Formalizing Information Security Knowledge

Stefan Fenz
Vienna University of Technology
Vienna, Austria
fenz@ifs.tuwien.ac.at

Andreas Ekelhart
Secure Business Austria
Vienna, Austria
aekelhart@securityresearch.at

ABSTRACT

Unified and formal knowledge models of the information security domain are fundamental requirements for supporting and enhancing existing risk management approaches. This paper describes a security ontology which provides an ontological structure for information security domain knowledge. Besides existing best-practice guidelines such as the German IT Grundschutz Manual also concrete knowledge of the considered organization is incorporated. An evaluation conducted by an information security expert team has shown that this knowledge model can be used to support a broad range of information security risk management approaches.

Categories and Subject Descriptors

I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representations (procedural and rule-based)

General Terms

Security

Keywords

Security ontology, information security, risk management

1. INTRODUCTION

Recent studies (e.g. [37]) have shown that the lack of information security knowledge at the management level is one reason for inadequate or non-existing information security risk management strategies and that raising management's information security awareness and knowledge level leads to more effective strategies. [35] and [31] identified information security risk management as one of the top ten grand challenges in information technology security and demanded sound theories and techniques to support and enhance existing risk management approaches. In 2006, the European Network and Information Security Agency (ENISA) addressed these issues in [15] and rated the establishment of unified information bases for information security risk management and the need for risk measurement methods as high priority issues. [1] shortly afterwards attested the lack of a set of well-defined formal models for supporting the information security risk management process in 2007.

Regardless of which information security risk management methodology is considered, it always includes the assessment of business crucial assets and the assessment of potential threats, corresponding vulnerabilities and controls which are able to minimize the risk to an acceptable level [6]. While intensive knowledge about the organization itself and the entire information security domain is fundamental to each information security risk management approach [21], only little research has been conducted on the formal knowledge representation of the domains which are relevant to information security risk management (cf. [17, 24, 33]).

Incomplete knowledge about the information security domain in general and the current information security status of the organization is one of the main problems in information security risk management. Therefore, the research question of this contribution is: *To what extent can the information security domain knowledge, including concepts and relations which are required by common information security risk management methodologies, be modeled formally? Which source can be used to enrich the knowledge model with concrete and widely accepted information security knowledge?*

In order to solve the research question a combination of conceptual-analytical, artifact-building and artifact-evaluating research approaches [20] has been utilized. The conceptual-analytical approach, including a second-order study of existing information security literature, is the foundation for creating the formal knowledge model and corresponding knowledge base (artifact-building). To evaluate the developed concepts an expert evaluation has been conducted to show the benefits of the research results (artifact-evaluating).

2. THE SECURITY ONTOLOGY IN A NUTSHELL

The security ontology was proposed based on the security relationship model described in the National Institute of Standards and Technology Special Publication 800-12 [28]. Figure 1 shows the high-level concepts and corresponding relations of our ontology. A threat gives rise to follow-up threats, represents a potential danger to the organization's assets and affects specific security attributes (e.g. confidentiality, integrity, and/or availability) as soon as it exploits a vulnerability in the form of a physical, technical, or admin-

ASIACCS'09, March 10-12, 2009, Sydney, NSW, Australia.

istrative weakness, and it causes damage to certain assets. Additionally, each threat is described by potential threat origins (human or natural origin) and threat sources (accidental or deliberate source). For each vulnerability a severity value and the asset on which the vulnerability could be exploited is assigned. Controls have to be implemented to mitigate an identified vulnerability and to protect the respective assets by preventive, corrective, deterrent, recovery, or detective measures (control type). Each control is implemented as asset concept, or as combinations thereof. Controls are derived from and correspond to best-practice and information security standard controls (e.g. the German IT Grundschutz Manual [11] and ISO/IEC 27001 [19]) to ensure the incorporation of widely accepted knowledge. The controls are modeled on a highly granular level and are thus reusable for different standards. When implementing the controls, a compliance with various information security standards is implicit. The coded ontology follows the OWL-DL (W3C Web Ontology Language) [40] standard and ensures that the knowledge is represented in a standardized and formal form to enable its utilization by automated systems.

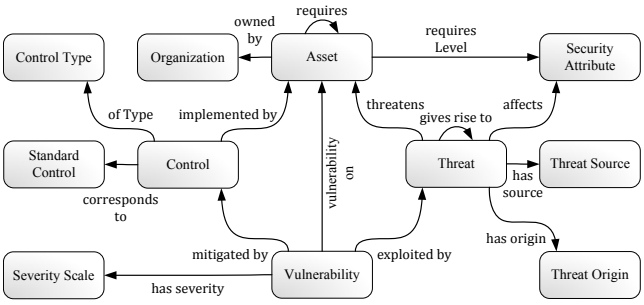

Figure 1: Security relationships

3. STRUCTURE

The security ontology comprises about 500 concepts and 600 formal restrictions; to ensure a minimal encoding bias the ontology is represented by either graphical, textual, or description logics (DL) representations, which were used to represent knowledge in a structured, formal, and reasonable form [4]. See Horrocks et al. [18] for the terminology which has been used to transform DL expressions to the actual ontology implementation language (OWL).

