Information & Computer Security

Mapping information security standard ISO 27002 to an ontological structure

Stefan Fenz Stefanie Plieschnegger Heidi Hobel

Mapping Information Security Standard ISO 27002 to an Ontological Structure

Abstract

Since information is becoming more valuable and today's businesses face frequent attacks on their infrastructure, enterprises need support at protecting their information based assets. Information security standards and guidelines provide baseline knowledge for protecting corporate assets. However, the efforts to check if the implemented measures of an organization adhere to the proposed standards and guidelines are still significantly high. This paper shows how the process of compliance checking can be supported by using machine-readable ISO 27002 control descriptions in combination with a formal representation of the organization's assets. We created a formal representation of the ISO 27002 standard and showed how a security ontology can be used to increase the efficiency of the compliance checking process.

Keywords:
ontology design, security, compliance management, risk management

1. Introduction

Nowadays information is one of the most important and valuable goods for organizations (e.g., usage of customer data in advertisements, tactical and strategical decision making). In 2012, a study by Symantec provides evidence for the phenomenon of information-based value. According to Symantec information is estimated to make 49% of an organization's total value [1]. On the other hand, today's businesses face a rising number of information security threats and vulnerabilities as a result of increased networking, interconnection, and electronic processing. Considering the possible multifaceted attack scenarios, it requires a significant effort to find and implement appropriate countermeasures and mitigation strategies in order to protect the valuable assets of a company.

Current information security standards and guidelines recommend appropriate security measures and discuss vulnerabilities and threats related to the information assets of an organization. However, especially small- and medium-sized organizations still underestimate the risk of data loss, corruption, and in case that the news report an incident, the overall impact on the reputation of the organization. The Internet security threat report 2013 [2] revealed that 31% of attacks in 2012 were aimed at businesses with less than 250 employees. Moreover, threats are not limited to attacks over the Internet. Information is a generic term and it includes various different forms like print outs, electronically stored and processed information, or even verbal communications. Thus, it must be ensured that information is appropriately protected to prevent unintended disclosure or loss, and also physical threats such as fire or vandalism must be considered [3].

Information security standards and guidelines serve as a baseline to assess and improve the security measures of a company. However, these standards incorporate several informal rules that are interwoven and not always unambiguous. Furthermore, these standards are steadily changed and improved, evolving with the rising number of threats. The ISO 27002 standard [3] is meant to provide concrete implementation guidelines for security measures. However, auditing and improving the security of an infrastructure entail a significant effort, since versatile mitigation strategies can be used, and when one strategy or implementation is altered, this could directly or indirectly affect the rating of other security measures.

This paper aims at solving this shortcoming by proposing a knowledge base, which supports organizations at the compliance checking of their implemented security measures with ISO 27002 controls [3]. Based on previous work [4, 5], we implemented a knowledge base that comprises the previously defined security ontology and enhanced it with the controls of the ISO 27002 standard. We illustrate how this knowledge base can be used for compliance checking and evaluated it based on concrete examples to illustrate the applicability of our approach. The overall work enables companies to stay competitive in securing their sensitive information. In summary, the main contributions of this paper are:

- We developed a methodology which enables the mapping of informal security guidelines into formal control implementation descriptions (modeled within an existing ontology).
- We analyzed the ISO 27002 standard controls and formalized them in formal control descriptions/rules that serve as a knowledge base for automated compliance checking. In total 118 formal implementation rules have been created within the ontology.
- Based on the controls we extended our previously defined security ontology by mapping appropriate countermeasures into the ontological structure (e.g., different types of data backup policies for implementing the data backup policy control).
- We evaluated and illustrated our approach based on specific use cases.

The paper is structured as follows: Section 2 outlines previous and related work. In Section 3 the fundamentals of the security knowledge base are explained. Sections 4-5 describe the ISO 27002 mapping process and the way in which the technical evaluation was conducted (including examples of the compliance checking process). Challenges that have been encountered during the mapping process are part of the discussion in Section 6. In Section 7, we conclude our work.

2. Related Work

Souag et al. [6] conducted a systematic mapping study regarding reusable knowledge in security requirements engineering. The authors developed a comparison framework for methods, techniques, modeling frameworks and tools for reuse in security requirements engineering and identified different forms of knowledge representation and reuse in the security domain. Five main types of security knowledge representation were identified: (i) security patterns, (ii) taxonomies and ontologies, (iii) templates and profiles, (iv) catalogs and generic models, and (v) mixed forms. With regard to security ontologies we identified the following approaches:

Raskin et al. [7] introduced an ontological approach to information security, which concentrates on two issues: (i) the inclusion of natural language data sources in information security applications, and (ii) a formal specification of the information security knowledge. While the authors presented an extensive list of security relevant terms gained from natural language sources, they did not present any ontology to embed and interrelate these terms in a systematic manner, though.

Schumacher et al. [8] introduced a high-level ontology about information security, including the concepts asset, stakeholder, security objective, threat, attack, attacker, vulnerability, countermeasure, and risk. The ontology enables only a limited representation of high-level information security knowledge due to the following reasons: (i) vulnerabilities are bound to assets, which makes it impossible to model an organizational weakness (e.g. no clean desk policy) as a vulnerability, (ii) threats harm assets, excluding the potential danger for human beings, (iii) while the concept attack realizes the concept threat, it is also possible that a threat is triggered by human errors or deliberate acts; the concepts attack and threat represent similar and not exclusive concepts, and (iv) the ontology is only presented at a very high-level perspective, while more granular definitions or concrete instances are missing.

Avizienis et al. [9] proposed the basic concepts and taxonomy of secure and dependable computing by defining dependability and security as the most generic concepts. Attributes such as reliability, maintainability, safety, integrity, or availability are specificities of the dependability concept, while confidentiality, integrity, and availability are included by the security concept.

Kim et al. [10] refined the security ontology approach by Denker et al. [11]. Security information such as mechanisms, protocols, objectives, algorithms, and credentials were described using ontologies. The ontologies were applied to a Service Oriented Architecture to annotate security aspects of web service descriptions and queries respectively. The authors divided the entire system into seven ontologies: (i) the *main security ontology* describes security concepts such as security policies or security protocols, (ii) authentication credentials are specified using a *credentials ontology*, (iii) security algorithms such as encryption algorithms, checksum algorithms, or signature algorithms are specified using a *security algorithms ontology*, (iv) the *security assurance ontology* can be used to classify different assurance standards such as FIPS and NSA standards, (v) security annotations of semantic web services are facilitated by the *service security ontology*, (vi) the *agent security ontology* enables querying security information such as security requirements or security capabilities of web services, (vii) the security of input and output parameters of web services are specified using an *information object ontology*. While the security ontologies by [10] define certain areas of the information security domain (e.g. credentials or encryption algorithms) in a highly granular way, other important areas such as vulnerabilities, assets, threats, or controls are completely missing.

Martimiano et al. [12] introduced an OWL (Web Ontology Language)-based security incident ontology, which defines a security-related vocabulary of terms and relations. On the one hand, the ontology should support the sharing of a common understanding and, on the other hand, it should support the reuse of the security domain knowledge. System administrators using security tools are seen as the target group. The main idea is that an *agent* performs an *attack* that can cause a *security incident*. To perform an *attack*, an *agent* uses a *tool* which explores a *vulnerability* to get *access*. A *security incident* causes a *consequence*, impacts an *asset*, and happens at a specified *time* [12]. While the ontology by Martimiano and Moreira considers several parts of an organization's IT-infrastructure as individual assets, it does not allow the mapping of the entire organization structure. Furthermore, it allows the definition of asset vulnerabilities but it does not provide the possibility to make statements about threats which could affect the organization and its security goals.

