the class *Threat* and

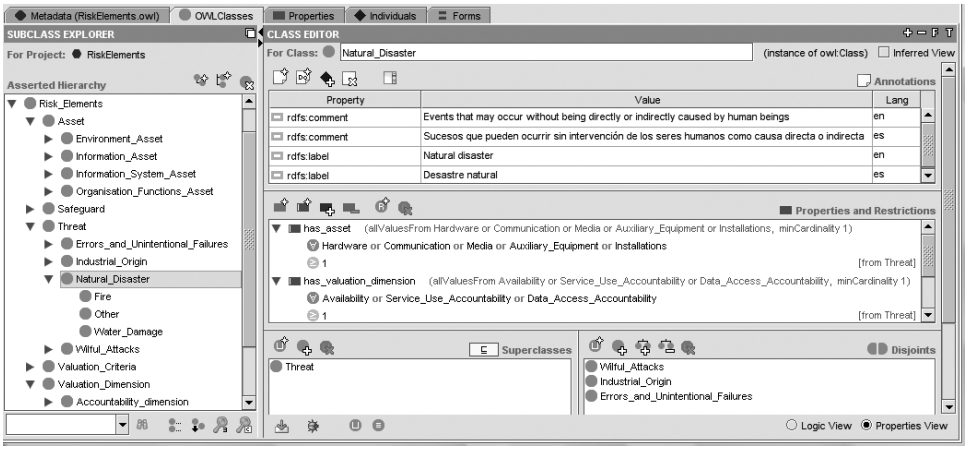


Figure 2: Taxonomy and constraints of the elements of the risk analysis ontology in the Protégé Editor

represents the fact that each *threat* has to affect at least one *asset*, and over at least one *valuation dimension*. On the other hand, the relation "has_asset" is a specific constraint representing the fact that *Natural Disaster threat* can only affect *Hardware*, *Communication*, *Media*, *Auxiliary Equipment* or *Installations assets*. In the relation "has_valuation_dimension", the valuation dimension on which a *Natural disaster threat* can act are only *Availability* or *Accountability* (*Service use* or *Data Access*). Moreover, as we can see in Figure 2, each concept of the ontologies has been described in English and Spanish using the "rdfs:label" and "rdfs:comment" *annotations property*.

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 and semantic web query languages (such as SPARQL – see Section 3.3 "Using Semantic Queries for Verifying the Correctness of the Requirements").

2.2 Requirements Ontology

The concepts, metainformation and relationships included in the requirements ontology have been mostly taken from our experience of requirements reuse, specifically of the reuse-based RE method called SIREN (Toval, Nicolás, Moros and García, 2002a; Toval *et al*, 2002b). 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 (IEEE, 1999a; IEEE, 1999b). To date, these requirements have been represented textually (organized in the IEEE documents for software requirements (IEEE, 1999a) and system requirements (IEEE, 1999b)), with their metainformation associated to them. Figure 3 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*"}

Figure 3: Requirements Taxonomy (an extract)

- **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 be Rule out, Approved, Modelled in Analysis, Modelled in Design, Implemented or Verify*}
- **Traceability:** in a requirements document the requirements can appear as 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*), with both being transitive.
  - **Parent-Child:** these are relationships through which the children requirements refine the meaning of the parent ones. This relation has been modeled like 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.
- **Source:** the source of the requirement. Although the client needs are the main source, other 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 section of the document in which the requirement is situated, if the requirement is stored in some document. *Datatype property*.
- **ValidatedBy:** the validation criteria needed to test the requirements. *Datatype property*.
- **ProposedBy:** the stakeholder – people involved in the project development – who requests that the requirement is included. *Datatype property*.
- **Responsible for:** the stakeholder who must satisfy or perform the requirement. *Datatype property*.
- **Parameterized value:** parameterized requirements contain some parts that have to be adapted to each application or system and that have to be instantiated at reuse time. This kind of requirement encourages reuse by factoring out system-specific details as parameters of the requirement, thus enabling engineers to specify requirements with a higher degree of flexibility. *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 (2005): "*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 (Figure 3) described in Section 2.2 above has been used to classify the security requirements, 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 for identifying this new metainformation has been MAGERIT (Section 2.1), other instructions from the family of security standards ISO 27000, specifically by the ISO 27002 (ISO27002, 2005), 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 below:

- **has_asset:** every security requirement has to be related to one asset. Thus, an *Object property* has been added to the concept security requirement to represent its associated asset. The range of the *Object property* is the class *Asset* defined in the risk analysis ontology. This *Object property* is inherited and restricted throughout the hierarchy (through a set of constraints which has been described in the Protégé editor as in Section 2.1). For example a *Physical System Requirement* can only be associated to *Hardware* and *Installations assets*, while a *Software External Communication Interface Requirement* can only be associated to the *Communication asset*. Furthermore storing the *owner*, the *person and the unit responsible for* the asset are relevant. This information is an objective in ISO 27002 (2005) (*Section 7 – Asset Management*).
- **has_threats:** it represents possible threats associated to the non-fulfilment of the requirements. This property is represented by an *Object property* over the hierarchy of threats of the risk analysis ontology (Section 2.1) and so its range is the class *Threat* defined in this ontology. The

risk analysis ontology has constraints of which threat can be occurred to which asset, so in the security requirements ontology we 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* or *Installations*) to the threat *Repudiation*, associated to the asset of *Service*. 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 (MAGERIT, 2006). Furthermore, the *Datatype property* "*rationale_of_valuation_criteria*" includes the rationale for selecting this value.
- **has_valuation_dimensions:** the features that make an asset (in this case, the asset associated to the requirement) valuable. There exist five valuation dimensions modelled using an *Object property*: "*Availability*", "*Integrity*", "*Confidentiality*", "*Accountability*" and "*Authenticity*".
- **has_safeguards:** This *Object property* associates the related requirement to the safeguard. 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 security standards or current legislation a requirement has been derived from. This is useful for the application of the framework shown in the Section 3. The current version is available at http://dis.um.es/~jolave/securityRequirements.owl.

2.4 Consistency of the Ontologies

The ontologies have been designed and implemented by using OWL, which is the current recommendation of the W3C for the exchange of semantic content on the web. Among the different OWL flavors, OWL-DL, based on Description Logics, has been used since it has the expressivity needed and allows for complete reasoning. Its formal model allows for performing automatically a series of operations, which can be supported by DL reasoners (e.g., Pellet, Fact++, Racer). One of such operations is to check the internal consistency of the ontology. An OWL ontology can be viewed from a logical point of view as a collection of axioms that must be satisfied. This does not only include classes and properties, but also restrictions such as disjoint classes. The consistency of our ontology has then been internally validated by using Pellet reasoner (http://pellet.owldl.com/). This ensures the correct results of the knowledge that can be inferred from the ontology by applying the corresponding axioms. Moreover, the existence of such restrictions is not only useful for building the model, but also in guaranteeing the consistency of the individuals built, which must satisfy the restrictions defined for their corresponding class. Moreover, the collection of conditions defined for the classes can be used by the reasoner for the automatic classification of individuals.

On the other hand, the ontologies have been developed according to a formal method to build and compare ontologies (Lozano-Tello and Gómez-Pérez, 2004). We have followed their recommendations about how the concepts, relationships, taxonomy and axioms should be identified and described, as we consider in our previous work (Blanco, Lasheras, Valencia-García, Fernández-Medina, Toval and Piattini, 2008). Table 1 shows some current information details about the ontologies.

3. 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 (MAGERIT, 2006). There are numerous sources

	Risk analysis ontology	Security Requirements ontology
Classes	309	56
Individuals	307	350
Number of datatype properties	13	18
Number of object properties	7	11
Restrictions	166	21
Disjoint Axioms	30	15

Table 1: Ontology Information Detailed

to identify these safeguards, including international standards – such as ISO 27002 (ISO27002, 2005) or ISO 15408 (ISO15408, 2005) –, national standards or regulations – such as data protection laws –, and sector standards.

In this 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 by quantifying the threats (MAGERIT, 2006). This is achieved thanks to all the properties, relationships and semantic properties, of the ontologies, enabling requirements to be elicited and specified, for instance, by assets or by probable threats to the system.

Our framework also 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" – see Section 2.3 – must be considered as a layer to search and identify requirements.