3.1 Purpose and Scope

The objective of the developed security ontology is to provide a knowledge model and subsequently a knowledge base on the information security domain incorporating the most relevant information security concepts (threats, vulnerabilities, assets, controls, and their implementation). While the application of such a knowledge base is manifold, the research activities have been concentrated on supporting the information security risk management domain. In contrast to related research activities (cf. [3, 14, 17, 22, 24, 26, 32, 33]) our approach focuses on providing a model for the entire information security domain including non-core concepts such as the infrastructure of an organization as well.

According to [16], the following design criteria have been considered to ensure that the final security ontology complies to existing best-practice designs in the field of ontology engineering: (1) *clarity*, (2) *coherence*, (3) *extendibility* and (4) *minimal encoding bias*. The ontological structure was derived from best-practice guidelines and information security standards, such as [28], to ensure that the concepts used and their relations are based on widely accepted standards.

3.2 Reused Ontologies and Taxonomies

Despite the fact that most existing ontologies in the field of information security only apply to a very limited scope, some of these approaches could still be used to extend our model.

First of all, the security and dependability taxonomy by [3] was used as a source for concept definitions (in the field of security attributes such as confidentiality, integrity, etc.) and high-level relationships between information security relevant concepts. [8] was used as a source for natural language definitions in the field of high-level information security concepts such as security mechanism or security policy. [7] provides an excellent source for concept definitions and high-level taxonomies, which was incorporated into our ontology in the fields of threats, cryptosystems, biometrics, and malicious logic. The cryptosystems taxonomy was refined by incorporating the work of [17]. Further concept definitions, especially for the internet security domain, were taken from the internet security glossary (RFC 2828) [34]. [30] and [42] helped us to create a high-level threat taxonomy which was refined by incorporating concrete threat concepts from the German IT Grundschutz Manual [11]. The United Nations Standard Products and Services Code [38] with its IT and telecommunication branch was used to establish the IT and telecommunication infrastructure element taxonomy, which is part of the infrastructure subontology. For interrelating high-level concepts the security relationship model presented in the National Institute of Standards and Technology Special Publication 800-12 [28] was used. The subsequent sections describe the basic concepts, relations, and formal axioms of the security ontology in more detail.

3.3 Concepts

Figure 2 shows the top-level concepts of our security ontology. The concepts were grouped in three subontologies (security, enterprise, and location) in order to enforce the context of the modeled concepts.

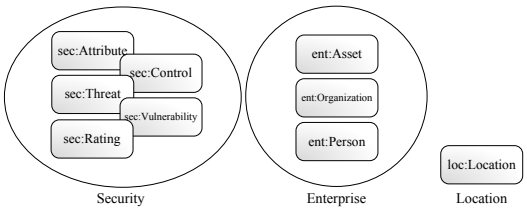

Figure 2: Security ontology concepts

3.3.1 Security Subontology

The core of the security ontology is the security subontology, consisting of the concepts (1) attribute, (2) control, (3) threat, (4) vulnerability, and (5) rating which were derived from well established information security standards such as the German IT Grundschutz Manual [11], ISO 27001 [19],

the French EBIOS standard [13], the NIST computer security handbook [28], and the NIST information security risk management guide [36].

- The **attribute** concept (sec:Attribute $\sqsubseteq \top$) and its corresponding subconcepts are used to describe the remaining top-level concepts in more detail. For this ontology, the following concepts have been modeled so far: (1) control type (sec:ControlType $\sqsubseteq$ sec:Attribute) $\rightarrow$ used to classify control concepts as either corrective, detective, deterrent, preventive, or recovery measures, (2) security attribute (sec:SecurityAttribute $\sqsubseteq$ sec:Attribute) $\rightarrow$ defines which security attributes (e.g. accountability, availability, confidentiality, integrity, reliability, or safety) can be affected by a certain threat (3) threat origin (sec:ThreatOrigin $\sqsubseteq$ sec:Attribute) $\rightarrow$ used to indicate if a threat's origin is either human or natural, (4) threat source (sec:ThreatSource $\sqsubseteq$ sec:Attribute) $\rightarrow$ used to indicate if a threat's source is either accidental or deliberate, and (5) scale (sec:Scale $\sqsubseteq$ sec:Attribute) $\rightarrow$ provides a three-point Likert scale [25] to rate (a) potential threat impacts on the asset level, (b) the control implementation effectiveness on the control level, and (c) the severity on the vulnerability level in terms of high, medium, and low.