Karyda et al. [13] concentrated on using a security ontology to support software developers in developing secure applications. An ontology is used to capture security domain knowledge from experts on sensitive application domains like electronic governmental services. With such an ontology in place, a developer can query the knowledge base and thereby gain security-related insights to make informed choices on security solutions and mechanisms. Competency questions guided the development of the security ontology which resulted in the following core elements: *Asset*, *Countermeasure*, *Objective*, *Person*, and *Threat*. As for an e-tax and an e-vote application development project, likely upcoming questions are presented in a description logic query language for retrieving individuals and consequently results from the ontology are presented. As the authors mention, filling this ontology with expert knowledge is a laborious process. Furthermore, the modeled knowledge only covers the surface by including high-level concepts. For example the query for countermeasures that protect the personal data of a tax-paying citizen returns *Encryption*, *Access Control*, *Certificates*, etc. A formal model of the highly complex following decision processes, security-related connections, and details on the collection and maintenance of the security related knowledge is not provided.

Herzog et al. [14] proposed an OWL-based (W3C Web Ontology Language) ontology which models assets, threats, vulnerabilities, and countermeasures as well as their interrelations. One shortcoming of the security ontology by [14] is the asset concept which is subdivided into countermeasure, credential, technology, and human. While credentials are described in much detail, technological assets only describe the IT-infrastructure like hardware, networks, or data, while the role of humans is reduced to the receiver or sender of messages. The ontology does not describe any other infrastructural facilities like buildings or systems for electric power supply and reduces the role of humans to an absolute minimum. The threats described mainly concentrate on risks arising from virtual attacks on information systems. The ontology offers detailed information about these threats but all kinds of physical threats like fire or the simple theft of devices are only mentioned marginally. Consequently the countermeasures offered by the ontology only pay attention to the threats mapped in-depth. Furthermore, the suggested countermeasures do not refer to standard controls offered by information security standards. The security ontology by Herzog et al. has a very technical focus and is hence appropriate to ensure security in the sector of authentication and access to information systems but not for the purposes of offering a holistic security approach for an organization's IT-infrastructure as a whole.

One project related to the Common Criteria (CC) for Information Technology Security Evaluation indicated the requirement for an automated certification process in order to make the certification easier and faster. Therefore, a CC ontology has been presented [15] that allows browsing the knowledge represented in the ontology with visualization tools that have been specially developed for RDF (Resource Description Framework) [16] or OWL [17] browsing. Additionally the data structure can be queried using SPARQL (SPARQL Protocol and RDF Query Language) [18]. To support the evaluation process, a CC certification support tool has been developed. The security ontology used as starting point for the underlying formalization has been described in [4].

Fenz et al. [19] describes an approach of mapping the ISO/IEC 27001 [3] standard into the security ontology in addition with a framework that allows to access, visualize, and reason on ontological data. An approach for mapping knowledge of the French EBIOS [20] standard and the German IT Grundschutz Manual [21] into the security ontology is explained in [5]. Another related project is discussed in [22] and deals with the evaluation of compliance with information security standards as well as with the evaluation of the respective effectiveness. The security ontology is a knowledge base created to formalize information security knowledge using the Web Ontology Language (OWL) [17]. It has been introduced by Fenz et al. in order to map security standards, guidelines and best practices [4]. As such the security ontology by Fenz et al. can be used to model a formal representation of the ISO 27002 controls directly in the ontology.

Furthermore, the ontology can also be used to automatically determine the implementation level of each control. Please see the following section for a detailed description of the ontology.

3. Fundamentals of the Security Knowledge Base

In this section, we provide a comprehensive overview of our complete approach, the fundamental concepts such as the security ontology, and the methodology used to map the ISO 27002 controls into the extended security ontology.

3.1. Overview

The proposed approach for creating the fundamental knowledge base is illustrated in Figure 1. It is based on the security ontology described in [4] and on previous work on mapping the ISO 27001 standard on the security ontology [19]:

- The developed mapping methodology shown on the top side of the figure is used to transform the informal ISO 27002 knowledge into a formal and machine-readable form. First, the ontology in which the knowledge will be mapped is analyzed regarding already existing concepts and relations. Second, the ISO 27002 standard is analyzed regarding domain-relevant concepts and relations. Based on the results of Step 1 and 2, the main concepts and relations are mapped to the ontology. In the last step the formal control implementation descriptions are created within the ontology based on the informal descriptions of the ISO 27002 standard.

- After the mapping, the knowledge base (ontology) consists of a formal representation of the ISO 27002 controls, their relations to the ISO 27001 control groups and objectives, and a formal representation of the control implementation rules.

- In the application phase we demonstrate how the research results can be used in a real-world setting. Supported by a tool, the organization models its assets within the ontological structure. By interpreting the formal control descriptions and the modeled assets, a reasoner (software which infers logical consequences from a set of asserted facts) classifies all assets which fulfill one or more of the ISO 27002 controls as compliant assets. A software tool visualizes the results and provides immediate feedback regarding the compliance level of the organization.

  - Example: The ISO 27002 standard states that there have to be data backup procedures in place. Within the security ontology this fact is modeled by a formal control description which requires that each organization which is modeled within the ontology has to be linked with some (at least one) instance of a data backup policy concept. In the application phase a reasoning software will process this formal control description and automatically determine if it is fulfilled or not. If the organization has a data backup policy in place it was modeled within the ontology in Step 1 of the application phase (see Figure 1) and the reasoner classifies the entire organization as compliant to this specific data backup control. In the following paper we describe how the reasoning process works in detail.

The usage of ontologies has many advantages over simple spreadsheets or relational databases. The main advantages are without doubt their interoperability and the possibility to analyze the stored facts by software-based semantic reasoners. Furthermore, due to their flexible structure, new entities and relations can be implemented without drawbacks, since the definitions can be implemented on-top and incrementally.

In this paper, we extended our previously introduced security ontology (see Section 3.2) and implemented ISO 27002 mappings with the relations and entities that are required to model the relations for the compliance control concepts as well as formalized the guidelines in form of formal control implementation rules in OWL DL (OWL sub-language). The overall purpose is to enrich the security ontology with formal rules derived from the ISO 27002 standard [3] to ease the compliance checking by enabling organizations to query, visualize and analyze the knowledge base. The proposed methodology in [5] is considered to keep the concepts reusable (see Figure 1). Moreover, the work is built upon the general instructions for identification of concepts and derivation of rules according to [4].