The advantages of protection by catalogue are speed, need for little effort (once the catalogue has been developed), and standardization, providing uniform results related to different projects within the organization and/or with other similar organisations. An additional advantage of using this framework is the possibility of identifying the measurement of percentage of satisfied requirements. In Martínez, Lasheras, Toval and Piattini (2006), 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. In this work, the percentage of the satisfied requirements from the catalog was presented as a measurement of security.

3.1 Application to a Reusable Requirements Repository

In a previous publication (Toval *et al*, 2002a), a reusable requirements catalog related to security with about 350 requirements (with their metainformation and traceability relationships) was defined. This previous catalog was validated in a real case in the Spanish Public Administration. 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 population process of the requirements ontology, 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 were detected: 8 were related to the traceability of the requirements, and 19 were identified by constraint of risk analysis. Some examples of these inconsistencies are:

- The requirement with text: "*Access to the installations must be controlled by a physical device which uses passwords*" was associated to a *Service asset*, but (as we have described in Section 2.3) a *Physical System Requirement* can only be associated to *Hardware* and *Installation assets*.
- Other inconsistency was the requirement with text: "*Any connection to a system with confidential information must be temporarily limited to a period not exceeding [Time in minutes]*". It was associated to the *Data Asset*, but it was an inconsistency, because (as we have described in Section 2.3) a *Software External Communication Interface Requirement* only can be associated to the *Communication asset*. It is worth noting that this requirement has a *parameterized value* (see Section 2.2), in this case a numerical value (time in minutes), that must be instantiated at reuse time.

3.2 Example of Security Requirement

Let us suppose that the requirement to model is a high-level one (close to the stakeholders and the problem-realm), and that 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* (Figure 3). Each property of this new individual has to be filled in (Figure 4). The system allows 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 (*Installations*), the user cannot relate any threat of the hierarchy of threats (Figure 1) identified as *Wilful Attacks* (deliberate failures caused by people) or *Error and Unintentional Failures* (unintentional failures caused by people) to it.

The source attribute allows this requirement to be identified as associated to the ISO 27002 control objective – 9.1.4 Protecting against external and environmental threats. However, this requirement has no direct correspondence on requirements extracted by Data Protection 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.

3.3 Using Semantic Queries for Verifying the Correctness of the Requirements

Let us suppose that the requirement to model is:

"Confidential information must remain, at all times, outside the scope of employees and people who have no need of knowing information by business reasons".

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.*

Figure 4: Properties of the requirement, an extract

It is an *Information Management System Requirement*, specifically a *Confidential Information Requirement* (Figure 3). Through the constraints of the ontologies we know that this kind of requirement only can be related to *Data*, *Services* or *Personnel assets*. In the reusable textual requirements catalogs this requirement was associated to a Data asset, so the constraints were fulfilled. Moreover the requirement, affects this asset in its *Confidentiality dimension*.

On the other hand, the requirements has associated the threat of the *Errors and Unintentional Failures – E.14 Information Leaks* (The information accidentally reaches people who should not have knowledge of it, without the information itself being altered), which has restricted the possible asset associated to *Communication*, *Software* and *Data*, and affects only the *Confidentiality dimension*.

In this case, the asset defined for the requirement is consistent with the threat that affects it. However this check could not be performed in Protégé with OWL, so we have to do some semantic queries to verify. We have used the semantic web query language SPARQL (http://www.w3.org/TR/rdf-sparql-query/) by means of two queries:

Query#1: to select the requirements and assets in the sense that all the assets associated to every requirement have to be the same asset on which the threat of the requirement acts.

```
SELECT distinct ?x ?asset
WHERE{ ?x :has_asset ?asset ; :has_threat ?threat.
        ?threat magerit:has_asset ?b. filter(sameTerm(?asset,?b) )
} orderby ?x ?asset
```

Query#2: to check that the asset associated to the requirement can be found in the assets that are affected by the threat. So, the next query selects all the requirements and their assets.

```
SELECT distinct ?x ?asset
WHERE{ ?x :has_asset ?asset ;
} orderby ?x ?asset
```

Finally we will have to implement the intersection of the two queries to check that the result is the same that the asset associated to the requirement.