- As displayed in Figure 1, the **control** concept (sec:Control $\sqsubseteq \top$) is used to mitigate defined vulnerabilities by implementing either organizational (e.g. non-smoking policy) or physical (e.g. fire extinguisher) measures. Most of the modeled controls are derived from and correspond to the German IT Grundschutz Manual [11]. Each control has the following attributes: (1) control type (corrective, detective, deterrent, etc.), (2) relation to established information security standards (e.g. control *IT training of employees* corresponds to the *ISO 27001 A.8.2.2* control), (3) implementation specification, to indicate which asset concepts are required to implement the control, and (4) mitigation specification, to indicate which vulnerabilities could be mitigated by implementing the control.

- According to [28], a vulnerability is the absence of a proper safeguard which could be exploited by a threat. The **vulnerability** concept (sec:Vulnerability $\sqsubseteq \top$) was subdivided into two distinct concepts: (1) administrative vulnerability (sec:AdministrativeVulnerability $\sqsubseteq$ sec:Vulnerability), and (2) technical vulnerability (sec:TechnicalVulnerability $\sqsubseteq$ sec:Vulnerability) which affects the tangible asset level. Each vulnerability concept can be exploited by predefined threat concepts, and mitigation is achieved by the implementation of one or more control concepts. In addition to that, the severity of each vulnerability concept is rated by a three-point scale (high, medium, and low) in order to enable a machine to interpret the significance of the vulnerability. All modeled vulnerabilities have been derived from the German IT Grundschutz Manual [11].

- Most of the modeled **threat** concepts (sec:Threat $\sqsubseteq \top$) have been derived from the German IT Grundschutz Manual [11] and Peltier's threat classification [30]. The threat taxonomy comprises natural (e.g. earthquake, monsoon, or lightning), accidental (e.g. hardware failure or liquid leakage), and intentional (e.g. theft or alteration of software) threats at the highest level, followed by a detailed classification. An in-depth threat description such as threat origin, threat source, corresponding vulnerabilities, as well as endangered security objectives, following the security and dependability taxonomy according to [3], are provided for each threat. Interrelations between threats are also modeled as this is essential information for risk analysis. Understanding the relationships between threats and endangered assets is vital for comprehensive security planning, which makes the integration of these connections crucial.

- The **rating** concept (sec:Rating $\sqsubseteq \top$) is used to provide a rating for control implementations and a priori threat probabilities. While the aforementioned control concept provides potential control implementation combinations, the control implementation rating concept rates the effectiveness of asset/control combinations. As certain assets can be used to implement different controls, it is obvious that certain attributes of a asset, such as the effectiveness, depend on its actual application. A safety door, for instance, can be used as a protection against several threats, such as break-in or fire. While a specific safety door is an excellent solution to contain the fire threat, it could be an insufficient solution in case of a break-in. Therefore, control implementations are rated in terms of asset effectiveness on the asset/control combination level and not directly on the asset level. The probability concept, as the second rating concept, connects threats to locations and stores the a priori threat probability distribution by using a three-point Likert scale (high, medium, and low) [25] for each threat/location combination.

3.3.2 Enterprise Subontology

The enterprise subontology provides a framework which is able to represent an organization and its environment ontologically in order to interrelate them with concepts from the information security domain. Therefore, the enterprise subontology consists of the following top-level concepts: (1) asset, (2) person, and (3) organization.

- Each **asset** concept (ent:Asset $\sqsubseteq \top$) is categorized either as a tangible (ent:TangibleAsset $\sqsubseteq$ ent:Asset) or an intangible (ent:IntangibleAsset $\sqsubseteq$ ent:Asset) asset. Typical subconcepts of intangible assets are data, role, software, or reputation.
  - The **data** concept (ent:Data $\sqsubseteq$ ent:IntangibleAsset) comprises meta-data on the knowledge of an organization. Examples of such data are policies, contracts, patents, communication data, and guidelines. Connecting this concept to the asset concept ensures that the storage site (certain servers or paper archives) is modeled. Just as it is the case with remaining asset concepts, the role of the data concept is twofold. On the one hand it can be used for control implementation (e.g. policies or guidelines), and on the other hand the data (e.g. contracts, patents, or building plans) has to be protected against defined threats.