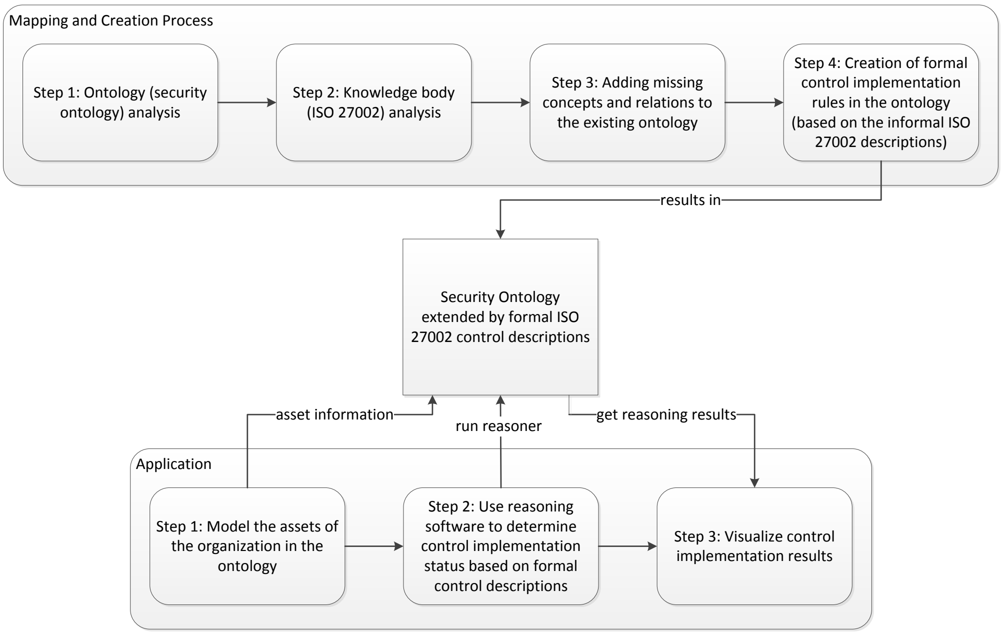

Mapping and Creation Process
Step 1: Ontology (security ontology) analysis
Step 2: Knowledge body (ISO 27002) analysis
Step 3: Adding missing concepts and relations to the existing ontology
Step 4: Creation of formal control implementation rules in the ontology (based on the informal ISO 27002 descriptions)
results in
Security Ontology extended by formal ISO 27002 control descriptions
asset information
run reasoner
get reasoning results
Application
Step 1: Model the assets of the organization in the ontology
Step 2: Use reasoning software to determine control implementation status based on formal control descriptions
Step 3: Visualize control implementation results

Figure 1: Overview of mapping ISO 27002 in the ontological structure and applying the results

For the compliance check, an organization's assets are mapped in the knowledge base and serve as snapshot of the organization's current security state that is considered in the reasoning process. The formalized controls and the snapshot are then evaluated by a reasoner, which infers the compliance status of the organization's implemented controls with the controls defined in the ISO 27002. Regardless of the inferred results, the implemented knowledge base and the inferred knowledge of the reasoner can by queried, visualized and analyzed. Due to the flexible ontological characteristics, extensions could be incrementally implemented and various mitigation strategies can be implemented in short time, leading to a sophisticated simulation system of security threats. Due to the huge effort to audit and analyze an organization's security measures, since one change in the security configuration can affect the compliance with various controls, the proposed framework is providing enterprises a more simple access to implement security concepts, through the possibility to model the whole infrastructure of an organization and find appropriate mitigation strategies based on a reasoner's findings.

3.2. Security ontology - Principles and Concepts

The security ontology, which was proposed in [4], is shown in Figure 2. The security ontology was developed to support especially small- and medium-sized organizations in the development of their security programs. Enterprises using the knowledge base could improve their IT security infrastructure without expensive audits and, moreover, could incrementally analyze the effects of the recently implemented security measures. Therefore, the security ontology can be used to model all *assets* of an organization. There are *threats* to each asset that can give rise to follow-up threats and impacts. For example, the threat Theft can lead to Impact Asset Loss. Threats and impacts have specific *security attributes*, e.g. confidentiality, integrity, availability. *Vulnerabilities* (can be physical, technical, or administrative weaknesses) can be exploited by threats and might damage assets. Countermeasures or mitigation strategies are implemented in form of *controls*. Each control has a *control type* (preventive, corrective, deterrent, recovery, or detective). These controls correspond to information security standards or best-practice controls.

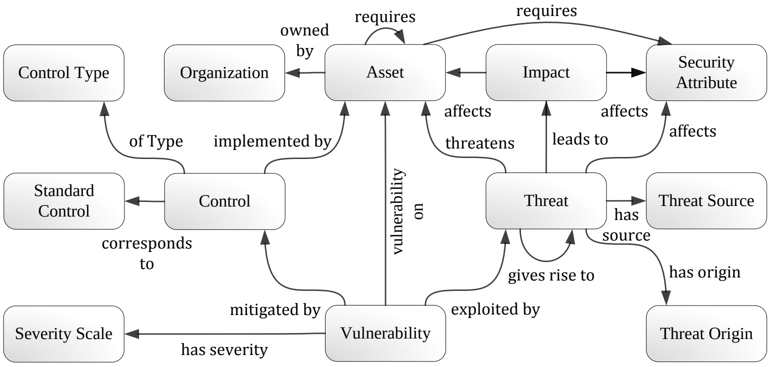


Figure 2: Concepts and Relations of the Security Ontology

The security ontology itself is a growing project and development carries on. The current version can be found at `http://sec.sba-research.org/webprotege/`. To make the developing process more efficient, the security ontology can be browsed and edited with Protégé OWL – an open source ontology editor [23]. In addition, reasoners are used to verify and evaluate the formal restrictions and axioms.

3.3. Methodology

The approach for mapping the ISO 27002 standard follows the best-practice guidelines suggested in [5]. It aims to make existing information security knowledge bases reusable and contains the following steps:

1. **Ontology analysis:** Analyzing existing concepts and relations of the selected ontology.

2. **Knowledge base analysis:** Identification of related entities and relations within the knowledge base that are similar to those already analyzed in the previous step.
3. **Mapping concepts and relations:** Mapping entities and relations according to the previous results.
4. **Mapping the knowledge:** Mapping of the actual knowledge.
5. **Evaluation**

The following sections discuss the analysis and mapping of the ISO 27002 controls to the security ontology.

4. Mapping the ISO 27002 into Ontological Structure

The ISO/IEC 27002 standard was prepared by Joint Technical Committee ISO/IEC JTC1, Information technology, Subcommittee SC 27, IT Security techniques. This international standard establishes guidelines and general principles for initiating, implementing, maintaining, and improving information security management in an organization. The objectives outlined in this international standard provide general guidance on the commonly accepted goals for information security management.

4.1. The Structure of ISO 27002

The ISO standard [3] contains 14 *security clauses* collectively containing a total of 35 *security categories*. Each category has a *control objective* and *controls* with a *control statement* to achieve the stated objective. This general information is also contained in the ISO 27001 [24]. Additionally the ISO 27002 standard includes an *implementation guidance* for each control that provides more detailed information and implementation suggestions to meet the control objective. In some cases, the implementation guidance is broken down into discrete steps. The control ends with the optional *other information* section that provides further information about some relevant considerations points or related topics, such as legal considerations or references to other standards or within the same standard. Table 1 shows an example of the implementation steps for an ISO 27002 control.

Table 1: Example ISO 27002 Implementation Guidance

security clause	11. Physical and Environmental Security
security category	11.1. Secure areas
control objective	To prevent unauthorized physical access, damage and interference to the organizations information and information processing facilities. . . .
control	Physical entry controls
control statement	Secure areas should be protected by appropriate entry controls to ensure that only authorized personnel are allowed access.
implementation guidance	f) access rights to secure areas should be regularly reviewed and updated, and revoked when necessary (see 9.2.4 and 9.2.5).
other information	-

The standard states that the order of clauses does not imply their importance. Each organization applying this standard should identify their right order of clauses – considering prior risk assessment analysis.