4. RELATED WORK

Several authors consider the definition of a security ontology an important area of research and a challenge within the community of security engineering (Tsoumas and Gritzalis, 2006; Mouratidis and Giorgini, 2007a). In our previous studies, a systematic review and comparison (based on the formal method (Lozano-Tello and Gómez-Pérez, 2004)) of Security Ontologies was made (Blanco *et al*, 2008), which has been the basis for identifying the contribution of this work.

In security, ontologies concerning dependability domain (Dobson and Sawyer, 2006), trust domain (Mouratidis, Giorgini and Manson, 2003) and other functional security issues – algorithms Security or Policy security – (Kim, Luo and Kang, 2005) have been used to provide a unified conceptualization, but none of them are associated to risk analysis. Tsoumas and Gritzalis (2006) present an ontology of risk analysis based on standards used in a security management framework for information systems, and in Fenz and Weippl (2006) it is focused on small and medium companies. Nevertheless, none of them has integrated it with RE techniques in order to deal with their benefits and the use of reuse requirements catalogs. Lee and Gandhi (2005) proposes an ontology-based active RE framework, although it is particularly adapted to the Department of Defense, Information Technology Security Certification and Accreditation Process (DITSCAP). Moreover, it is not based on risk analysis and catalogs of requirements are not used.

On the other hand, the increasing importance of security in organizations has given rise to much research that focuses on the inclusion of security concerns in the system development process, specifically analysis and modelling security requirements. These proposals tackle the problem of designing security requirement from different perspectives. Some of them define the undesirable behaviours that a system should prevent (McDermott and Fox, 1999; Sindre and Opdahl, 2005) by means of abuse cases and misuse cases (respectively); there are also proposals that focus on security related features such as confidentiality and access control (Basin, Doser and Lodderstedt, 2006; Jürjens, 2005), based on MDA (Model-Driven Approach) and the use of UML profiles – UMLSec – (respectively); and other approaches that focus on security trust requirements following a goal-oriented framework (Mouratidis and Giorgini, 2007b). However, none of them focus on textual requirements and risk analysis. Anyway our framework must consider them in order to extend the framework as further work. In this sense, it is worth noting the work (Jürjens and Houmb, 2004) which applies a Risk-Driven Development based on a UML profile (UMLSec) but to Security-Critical System – so dependable requirements –. In this case, we identified, as further work, the construction of a link to that approach through the UMLsec notation, in order to allow our RE based approach to be linked to the design analysis phase.

Consequently, although several related works have been identified, none of them combines the use of an RE method, reuse, ontologies and pre-existing catalogs, as the basis of a "lightweight" method to elicit security requirements in line with security standards like the ISO 27002 (ISO27002, 2005), with the aim that the results obtained could be adapted and certified in under these standards. Besides, none of them have followed a formal method in the development of the ontologies.

5. CONCLUSIONS AND FURTHER WORK

This contribution proposes an ontological representation for reusable requirements, which allows *incompleteness* and *inconsistency* in requirements to be detected and semantic processing in requirements analysis to be achieved without rigorous NLP (*Natural Language Processing*) techniques. The ontologies have been developed according to a formal method for building and comparing ontologies (Lozano-Tello and Gómez-Pérez, 2004) and implemented in a standard language, OWL. Moreover, their definition is based on requirements engineering standards (IEEE, 1999a; IEEE, 1999b) and security standards and regulations (MAGERIT, 2006; ISO27002, 2005). 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 and specify security requirements which permits us to reduce significantly the impact of risks without making large investments 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 3). A measurement of security is also offered by identifying the percentage of satisfied requirements with respect to standards or regulation (see Section 3).

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. We have made the ontologies available (http://dis.um.es/~jolave/RiskElements.owl and http://dis.um.es/~jolave/securityRequirements.owl). Work in progress is focused on this issue, considering the ontologies identified (Blanco *et al*, 2008) and the related work in Section 4, such as is the case of integrating our work with the UMLSec framework (Jürjens, 2005). In this way, we are also planning to extend the risk analysis top level ontology using other widely-accepted standards, such as the CCTA Risk Analysis and Management

Method (CRAMM) or OCTAVE (*Operationally Critical Threat, Asset and Vulnerability Evaluation* by the Carnegie-Mellon). On the other hand, as further work, our ontologies might be used to perform new semantic restrictions by using Semantic Web Rules (SWRL), and we expect to improve the selection of the requirements with the use of expert systems to prioritize the requirements. This framework is currently being evaluated by several Spanish companies. So far, we are getting positive feedback for its usefulness.