– The **role** concept (ent:Role ⊑ ent:IntangibleAsset) distinguishes between internal (ent:InternalRole ⊑ ent:Role; e.g. administrator, developer, or internal attacker) and external (ent:ExternalRole ⊑ ent:Role; e.g. customer, vendor, or external attacker) roles. Every physical person (ent:Person) or organization (ent:Organization) relevant to the organization is connected to one or more roles, which enables a flexible handling if those concepts are to be modeled as control implementations or threatened elements. Most of the internal as well as external role concepts have been derived from the German IT Grundschutz Manual [11] which provides a comprehensive and well-defined list of roles.

– In contrast to the data concept, the **software** concept (ent:Software ⊑ ent:IntangibleAsset) has been introduced to provide an ontological structure for those virtual elements which only possess processing characteristics such as text editors, cryptosystems, or operating systems. The entire software concept and its subconcepts have been derived from the United Nations Standard Product and Service Code [38]. The cryptosystems taxonomy by [17] has been used to refine the encryption software branch (ent:EncryptionSoftware). Concrete software instances can be described by a general description, vendor, product edition, product name, and product version. Especially for threats such as computer viruses this information is essential to determine the potential impact on the organization.

Each tangible asset (ent:TangibleAsset ⊑ ent:Asset) is classified as a movable (ent:MoveableAsset ⊑ ent:TangibleAsset; e.g. computer or fire extinguisher) or an immovable asset (ent:ImmovableAsset ⊑ ent:TangibleAsset; e.g. building or door). The connections between the asset concepts allow an organization to ontologically map its entire physical infrastructure (including buildings, floors, rooms, computers, alarm systems, etc.). Since the information security risk management process requires the definition of concrete security goals, the asset's importance to the organization's mission has to be known. Therefore, attributes have been introduced to qualitatively rate the potential impact regarding confidentiality, integrity, and availability should the considered asset go down. According to Figure 1, an organization's assets are not only threatened by threats and, thus, require certain protection level, but they are also used for control implementation to subsequently mitigate vulnerabilities. Since controls could require implementation combinations or alternatives, the optimal combination or alternative is strived for. Therefore, the aforementioned asset/control effectiveness rating (ent:ControlImplementation) was established which rates the asset in the context of the actual control by a three-point Likert scale (high, medium, and low). Furthermore, domain-specific attributes such as host names for IT and telecommunication concepts or financial information such as asset and outage costs are provided. For further details request the OWL version of our ontology at `http://securityontology.securityresearch.at`.

- The **person** concept (ent:Person ⊑ ⊤) is used to model physical persons in the ontology. Currently, only the person's first and last name is stored. Future extensions may include further information, such as social insurance number, salary, or home address information, to support disaster recovery planning.

- Like the person concept, the **organization** concept (ent:Organization ⊑ ⊤) comprises organizations in the broadest sense and assigns roles to them. First of all, the organization using our ontology will be modeled to interrelate it with its assets. If a higher level of granularity was required, it would be also possible to split the considered organization up into smaller elements such as departments.

3.3.3 *Location Subontology*

Since the aim of this work is to support risk management with machine-readable and machine-interpretable knowledge, information on a priori threat probabilities have been incorporated into the security ontology. Therefore, the **location** concept (loc:Location ⊑ ⊤) only stores a list of locations, while the granularity level is not specified. The aforementioned threat probability rating concept (sec:Probability) is used to interrelate location and threat information in order to assign a priori threat probabilities. Concrete location dependent figures for a priori probabilities can be gained from, for example, insurance agencies, police records, and historical organization data. In most cases the most useful level models the locations on the city or region level, on which the probability of most threats differs only slightly.

3.4 Relations

While the previous subsection has introduced the top-level concepts, this subsection gives an overview of the most important relations between these concepts.

- Each **threat** concept could give rise to or be a consequence of another threat, thereby enabling the modeling of threat chains. Further information on the threat's nature is given by the concepts **threat source** (accidental or deliberate), **threat origin** (human or natural), and the affected **security attributes** (confidentiality, integrity, availability, etc.). To model the threat's damage potential, each threat was connected to **asset** concepts by the *threatens* relation. Connecting the threats to **vulnerability** and **probability** concepts enables the integration of threat enablers and a priori threat probabilities (ent:Probability) to support the risk assessment with a comprehensive threat knowledge base. In order to further refine the definition of vulnerabilities, the *vulnerabilityOn* relation was introduced to specify the scope of a certain vulnerability and the respective control (e.g. the vulnerability *no fire suppression system* is bound to the section concept, which means that the corresponding control is fulfilled if the concrete control implementation is able to protect the given section).