4.2. Notation

For an ontology ambiguous definitions are available. The original term originates from Greek and describes the study of being, existence or reality. In this paper we refer to the definition of [25]:

> ”... an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application.”

We use the standardized concepts from the World Wide Web Consortium (W3C) for ontology implementations: Resource Description Framework (RDF) [16] and Web Ontology Language (OWL) [17]. A selection of Web Ontology Language (OWL) class descriptions, used for the examples in this paper, is summarized in Table 2 and compared to Description Logic (DL) as well as First Order Logic (FOL) notation.

Table 2: Mapping of OWL Class Concepts to Description Logic (DL) and First Order Logic (FOL)

OWL class descriptions	**DL syntax**	**FOL short representation**
$owl{:}\,EquivalentClasses(C_1\ \dots\ C_n)$	$C_1 \equiv \dots \equiv C_n$	$C_i(x) \leftrightarrow C_j(x)\ for\ 1 \leq i < j \leq n$
$owl{:}\,SubClassOf(C_1\ \ C_2)$	$C_1 \sqsubseteq C_2$	$C_1(x) \rightarrow C_2(x)$
$owl{:}\,restriction(P\ owl{:}\,someValuesFrom(C))$	$\exists P.C$	$\exists y.P(x,y) \wedge C(y)$
$owl{:}\,restriction(P\ owl{:}\,hasValue(I))$	$\ni P.I$	$\exists y.P(x,y) \wedge I(y)$

In the following, we use the triple syntax of RDF (subject, property, object) to define relations of individuals, where a triple expresses a proposition or fact (e.g. Table 3), and the Description Logic syntax to define expressions between the entities (individuals and classes) based on properties, such as subclasses (necessary conditions) and equivalent classes (necessary and sufficient conditions).

4.3. Mapping concepts and relations

With the implementation guidance of ISO 27002, the security ontology (cf. Section 2) has been changed. The asset concept (controls are implemented as assets) has been replaced with a compliance control concept. Each *compliance control* is a subclass of control and defines the conditions to fulfill a standard control. The flat structure of the ISO 27002 implementation guidelines could be directly mapped in the structure provided from the adopted security ontology. Clauses, main security categories, and controls of the ISO 27002 are mapped as individuals of *Standard Control*. Each clause is linked via the property *standardControl_hasChild_StandardControl* (or vice versa with the inverse property *standardControl_hasParent_StandardControl*) to the respective main security categories, which in turn are again linked via the same property to the controls.

An example of modeling standard controls”. Table 3 illustrates the relations necessary to model the structure of the standard controls. For instance, the clause:

5. Security Policy contains the main category

5.1 Information Security, which in turn contains the controls

5.1.1 Information security policy document and
5.1.2 Review of the information security policy.

Table 3: Example for mapping the hierarchical structure of Standard Controls

`‘A.5 Security Policy’`	
standard_Control_hasChild_StandardControl	`‘A.5.1 Information Security Policy’`
`‘A.5.1 Information Security Policy’`	
standard_Control_hasChild_StandardControl	`‘A.5.1.1 Information Security Policy Document’`
`‘A.5.1 Information Security Policy’`	
standard_Control_hasChild_StandardControl	`‘A.5.1.2 Review of the information security policy’`

Further relevant information about the control objectives and control statements are provided by an annotation field *objective*, containing the information described in the standard. Furthermore, each control has an annotation field *control* (and where applicable also a field *otherInformation*) with the respective additional descriptions.

An example of modeling controls. The *Entry Checkpoint Control* has been introduced in the security ontology as a standard control which demands an organization to have an entry checkpoint. The Entry Checkpoint Control Compliance Building class defines the conditions that defines the required implementation (see Table 4). The Entry Checkpoint Control Compliance Building class comprises all individuals that are classified as Building and has an *asset_contains_asset* relation to an individual in the class *Entry Checkpoint* (cf. Eq. (1)). The Entry Checkpoint Control Compliance Building is further modeled as a subclass ($\sqsubseteq$) of compliance control (cf. Eq. (2)). Individuals that are connected ($\ni$) to the standard control *Entry Checkpoint Control* via the property *control_compliantWith_control* (a necessary condition) are assigned as individuals of Entry Checkpoint Control Compliance Building (cf. Eq. (3)). Thus, when reasoning the ontology, every individual of the class Building, which contains an Entry Checkpoint will be announced as compliant with the Entry Checkpoint Control.

Table 4: Entry Checkpoint Control Compliance Building

$\equiv$	`Building` *and* $\exists$ `asset_contains_asset ‘Entry Checkpoint’`	(1)
$\sqsubseteq$	`‘Compliance Control’`	(2)
$\sqsubseteq$	$\ni$ `control_compliantWith_Control ‘Entry Checkpoint Control’`	(3)

4.4. *Mapping the knowledge*

The mapping of knowledge is the manual analysis of the informal descriptions provided from ISO 27002 to the concepts introduced in the security ontology. The major challenge is the mapping of textual information of control objectives, control statements, implementation guidance, and other information. The security clause, security category and control can be directly mapped from the ISO 27001 security ontology. For the fundamental ontological structure, we defined that one step of a standard control's implementation guidance requires one or more security controls. This way security controls can be reused – either for similar standard controls within the ISO 27002 standard or for other standards and guidelines that are formalized as well.

Overall, the following steps have been conducted to ensure accurate mapping and reusing of already existing controls and formal rules:

1. Analyze the existing security ontology controls and their descriptions.
2. Analyze the ISO 27002 controls starting with the first control.
3. Decide if the ISO 27002 control requires one or more controls according to its implementation guidance.
4. Compare and search existing controls within the security ontology and decide if controls can be reused.

In case controls cannot be reused:

5. Create a new security control in the ontology considering the description of the ISO 27002 control.
6. Develop an accurate description and corresponding formal rules which describe the correct control implementation in a machine-readable way.

Mapping the Implementation Guidance of ISO 27002 - Example 1

In this example, we present some of our important findings when mapping two controls of the security clause *5 Security Policy*.

An example of mapping 5.1.1 Information security policy document. Starting with the first standard control of this clause, we could summarize all implementation guidance steps to the control *Information Security Policy Document Control* and directly relate it with the respective policy (cf. Table 5).

Table 5: Example 1 - Mapping of the Information Security Policy Document Control

`‘A.5.1.1 Information security policy document’`	
standardControl_correspondsTo_Control	`‘Information Security Policy Document Control’`

The definition of the corresponding policy is shown in Table 6. *Information Security Policy Document Control Compliance Organization* is a new subclass of Compliance Control. A reasoner can then classify each individual of class Organization implementing an *Information Security Policy* as subclass of *Information Security Policy Document Control Compliance Organization*. Based on this definition, the reasoner can decide if the respective organization is compliant with the *Information Security Policy Document Control*.

Table 6: Example 1 - Information Security Policy Document Control Compliance Organization

≡	`Organization` *and* ∃ *organization_implements_Policy* `Information Security Policy’`
⊑	`‘Compliance Control’`
⊑	∋ *control_compliantWith_Control* `‘Information Security Policy Document Control’`

An example of mapping 5.1.2 Review of the information security policy. In the next implementation guidance, two relevant requirements could be extracted:

1. it requires an owner with approved management responsibility for the development, review, and evaluation of the security policy.
2. it demands reviewing the policy should take defined management review procedures into account.