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

BASIN, D., DOSER, J. and LODDERSTEDT, T. (2006): Model driven security: From UML models to access control infrastructures. *ACM Trans. Softw. Eng. Methodol.*, 15(1): 39-91.

BERNERS-LEE, T., HENDLER, J. and LASSILA, O. (2001): The semantic web. *Scientific American*.

BLANCO, C., LASHERAS, J., VALENCIA-GARCÍA, R., FERNÁNDEZ-MEDINA, E., TOVAL, A. and PIATTINI, M. (2008): A systematic review and comparison of security ontologies. *International Conference on Availability, Reliability and Security (ARES)*. Barcelona, IEEE Computer Society 813-820.

BREWSTER, C., O'HARA, K., FULLER, S., WILKS, Y., FRANCONI, E., MUSEN, M.A., ELLMAN, J. and BUCKINGHAM, S. (2004): Knowledge representation with ontologies: The present and future. *IEEE Intelligent Systems*, 19(1): 72-81.

CYBULSKY, J.L. and REED, K. (2000): Requirements classification and reuse: Crossing domains boundaries. *6th International Conference on Software Reuse (ICSR'2000)*. Viena, Springer, LNCS. 190-210.

CHENG, B.H.C. and ATLEE, J.M. (2007): Research directions in requirements engineering. *Future of Software Engineering 2007 (FOSE), in ICSE*. Minneapolis, Minnesota, IEEE Computer Society 285-303.

DEVANBU, P. and STUBBLEBINE, S. (2000): Software engineering for security: a roadmap. *ICSE, Future of Software Engineering*. Limerick, Ireland. 227-239.

DOBSON, G. and SAWYER, P. (2006): 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.

FENZ, S. and WEIPPL, E. (2006): Ontology based IT-security planning. *12th Pacific Rim International Symposium on Dependable Computing PRDC '06*. IEEE Computer Society. 389-390.

FIRESMITH, D. (2004): Specifying reusable security requirements. *Journal of Object Technology*, 3(1): 61-75.

GLASS, R.L. (2002): *Software Engineering: Facts and Fallacies*, Addison-Wesley.

GRUBER, T. (1995): Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5/6): 907-928.

IEEE (1999a): Std 830-1998 *Guide to Software Requirements Specifications Volume 4: Resource and Technique Standards*. IEEE Software Engineering Standards Collection.

IEEE (1999b): Std 1233-1998 *Guide for Developing System Requirements Specifications. Volume 1: Customer and Terminology Standards*. IEEE Software Engineering Standards Collection.

ISO15408 (2005): ISO/IEC 15408 (Common Criteria) Information technology security techniques-evaluation criteria for IT Security.

ISO27002 (2005): ISO/IEC 17799-27002 Code of Practice for Information Security Management.

JÜRJENS, J. (2005): *Secure Systems Development with UML*, Springer-Verlag.

JÜRJENS, J. and HOUMB, S.H. (2004): Risk-driven development of security-critical systems using UMLsec. *IFIP Congress Tutorials*. 21-54.

KIM, A., LUO, J. and KANG, M. (2005): Security ontology for annotating resources. *4th International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE'05)*. Agia Napa, Cyprus. 1483-1499.

LEE, S.W. and GANDHI, R.A. (2005): Ontology-based active requirements engineering framework. *APSEC*. 481-490.

LOZANO-TELLO, A. and GÓMEZ-PÉREZ, A. (2004): 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): 1-18.

MAGERIT (2006): Methodology for information systems risk analysis and management. http://www.csi.map.es/csi/pg5m20.htm.