- To counteract certain threats, the mitigation relation between the **control** and the vulnerability concept was modeled. Each control corresponds to a **standard control** (ISO 27001, German IT Grundschutz

Manual, etc.) and is of a certain **control type** (corrective, detective, deterrent, preventive, or recovery). Controls are implemented by **asset** concepts (e.g. fire extinguisher, software firewall, security guard, etc.). Complementary implementations (e.g. the need for a smoke detector and a fire extinguishing system) as well as implementation alternatives (e.g. facial scan or fingerprint scan) are incorporated into the knowledge base.

- What is to be protected against threats are the organization and its business mission. Therefore, the **organization** concept has been connected to its assets by the *ownedBy* relation. The impact on the organization in the case of a loss is rated for each asset regarding confidentiality, integrity, and availability. The **requires** relation is used to model dependent assets (e.g. electronic data which requires an IT and telecommunication element to be stored on).

- To model an organization's physical infrastructure, the following model is proposed: the **building** concept represents the building in which the organization is located, contains one or more **levels** (building *contains* level), and is located on a specific **site** (building *locatedIn* site). Each level within the level concept is described by its vertical position in the building by connecting it by relation *isAbove* to the level which is beneath and by relation *isUnder* to the level which is above the considered level. Each level and site could contain one or more areas, which again could contain one or more sections (e.g. rooms), which are connected to each other by section connectors (e.g. doors or windows). Each section could contain zero or more movable assets or roles. This model enables the organization to model its physical infrastructure in a formal and machine-readable way.

3.5 Formal Axioms

Up to this point, the concepts and the corresponding relations, which are necessary to represent the information security domain as an ontological framework, have been discussed. In addition, this subsection introduces the most relevant selection of formal axioms, which have been modeled in our ontology in order to represent the information security knowledge most granularly but still formally. While formal axioms are used throughout the ontology, this section concentrates on axioms which have been used to model the organization's physical environment and recommended control implementations. For a comprehensive set of the axioms used request the OWL version of our security ontology at `http://securityontology.securityresearch.at`.

As already mentioned, controls are implemented by asset concepts. The connection between the control and asset concepts is realized by a n:m relation. Since one control allows different implementation possibilities, it has to be expressed formally to ensure the reusability of these definitions. Therefore, the concept of restricted quantifiers [5] was used. The value restriction constructor ($\forall$) is used to describe a given control in more detail by constraining the *implementation* side to specific concepts. For example, a fully implemented *access regulation* control has to implement a security guard, an entry checkpoint, and/or an access system, which is expressed as follows:

$$\text{AccessRegulationControl} \sqsubseteq \text{PhysicalAndEnvironmentalSecurityControl}$$

$$\text{AccessRegulationControl} \sqsubseteq \forall\ \text{implementedBy}\ (\text{AccessSystem} \sqcup \text{EntryCheckpoint} \sqcup \text{SecurityGuard})$$

So far, the ontology is aware of the concepts that are potentially required to implement a certain control, but a description of the possible combinations is still missing. Therefore, the existential restriction constructor ($\exists$) was used which states that at least one value for that property is of a certain type [40]. For example, the *access regulation* control requires an access system and either a security guard or an entry checkpoint in all implementation variations:

$$\text{AccessRegulationControl} \sqsubseteq \text{PhysicalAndEnvironmentalSecurityControl}$$
$$\text{AccessRegulationControl} \sqsubseteq \forall\ \text{implementedBy}\ (\text{AccessSystem} \sqcup \text{EntryCheckpoint} \sqcup \text{SecurityGuard})$$
$$\text{AccessRegulationControl} \sqsubseteq \exists\ \text{implementedBy}\ (\text{EntryCheckpoint} \sqcup \text{SecurityGuard})$$
$$\text{AccessRegulationControl} \sqsubseteq \exists\ \text{implementedBy}\ \text{AccessSystem}$$

$$\text{AccessRegulationControl} \sqsubseteq \forall\ \text{mitigates}\ \text{NoAccessRegulationControl}$$

On this account, two implementation combinations are possible: the combinations access system/security guard and the access system/entry checkpoint. This means that the *access regulation* control can be considered as fulfilled, if an access system and either an entry checkpoint or a security guard is in place. The place were the control has to be implemented is determined by the corresponding vulnerability and its *vulnerabilityOn* relation. For example: if the *vulnerabilityOn* relation of a certain vulnerability points to the section concept, the corresponding control has to be implemented on the section level to mitigate the vulnerability.

To model the organization's physical environment (immovable assets) and to embed roles and movable assets into this model, the following structure was used: first, the considered organization (or one of its branches) is limited to a certain site. Such a site is located in one location (e.g. city or region) and contains zero or more buildings. If the site also contains some kind of non-building concepts, such as a forecourt, the ontology allows its integration by relating the site to zero or more areas. The site and all subsequently described concepts are subconcepts of the *immovable asset* concept which is again a subconcept of the *asset* concept (ent:ImmovableAsset $\sqsubseteq$ ent:Asset $\sqsubseteq \top$). See Table 1 for the formal definitions.