Both of these extracted requirements can be directly integrated in the *Information Security Policy Document Control* by adding them to the previous defined controls.

An example of extending the Information Security Policy Document Control with question annotations and security ontology relations. The justification why a control is applicable to a standard control (respectively to one or more guidance steps), is modeled with the annotation called *question*. For each important information or suggested implementation, one question is added to the respective control. These question annotations are aimed at aiding organizations if the obligations are fulfilled in an adequate manner with their implemented prevention measures. Table 7 presents some questions that are part of the 'Information Security Policy Document Control' (incomplete listing).

Table 7: Example 1 - Question Annotations for Information Security Policy Document Control

question	5.1.1 a) Is information security defined with objectives and scope?
question	5.1.1 b) Is the information security policy in line with the business strategy and objectives and does the management support the goals of information security?
question	5.1.1 c) Is there a framework for handling control objectives/controls that also includes risk assessment and risk management?
question	5.1.2 a) Does the policy have an owner with approved management responsibility regarding development, review, and evaluation of the security policy?

In the next step, we modeled the relations to the standards introduced in the security ontology. Thereby, we considered the following concepts relating to the newly introduced controls:

- determine the control type of the control,
- identify the vulnerabilities that could be mitigated by the control,
- identify the threats that exploit a corresponding vulnerability, and
- identify the impacts which are the consequences of the realized threats.

Table 8 shows the relations for the previous introduced control *Information Security Policy Document Control*. The control type is *preventive* as the policy should define overall goals and management of information in general. It can also be considered as a *corrective control* type as it should include some kind of business continuity management. The related threats to the vulnerability *No Information Security Policy* are various, as illustrated in Table 8. It can be assumed that a missing *Information Security Policy* has a wide impact on the organization and so the *Impacts* are addressed and for the sake of simplicity presented in a condensed form.

Table 8: Example 1 - Mapping of Vulnerabilities, Control Types, Threats, Impacts

‘Information Security Policy Document Control’	
control_mitigates_Vulnerability	‘No Information Security Policy’
control_ofType_ControlType	‘Preventive’ ‘Corrective’
‘No Information Security Policy’	
vulnerability_exploitedBy_Threat	‘Sabotage’, ‘Employees Misconduct’, ‘Unauthorized Physical Access’, ‘Vandalism’
‘Sabotage’	
threat_leadsTo_Impact	‘Reputation Loss’, ‘Data Loss’, ‘Data Disclosure’, ‘Asset Loss’, ‘Asset Damage’, ‘Data Integrity Loss’

Mapping the Implementation Guidance of ISO 27002 - Example 2

In this example, we present one of the mapped controls of the security clause *11 Physical and environmental security*.

An example of mapping 11.2.2 Supporting Utilities. The guideline *11.2.2 Supporting Utilities* is subdivided into several steps and we identified 10 questions in the analysis process. However, the guideline requires the implementation of two sub-controls to implement the control correctly:

1. the Supporting *Utilities Control* and
2. the *Uninterruptible Power Supply (USP) Control*.

While the first control demands a policy for supporting utilities, the second needs an uninterruptible power supply for certain assets (cf. Table 9).

Table 9: Example 2 - Mapping of Supporting Utilities to Controls

‘Supporting Utilities’	
standardControl_correspondsTo_Control	‘Supporting Utilities Control’, ‘Uninterruptible Power Supply Control’

The first control requires only a policy – and is therefore very similar to Example 1 above. The second control *Uninterruptible Power Supply Control* on the other side is more complex – from the guidance we could reveal that all supporting utilities like water supply, air conditioning etc. should be adequately supported by UPS systems. The derived rules for *Uninterruptible Power Supply Control Compliance Movable Asset* are shown in Table 10.

Table 10: Example 2 - Uninterruptible Power Supply Control Compliance Movable Asset

≡	‘Air Condition System’ *or* ‘(Intrusion) Alarm System’ *or* ‘Humidity Surveillance System’ *or* ‘IT Component’ *or* ‘Smoke Detector’ *or* ‘Temperature Surveillance System’ *or* ‘Water Alarm System’ *and* ∃ *‘asset_connectedTo_Asset ’* ‘Uninterruptible Power Supply Unit’
⊑	‘Compliance Control’
⊑	*control_compliantWith_Control* ‘Uninterruptible Power Supply Control’

4.5. Linking related controls

In addition to the previous introduced relationships, we defined a property that links standard controls to controls that are not directly applicable but somehow related.

For instance, the control *5.1.2 Review of the information security policy* contains references to other guidelines, i.e. the textual description contains links such as "see 18.2.1". Therefore, the most relevant relations have been extracted (including relations that are not explicitly mentioned) and are also mapped via the *standardControl_relatedTo_Control* property. E.g., the standard control *5.1.2 Review of the information security policy* is related to the control *Information Security Independent Review Control* (introduced with standard control 18.2.1 Independent review of information security).

4.6. Mapping of Hardware and Software Components

The concept of mapping hardware and software has been slightly adapted. Hardware as IT Component is a Tangible Asset, while Software is an Intangible Asset. The original approach demanding for example an IT Component to have antivirus software installed cannot fit all requirements. Assuming the antivirus software is managed in a central way over the network, or having virtual machines the original concept cannot be used. The resulting consequence is demanding a System (intangible) having Software installed. IT components can be linked to systems. In this way, however, a system must be defined as required – it can be simply an operating system, a ticket system, a development or test system, network operating system etc.

For instance, a rather small organization might have antivirus software installed independently on each workstation (e.g., there are various instances, requiring separate updates). This way a workstation will be connected to an operating system that in turn has the respective software installed. Larger organizations might have a centralized management of antivirus software that allows handling updates automatically on each workstation. In this situation all workstations can be connected to a single system (e.g., company network) and this system is connected to the required software.

5. Application and Evaluation

After mapping the ISO 27002 standard to the security ontology, the ontology can be used to support automated ISO 27002 compliance checking. For evaluation purposes, we modeled a small software development company together with its assets to the security ontology. Subsequently, we highlight the compliance check with three representative ISO 27002 controls that are based on different implementation types:

1. a control that contains an organizational implementation (*6.1.6 Contact with authorities*),
2. a control that requires a physical implementation (*11.1.4 Protecting against external and environmental threats*), and
3. a control that relies on a technical implementation (*12.2.1 Controls against malware*).

5.1. Description of the Sample Standard Controls

In the following, the formal rules are listed that have been derived for the controls to be evaluated. The necessary and sufficient conditions are used to check the compliance of the organization, while the necessary conditions indicate the related security controls which in turn lead to the associated standard control.

Table 11 presents the derived compliance controls for *6.1.6 Contact with authorities*. For this standard control only one control has been derived that demands an organizational implementation. The standard control *11.1.4 Protecting against external and environmental threats* requires three controls (cf. Table 12). Two of them rely on an organizational implementation while the third one demands a technical implementation (fire extinguisher or fire suppression system). The standard control *12.2.1 Controls against malware* comprises several steps and thus four controls have been derived in the end (see Table 13). Only one of them does not have an organizational character – the control for antivirus software requires a technical implementation.