MARTÍNEZ, M.A., LASHERAS, J., TOVAL, A. and PIATTINI, M. (2006): An audit method of personal data based on requirements engineering. *The 4th International Workshop on Security In IS (WOSIS-2006)*. Paphos, Chipre. 217-231.
McDERMOTT, J. and FOX, C. (1999): Using abuse case models for security requirements analysis. *Annual Computer Security Applications Conference (ACSAC)*. Phoenix, Arizona. 55-66.
MOURATIDIS, H. and GIORGINI, P. (2007a): *Integrating Security and Software Engineering: Advances and Future Visions*, Idea Group Publishing.
MOURATIDIS, H. and GIORGINI, P. (2007b): Secure tropos: a security-oriented extension of the tropos methodology. *International Journal of Software Engineering and Knowledge Engineering*, 17(2): 285-309.
MOURATIDIS, H., GIORGINI, P. and MANSON, G. (2003): An ontology for modelling security: The Tropos approach. *Knowledge-Based Intelligent Information and Engineering Systems*. Springer Berlin / Heidelberg.
PRIETO-DÍAZ, R. (1993): Status Report: Software Reusability. *IEEE Software*, 10(3): 61-66.
RASKIN, V., HEMPELMANN, C.F., TRIEZENBERG, K.E. and NIRENBURG, S. (2001): Ontology in information security: A useful theoretical foundation and methodological tool. *New Paradigms Security Workshop NSPW'01*. ACM Press Clouford, New Mexico, USA. 53-59.
RINE, D.C. and NADA, N. (2000): An empirical study of a software reuse reference model. *Inf. and Software Technology*, 42(1): 47-65.
ROTHENBERGER, M.A., DOOLEY, K.J., KULKARNI, U.R. and NADA, N. (2003): Strategies for software reuse: A principal component analysis of reuse practices. *IEEE Trans. on Soft. Eng.*, 29(9): 825-837.
SINDRE, G. and OPDAHL, A.L. (2005): Eliciting security requirements with misuse cases. *Requirements Engineering*, 10(1): 34-44.
SMITH, S.W. and SPAFFORD, E.H. (2004): Grand challenges in information security: Process and output. *IEEE Security & Privacy*, 2(1): 69-71.
SOMMERVILLE, I. (2006): *Software Engineering (8th edition)*, Addison-Wesley.
TOVAL, A., NICOLÁS, J., MOROS, B. and GARCÍA, F. (2002a): Requirements reuse for improving information systems security: A practicioner's approach. *Requirements Engineering*, 6(4): 205-219.
TOVAL, A., OLMOS, A. and PIATTINI, M. (2002b): Legal requirements reuse: A critical success factor for requirements quality and personal data protection. *Int. Conference on Requirements Engineering (RE)*. Essen, Alemania. 95-103.
TSOUMAS, B. and GRITZALIS, D. (2006): Towards an ontology-based security management. *20th International Conference on Advanced Information Networking and Applications (AINA'06)*. Vienna, Austria, IEEE Computer Society. 985-992.

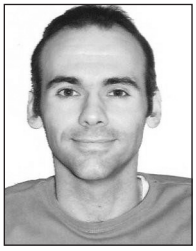
BIOGRAPHICAL NOTES

Joaquín Lasheras Velasco, 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.

Joaquín Lasheras Velasco

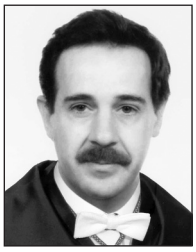
Dr Rafael Valencia-García received his BA, MSc and PhD degrees in Computer Science from the University of Murcia. He is a lecturer at the Department of Informatics and Systems, University of Murcia. His main research interests are Natural Language Processing and the application of Knowledge Technologies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

Rafael Valencia-García



Jesualdo Tomás Fernández-Breis

Dr Jesualdo Tomás Fernández-Breis received his BA, MSc and PhD degrees in computer science from the University of Murcia. He is an associate professor at the Department of Informatics and Systems, University of Murcia. His research interests include the development and application of knowledge technologies to different fields such as Image Processing, Medicine, the Semantic Web, E-learning and Bioinformatics. He is currently the prime investigator of three national projects and has published over 40 articles in journals, conferences and book chapters. He is the author or coauthor of several books.



Ambrosio Toval Álvarez

Dr Ambrosio Toval Álvarez is a full professor at the University of Murcia, in Spain. He holds a BS 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 requirements engineering tools, and security requirements. Dr Toval is currently the Head of the Software Engineering Research Group at the University of Murcia.