$\text{Site} \sqsubseteq \text{ImmovableAsset}$	$\text{Site} \sqsubseteq \forall\ \text{locatedIn}\ \text{Location}$
$\text{Site} \sqsubseteq \forall\ \text{contains}\ (\text{Area} \sqcup \text{Building})$	$\text{Site} \sqsubseteq = \text{locatedIn}\ \text{Location}$

Table 1: Formal site concept definitions

Each building is located in one specific site and contains one or more levels. See Table 2 for the formal definitions.

$\text{Building} \sqsubseteq \text{ImmovableAsset}$	$\text{Building} \sqsubseteq \forall\ \text{contains}\ \text{Level}$
$\text{Building} \sqsubseteq \forall\ \text{locatedIn}\ \text{Site}$	$\text{Building} \sqsubseteq \exists\ \text{contains}\ \text{Level}$
$\text{Building} \sqsubseteq = \text{locatedIn}\ \text{Site}$	

Table 2: Formal building concept definitions

Each level is located in one building and contains one or more areas. The definition of areas enables the aggregation of sections (e.g. rooms), to allow for the definition of areas with different security needs. See Table 3 for the formal definitions.

Furthermore, specific level concepts have been defined in order to automatically classify basement, regular, and top levels of a given building. The levels are defined, in addition

Level $\sqsubseteq$ ImmovableAsset	Level $\sqsubseteq$ $\forall$ contains Area
Level $\sqsubseteq$ $\forall$ locatedIn Building	Level $\sqsubseteq$ $\exists$ contains Area
Level $\sqsubseteq$ = locatedIn Building	

Table 3: Formal level concept definitions

to the definition above, as follows: BasementLevel $\equiv$ (=1 isAbove.ent:Level) $\sqcap$ (=0 isUnder.ent:Level), TopLevel $\equiv$ (=0 isAbove.ent:Level) $\sqcap$ (=1 isUnder.ent:Level). All other levels which do not fit these definitions are classified by the ontology reasoner as regular levels. Each area is located in one or more levels or sites and contains one or more sections. Sections could be physical constructs such as rooms or logical constructs such as zones in the surrounding area of the organization's building. The introduction of the section concept enables the organization to model its environment on a more granular level than just on the room level. Even big rooms such as warehouses can be subdivided into several logical sections, if different security requirements have to be modeled for one room. See Table 4 for the formal definitions.

Area $\sqsubseteq$ ImmovableAsset	Area $\sqsubseteq$ $\forall$ contains Section
Area $\sqsubseteq$ $\forall$ locatedIn (Level $\sqcup$ Site)	Area $\sqsubseteq$ $\exists$ contains Section
Area $\sqsubseteq$ $\exists$ locatedIn (Level $\sqcup$ Site)	

Table 4: Formal area concept definitions

The most granular entity in our infrastructure modeling concept is represented by the section concept. Each section is located in exactly one area, contains zero or more movable assets or roles, and is connected by one or more section connectors (e.g. windows, gates, or doors). See Table 5 for the formal definitions.

Section $\sqsubseteq$ ImmovableAsset	Section $\sqsubseteq$ $\forall$ contains (MoveableAsset $\sqcup$ Role)
Section $\sqsubseteq$ = locatedIn Area	Section $\sqsubseteq$ $\exists$ connectedBy SectionConnector
Section $\sqsubseteq$ $\forall$ locatedIn Area	Section $\sqsubseteq$ $\forall$ connectedBy SectionConnector

Table 5: Formal section concept definitions

After describing the purpose, concepts, relations, and formal axioms of our security ontology, it is shown how concrete information security domain knowledge, from the German IT Grundschutz Manual [11], was incorporated into the ontology.

4. KNOWLEDGE BASE

While the structure of the security ontology including top-level information security concepts, relations, and formal axioms could be applied and discussed theoretically, the structure is useless for supporting the information security risk management domain as long as there is no concrete information security domain knowledge modeled in the ontology. Therefore, common information security standards and best-practice guidelines have been evaluated regarding their acceptance, completeness, availability, and knowledge representation. Finally, the German IT Grundschutz Manual [11] has been identified as the most promising knowledge base. It provides an excellent and comprehensive knowledge base of the information security domain, is accessible without any limitations, and models the information security knowledge on a very concrete and highly granular level.