Table 11: Compliance Controls for '6.1.6 Contact with authorities'

'Contact with Authorities Control Compliance Organization'	
$\equiv$	'Organization' *and* $\exists$ *organization_implements_Policy* 'Contacts with Authorities Policy'
$\sqsubseteq$	'Compliance Control'
$\sqsubseteq$	$\ni$ *control_compliantWith_Control* 'Contacts with Authorities Control'

Table 12: Compliance Controls for '11.1.4 Protecting against external and environmental threats' (excerpt)

'Material Storing Control Compliance Organization'	
$\equiv$	'Organization' *and* $\exists$ *organization_implements_Policy* 'Material Storing Policy'
$\sqsubseteq$	'Compliance Control'
$\sqsubseteq$	$\ni$ *control_compliantWith_Control* 'Material Storing Control'
'Data Backup Storage Control Compliance Organization'	
$\equiv$	'Organization' *and* $\exists$ *organization_implements_Policy* 'Data Backup Storage Policy'
$\sqsubseteq$	'Compliance Control'
$\sqsubseteq$	$\ni$ *control_compliantWith_Control* 'Data Backup Storage Control'
'Fire Extinguisher Control Compliance Section'	
$\equiv$	'Section' *and* $\exists$ *asset_contains_Asset* ('Fire Extinguisher' *or* 'Fire Suppression System')
$\sqsubseteq$	'Compliance Control'
$\sqsubseteq$	$\ni$ *control_compliantWith_Control* 'Fire Extinguisher Control'

5.2. Running a Compliance Checking Process

For evaluation purposes, Protégé 4 has been used, together with the pre-installed reasoner Fact++ (Protégé also provides Hermit as standard reasoner; others can be installed as plug-ins like Pellet). In the following, we illustrate how we modeled the assets of an organization and controls based on an exemplary use case and analyzed how the inferred results can be interpreted.

5.2.1. Mapping the Assets to the Security Ontology

Initially, a new instance of Organization is created, called *Test Business*, and all assets of the organization are modeled in the ontology. The organization has a policy for *Contact with Authorities*. *Test Business* also implements a policy regarding *Material Storing*. Moreover, *Test Business* possesses a small *Building* with three *Sections* (rooms). *Section One* contains two *Fire Extinguishers* and *Section Two* has a *Fire Suppression System* installed. Table 14 presents the mapping of the organization's assets and the defined policies.

Table 13: Compliance Controls for ‘12.2.1 Controls against malware’

‘Private Software and Hardware Control Compliance Organization’	
≡	‘Organization’
	and ∃ *organization_implements_Policy* ‘Private Software and Hardware Use Policy’
⊑	‘Compliance Control’
⊑	∋ *control_compliantWith_Control* ‘Private Software and Hardware Use Control’
‘Internet Regulation Control Compliance Organization’	
≡	‘Organization’ *and* ∃ *organization_implements_Policy* ‘Internet Regulation Policy’
⊑	‘Compliance Control’
⊑	∋ *control_compliantWith_Control* ‘Internet Regulation Control’
‘Malicious Code Protection Procedures Control Compliance Organization’	
≡	‘Section’ *and* ∃ *asset_contains_Asset* ‘Malicious Code Protection Policy’
⊑	‘Compliance Control’
⊑	∋ *control_compliantWith_Control* ‘Malicious Code Protection Control’
‘Antivirus Software Control Compliance System’	
≡	‘System’
	and ∃ *system_hasInstalled_Software* ‘Transaction Security and Virus Protection Software’
⊑	‘Compliance Control’
⊑	∋ *control_compliantWith_Control* ‘Antivirus Software Control’

Table 14: Example for Mapping Policy, Building, And Section

‘Test Business’	
organization_implements_Policy	‘Contacts with Authorities Policy of Test Business’
organization_housedIn_Building	‘Building of Test Business’
‘Building of Test Business’	
building_contains_Section	‘Section One’
‘Section One’	
asset_contains_asset	‘Fire Extinguisher One’

Furthermore, we modeled a *Workstation* in *Section One*, and three *Workstations* and a *Server* in *Section Two*. Considering our example company as relatively small, each computer has its own *Transaction Security* and *Virus Protection Software* installed. The workstations are using *Windows 7* as *Operating System*, and the server has *Windows Server 2008* installed. Our exemplary organization has policies for *Malicious Code Protection*, *Private Software and Hardware Use*, and *Internet Regulation* defined. The computers are connected to our organization based on the *organization_owns_asset* property (cf. Table 15). The computer is then connected with a system – in this case it is an operating system. The systems, i.e. the operating systems, in turn have the antivirus software installed. Additionally, it is also possible to map the computers to the place (rooms) they belong to. This mapping is conducted for all workstations and the server.

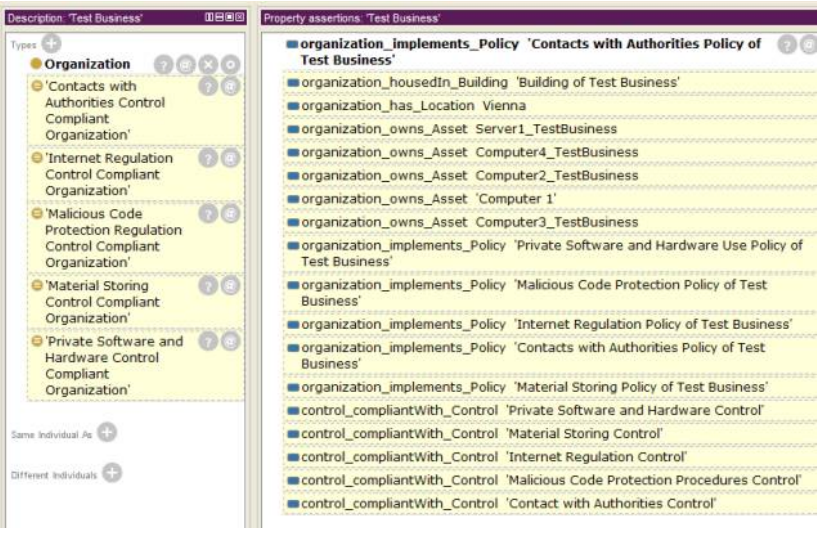
The modeled individuals and their relations serve as representation of the organization's assets and installed controls and thus as snapshot for the analysis process, which is presented in the next section.

Table 15: Example for Mapping Computer with Installed Software

`'Test Business'`	
organization_owns_Asset	`'Computer 1'`
`'Computer 1'`	
ITComponent_connectedTo_System	`'Windows 7 License 1'`
`'Windows 7 License 1'`	
system_hasInstalled_Software	`'Antivirus Software 1'`
`'Section Two'`	
asset_contains_Asset	`'Computer 2'`

5.2.2. *Results for 6.1.6 Contact with authorities - Sample 1*

In the first compliance check, the reasoner can directly infer that *Test Business* fulfills the requirements (cf. Table 11) of being classified as *Organization* which has implemented the policy *Contacts with Authorities Policy* (∃ *organization_implements_Policy*). This security control is the only control to meet the standard control *6.1.6 Contact with authorities* as defined in Table 11. Therefore, *Test Business* is compliant with the ISO 27002 control *6.1.6 Contact with authorities*. Figure 3 illustrates the inferred results in Protégé showing the compliance results of *Test Business*.


Description: 'Test Business'
Types
Same Individual As
Different Individuals
Property assertions: 'Test Business'

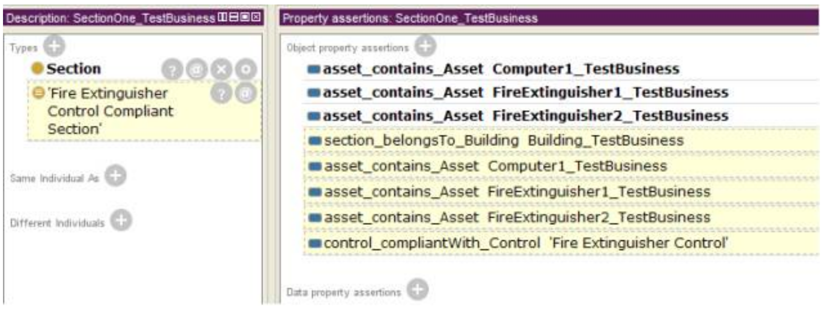
Organization
'Contacts with Authorities Control Compliant Organization'
'Internet Regulation Control Compliant Organization'
'Malicious Code Protection Regulation Control Compliant Organization'
'Material Storing Control Compliant Organization'
'Private Software and Hardware Control Compliant Organization'

organization_implements_Policy 'Contacts with Authorities Policy of Test Business'
organization_housedIn_Building 'Building of Test Business'
organization_has_Location Vienna
organization_owns_Asset Server1_TestBusiness
organization_owns_Asset Computer4_TestBusiness
organization_owns_Asset Computer2_TestBusiness
organization_owns_Asset 'Computer 1'
organization_owns_Asset Computer3_TestBusiness
organization_implements_Policy 'Private Software and Hardware Use Policy of Test Business'
organization_implements_Policy 'Malicious Code Protection Policy of Test Business'
organization_implements_Policy 'Internet Regulation Policy of Test Business'
organization_implements_Policy 'Contacts with Authorities Policy of Test Business'
organization_implements_Policy 'Material Storing Policy of Test Business'
control_compliantWith_Control 'Private Software and Hardware Control'
control_compliantWith_Control 'Material Storing Control'
control_compliantWith_Control 'Internet Regulation Control'
control_compliantWith_Control 'Malicious Code Protection Procedures Control'
control_compliantWith_Control 'Contact with Authorities Control'

Figure 3: Reasoning Result for Sample 1 in Protégé

5.2.3. *Results for 11.1.4 Protecting against external and environmental - Sample 2*

The reasoner infers that our organization is only compliant with the Material Storing Control (cf. Figure 4). The organization does not implement the policy for Data Backup Storage that is part of the standard control *11.1.4 Protecting against external and environmental threats*. Moreover, the compliance control *Fire Extinguisher Control Compliance Section* classifies sections that contain a fire extinguisher or fire suppression system (cf. Table 12). Figure 5 illustrates that *Section One* and *Section Two* fulfill the obligations to have a *Fire Extinguisher* installed. Since *Section Three* has neither a *Fire Extinguisher* nor a *Fire Suppression* system installed, it is not compliant with the defined control.


Description: SectionOne_TestBusiness
Types
Same Individual As
Different Individuals
Property assertions: SectionOne_TestBusiness
Object property assertions
Data property assertions

Section
'Fire Extinguisher Control Compliant Section'

asset_contains_Asset Computer1_TestBusiness
asset_contains_Asset FireExtinguisher1_TestBusiness
asset_contains_Asset FireExtinguisher2_TestBusiness
section_belongsTo_Building Building_TestBusiness
asset_contains_Asset Computer1_TestBusiness
asset_contains_Asset FireExtinguisher1_TestBusiness
asset_contains_Asset FireExtinguisher2_TestBusiness
control_compliantWith_Control 'Fire Extinguisher Control'

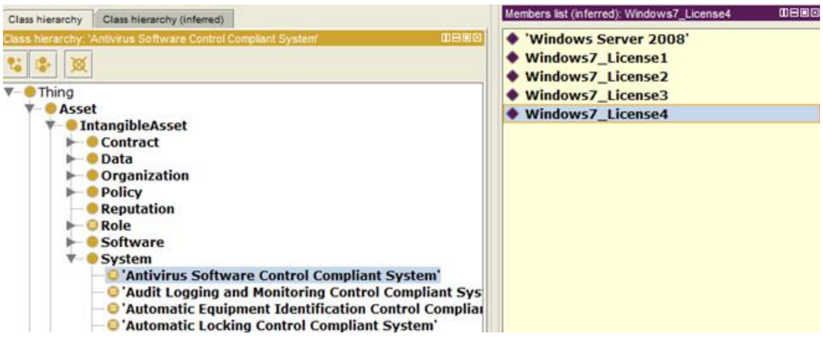
Figure 4: Reasoning Result for Sample 2 in Protégé

The results of the visualization shows that the test organization is partly compliant with *11.1.4 Protecting against external and environmental* since the backup storage policy is not implemented and not all sections have fire countermeasures installed.

5.2.4. Results for 12.2.1 Controls against malware - Sample 3

Figure 4 illustrates that Test Business is compliant with the controls *Private Software and Hardware Control, Malicious Code Protection Procedure Control, and Internet Regulation Control*. Those are three of the four controls necessary to be compliant with the ISO 27002 control *12.2.1 Controls against malware*.

The fourth control requires the organization's system having a *Transaction Security* and *Virus Protection Software* installed. The defined class *Antivirus Software Control Compliance System* and its inferred members reveal that all systems of *Test Business* are compliant as illustrated in Figure 5.


Class hierarchy
Class hierarchy (inferred)
Class hierarchy: 'Antivirus Software Control Compliant System'
Members list (inferred): Windows7_License4

Thing
Asset
IntangibleAsset
Contract
Data
Organization
Policy
Reputation
Role
Software
System
'Antivirus Software Control Compliant System'
'Audit Logging and Monitoring Control Compliant Sys
'Automatic Equipment Identification Control Complia
'Automatic Locking Control Compliant System'

'Windows Server 2008'
Windows7_License1
Windows7_License2
Windows7_License3
Windows7_License4

Figure 5: Inferred Members of Antivirus Software Control Compliance System, Sample 3

Thus, all controls for the standard control are fulfilled, and thus *Test Business* is automatically inferred as compliant with *12.2.1 Controls against malware*.

5.3. Lessons Learned

We conducted the evaluation in a small software development company to learn about potential problems during the assessment phase (i.e., modeling th assets within the ontology) and the compliance checking phase (i.e., visualizing and interpreting the results).

As described in the previous chapters, the developed methodology requires the user to model the organization's assets within the security ontology. That is, the user has to model the physical environment such as buildings, rooms, etc., the virtual environment such as computers, networks, etc., and organizational environment such as existing policies within the ontology. For evaluation purposes we build an intuitive tool which supports the user with integrating this kind of information into the ontology. We did not experience any problems with integrating the actual information, but saw that it required some effort to extract the required information from various places. The required information included: (i) relevant buildings and rooms, (ii) departments and employees, (iii) IT infrastructure (PCs, notebooks, VMs, smart phones, etc.) and their physical and virtual location, and (iv) implemented countermeasures such as policies or fire extinguishers and their sphere of action.

As in every compliance and risk management project it was no easy task to compile all of this information. In our evaluation case there was no single person that knew everything, so it was necessary to arrange several meetings with different people. Some employees were very reluctant to reveal certain types of information. Cooperating with long-term employees which know the building and people was definitely an advantage. Support from the management side was also helpful for the data collection.