According to [2] and [23] three distinct knowledge types were mapped to the security ontology: (1) the declarative type $\rightarrow$ know-about knowledge, (2) the procedural type $\rightarrow$ know-how knowledge, and (3) the relational type $\rightarrow$ know-with knowledge. While the declarative knowledge type is represented by the ontological modeled information security concepts, procedural knowledge is provided by natural language descriptions which are stored at the corresponding concepts (e.g. the fire extinguisher concept provides a natural language definition regarding its correct usage). Relational knowledge is represented by relations between the modeled concepts (e.g. the *requires* relation, which connects concepts depending on each other to function properly). The conducted information security knowledge mappings aimed at incorporating explicit knowledge, which could be easily formalized. Further research efforts, based on existing approaches such as [12], will also concentrate on incorporating tacit knowledge, which is much more difficult to articulate and to formalize than explicit knowledge [27].

4.1 Incorporating the German IT Grundschutz Manual

After evaluating common information security standards and best-practice guidelines, the German IT Grundschutz Manual [11] was chosen to be one of the sources of information security knowledge to enrich the security ontology with concrete knowledge. The next step was to extract information from the over 3000 pages of text, to alter it to fit the security ontology structure, and to extend the security ontology knowledge base by the newly generated information.

Unfortunately, the German IT Grundschutz Manual [11] has not been designed to fit the security ontology structure and so first some incompatibility problems had to be overcome. The German IT Grundschutz Manual [11] is very beneficial when used by the human reader. It is probably the most comprehensive accumulation of generic information security related knowledge worldwide, but due to its immense coverage and complexity, it is difficult to present the results on a consistent granularity level. The German IT Grundschutz Manual [11] has grown over time and many different authors have worked on its content, so the perspectives and specificities of the various topics vary (sometimes even tremendously) resulting in incompatibilities on a logical level. As a result, the following problems and incompatibilities had to be solved and compensated during the process of mapping the German IT Grundschutz Manual [11] to the security ontology:

- **No concept for vulnerabilities:** The German IT Grundschutz Manual [11] does not work with the concept of vulnerabilities, unlike the NIST Handbook [28] on which the security ontology structure has been built. Instead, the German IT Grundschutz Manual [11] mixes the concepts of threats and vulnerabilities (e.g. the threat *inadequate domain planning* is not really a threat in the NIST Handbook's [28] point of view). In the NIST Handbook [28], *a threat is an entity or event with the potential to harm the system* and *a vulnerability is a condition or weakness in (or absence of) security procedures, technical controls, physical controls, or other controls that could be exploited by a threat.* Thus, *inadequate domain planning* can be identified as a weakness and, therefore, would be a vulnerability. In the cases the German IT Grundschutz Manual has mixed up classic vulnerabilities in their list of threats, the following approach was used: first of all it was determined if the threat was really a

threat or if it was a vulnerability according to the NIST security relationship model [28]. If a threat was identified as a vulnerability, it was classified in the security ontology and linked to the corresponding threat and control concepts. If the German IT Grundschutz Manual did not provide vulnerabilities for listed threats, vulnerabilities had to be created artificially. Our approach is again based on the NIST Handbook [28] which states: *vulnerabilities are often analyzed in terms of missing safeguards*. Therefore, vulnerabilities were derived from the existing IT Grundschutz controls by implication. For example, interpreting the control *fire doors* as *fire doors should be in place*, the derived vulnerability would be *no fire doors* meaning that no fire doors are in place. This mapping mechanism enables the incorporation of the German IT Grundschutz Manual knowledge in the security ontology while keeping its knowledge model consistent.

- **Vague connections between threats and controls:** As mentioned before, the German IT Grundschutz Manual [11] is intended to be used by the human reader. It is a 3000 pages strong reference book for system administrators, IT managers, and other people dealing with information security. Unfortunately, on a logical level it is inconsistent. Connections between threats and controls are not clearly evident. Some, but not all are described in the scrolling text. Threats and controls are listed together with the object they relate to, but no connection between a threat and the corresponding control is presented. In return, the controls are distinguished on the basis of the phase in which they should be applied. This may be useful for modeling complying to IT Grundschutz but for our purpose it is inapplicable and confusing. To sum it up, the problem was to create clear relations between a threat and the corresponding control, which initially was not possible due to the structure of the German IT Grundschutz Manual [11]. As a solution 72 cross-reference tables[1], one for each IT Grundschutz Manual module, were used to identify the connections between threats and corresponding controls to get a more structured access to the relations.

- **No relations between threats:** It should be possible to model potential threat dependencies in order to improve the risk assessment. Unfortunately, the German IT Grundschutz Manual [11] does not describe connections between individual threats. Therefore, further information security standards and best-practice guidelines such as the French EBIOS standard [13] had to be used to model them. To simplify this process a few top-level threats were identified (e.g. data disclosure, data tampering, and data loss) affecting certain security attributes (confidentiality, integrity, and availability).