In the compliance checking phase we showed the result set, i.e., compliant and non-compliant controls, to the management and got the following feedback:

- The initial information gathering regarding the infrastructure and already implemented countermeasures took some time but it has shown that it is necessary to have this information in a central place and keep it up to date (independent of the applying compliance checking methodology).

- It was helpful that the developed methodology only asked 'questions' regarding the status quo and that it did not require the employees to understand the logic of ISO 27002. As such the methodology replaces external consultants in some parts.

- One problem was that the methodology asks simple yes/no questions regarding the implementation of countermeasures. It would be helpful to also allow the specification of different implementation qualities (e.g., different qualities of data back up policies).

- Although the initial information gathering was associated with some effort, management is confident that the yearly review of the compliance status will take significantly less time as the modeled information can be reused without any additional effort.

- Management was aware that this was no 'real' ISO 27002 certification, but mentioned that the preparation for a real certification can be lowered by going through this structured approach.

6. Conclusion

Handling information security and implementing appropriate controls became more and more important especially as the interconnectivity between organizations is increasing. Organizations need an easy way for identifying vulnerabilities, related threats and appropriate controls.

This paper has proposed an ontology mapping methodology for security standards and has demonstrated it on the example of ISO 27002. The resulting ontology allows an organization to model its assets within the ontology, and automatically check their compliance with the ISO 27002 standard controls. In the evaluation part we showed the compliance checking process on use cases in a real-world setting. The evaluation has shown that the developed methodology does not require the user to understand the full logic of the information security standard. On the one hand it enables users to conduct the audit more efficiently; on the other hand it poses the risk that the dependencies among the controls and the purpose of the controls are not understood or overseen by the user. Therefore, we recommend using the methodology only with the necessary background knowledge (logic of the information security standard, understanding regarding the application field, purpose of the security controls, and their interdependencies).

During the mapping process the following problems had to be solved to achieve the proposed goal:

- *ISO 27002 guidance numeration:* ISO 27002 is not really a norm, instead, it is a code of best practice recommendations. These recommendations are partly structured and numerated. Nevertheless, we identified the need to extract questions and relate concepts since numerations have been continued, changed or even introduced in order to provide a consistent granularity and a clear presentation of the ISO 27002 standard.

- *Harmonization and detection of similar concepts:* Some necessary concepts (mainly controls) have already been defined in the fundamental ontology. However, the naming sometimes was misleading or was too specific and required harmonization to meet a general interpretable and understandable concept.

- *Decision of reusing or introducing controls:* Some controls of the ISO 27002 are very similar and sometimes they overlap. Regardless to which security clause they belong, overlapping controls make it difficult to say whether a new security control is necessary or an existing one can be reused. However, as each security control contains at least one question annotation corresponding to the control in ISO 27002, it is clarified what important aspects should be considered for the respective security control.

Future work is directed towards addressing the aforementioned challenges and increasing the efficiency of the inventory process (i.e., mapping the organizational structure to the ontology).

References

[1] Symantec Corporation, State of Information Global Results, 2012. Available at: http://www.symantec.com/content/de/de/about/downloads/press/pr-sym-2012-state-of-information-report-global.pdf, last accessed on 2013-11-23.

[2] Symantec Corporation, Internet Security Threat Report 2013, 2013. Available at: http://www.symantec.com/content/de/de/about/downloads/press/pr-sym-2012-state-of-information-report-global.pdf, last accessed on 2013-11-23.

[3] ISO/IEC, International Standard ISO/IEC 27002. Information technology - Security techniques - Code of practice for information security management, 2013.

[4] S. Fenz, A. Ekelhart, Formalizing information security knowledge, in: Proceedings of the 4th international Symposium on information, Computer, and Communications Security, 2009.

[5] S. Fenz, T. Pruckner, A. Manutscheri, Ontological Mapping of Information Security Best-Practice Guidelines, in: Proceedings of the 12th International Conference on Business Information Systems, 2009, pp. 49–60.

[6] A. Souag, R. Mazo, C. Salinesi, I. Comyn-Wattiau, Reusable knowledge in security requirements engineering: a systematic mapping study, Requirements Engineering Journal (2015) 1–33.

[7] V. Raskin, C. F. Hempelmann, K. E. Triezenberg, S. Nirenburg, Ontology in information security: a useful theoretical foundation and methodological tool, in: NSPW '01: Proceedings of the 2001 workshop on New security paradigms, ACM Press, New York, NY, USA, 2001, pp. 53–59. doi:http://doi.acm.org/10.1145/508171.508180.

[8] M. Schumacher, Security Engineering with Patterns - Origins, Theoretical Model, and New Applications, Springer, 2003.

[9] A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, IEEE Transactions on Dependable and Secure Computing 1 (2004) 11–33.

[10] A. Kim, J. Luo, M. Kang, Security ontology for annotating resources., in: OTM Conferences (2), 2005, pp. 1483–1499.

[11] G. Denker, L. Kagal, T. W. Finin, M. Paolucci, K. P. Sycara, Security for DAML web services: Annotation and matchmaking., in: International Semantic Web Conference, 2003, pp. 335–350.

[12] L. A. F. Martimiano, E. dos Santos Moreira, An OWL-based security incident ontology, 2005. URL: protege.stanford.edu/conference/2005/submissions/posters/poster-martimiano.pdf.

[13] M. Karyda, T. Balopoulos, L. Gymnopoulos, S. Kokolakis, C. Lambrinoudakis, S. Gritzalis, S. Dritsas, An ontology for secure e-government applications, in: ARES '06: Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06), IEEE Computer Society, Washington, DC, USA, 2006, pp. 1033–1037. doi:http://dx.doi.org/10.1109/ARES.2006.28.

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

[15] A. Ekelhart, S. Fenz, G. Goluch, E. Weippl, Ontological mapping of common criteria's security assurance requirements, in: 22nd IFIP TC-11 International Information Security Conference (IFIPSEC07), 2007.

[16] W3C, RDF Resource Description Framework, 2004. Available at: www.w3.org/RDF/, last accessed on 2013-11-23.

[17] W3C, OWL web ontology language, 2004. Available at: http://www.w3.org/TR/owl-features/, last accessed on 2013-11-23.

[18] W3C, SPARQL query language for RDF, 2007. Available at: http://www.w3.org/TR/rdf-sparql-query/, last accessed on 2013-11-23.

[19] S. Fenz, G. Goluch, A. Ekelhart, B. Riedl, E. Weippl, Information security fortification by ontological mapping of the iso/iec 27001 standard, in: Proceedings of the 13th Pacific Rim International Symposium on Dependable Computing, 2007, pp. 381–388.

[20] DCSSI, Expression des Besoins et Identification des Objectifs de Scurit (EBIOS), 2004.

[21] BSI, IT-Grundschutz-Manual, 2004.

[22] S. Fenz, Ontology-based generation of it-security metrics, in: Proceedings of the 2010 ACM Symposium on Applied Computing, 2010, pp. 49–60.

[23] Protégé Project, Protégé, ???? Available at: http://protege.stanford.edu/, last accessed on 2013-11-23.

[24] ISO/IEC, International Standard ISO/IEC 27001. Information technology - Security techniques Information security management systems - Requirements, 2001.

[25] T. Gruber, Ontology (Computer Science) - definition in Encyclopedia of Database Systems, 2008. URL: http://tomgruber.org/writing/ontology-definition-2007.htm.