- **Inconsistent granularity of information:** As the German IT Grundschutz Manual [11] has grown over time and many different authors have worked on it, the extent of information with which a topic has been treated and the perspective from which certain concepts have been described are inconsistent. For example, on the one hand the German IT Grundschutz Manual [11] proposes the threat *improper IT system administration* which objectively examined is a very vague description of a threat and can mean anything. On the other hand it covers threats like *poor planning of the migration of Exchange 5.5 to Exchange 2000* which is a very specific case and is not important for most of the readers. Since the production of a consistent knowledge base with a similar grade of information detail is aimed for, the information of the German IT Grundschutz Manual had to be filtered and changed, and topics covering very specific topics were left out in the mapping process. The mapping of topics mentioned in the allocation table *ISO 27001/27002 to German IT Grundschutz Manual*[2] were defined as the minimum for the mapping process.

- **Redundancy and overlapping of information:** Due to a number of different authors and a certain timeframe in which the document has grown, many topics contain redundant information and cover an overlapping scope. Examples for redundant information would be the threats *computer viruses* and *macro viruses*. The description of the threat *computer viruses* already covers the topic *macro viruses* and, therefore, there is no need for an extra listing, at least not on the same hierarchy level. This kind of information presentation is confusing and inconsistent on a logical level. Thus, the flat hierarchy of the German IT Grundschutz Manual [11] had to be apdopted to support a class-based information container. To overcome the problem of redundant and overlapping information, the class-based nature of OWL was used. Topics which contained redundant concepts and information were modeled with the help of subconcepts. Characteristics that are similar for certain subconcepts were modeled on the superconcept level. Only specific properties which were not shared with all other subconcepts on the same level have to be modeled individually. This enables the creation of a class-based knowledge base which can be easily improved or extended.

Summarizing we conducted the following steps for each threat contained in the IT Grundschutz Manual to map the IT Grundschutz Manual knowledge to the security ontology: (1) evaluation of the threat description granularity: if the IT Grundschutz threat description is too complex, we subdivide it into more granular threat descriptions to achieve a more granular level which fits our ontological requirements (e.g. IT Grundschutz threat T 5.3 *Unauthorized entry into a building* was subdivided into threats *sec:BreakIn* and *sec:UnauthorizedPhysicalAccess*), (2) control analysis: we use cross reference tables provided by the BSI to determine which safeguards could mitigate the considered threat; the safeguard list is analyzed regarding its potential to mitigate the considered threat (note that the considered threat does not represent the original threat granularity of the IT Grundschutz Manual); potential safeguards are modeled as

[1]German IT Grundschutz cross reference tables: `http://www.bsi.de/gshb/deutsch/download/kreuzreferenz_tabellen.zip`

[2]Allocation table - ISO 27001/27002 to German IT Grundschutz Manual: `http://www.bsi.bund.de/gshb/deutsch/hilfmi/isovergleich/Vergleich_ISO27001_GS.pdf`

controls (including control implementation descriptions) in the security ontology and are connected to corresponding standard controls (e.g. ISO27001 controls), (3) vulnerability creation: for each safeguard we create corresponding vulnerabilities and connect them to the considered threat, and (4) threat analysis: due to the highly granular threat structure in the security ontology we interrelate the considered threat to those threats which are enabled by it (sec:givesRiseTo relation) and also to those threats which act as enablers for the considered threat (sec:canBeConsequenceOf relation); further information such as threatened assets, affected security attributes, threat origin, and threat source is also incorporated in the ontological threat description at this step.

4.2 Example

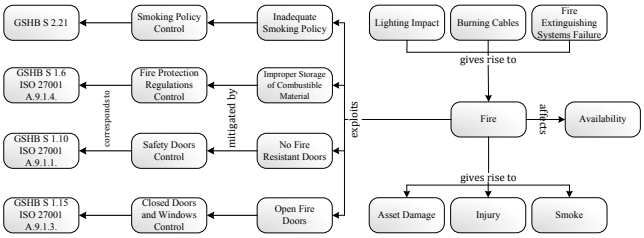

Figure 3: Fire threat

In this subsection, the applicability of the framework is shown using the example of the fire threat. Figure 3 illustrates the threat including its relationships with reference to the presented security ontology model. There is no compulsory reading direction to interpret the graph; users can start at any desired point and follow their paths of interest. In this description, the start is the threat concept *Fire*. Choosing a threat as a starting point is a common and logical decision in conducting a threat analysis. As the figure indicates, the event of fire leads to the threat Asset Damage which affects the security attribute availability. Fire has to exploit one or more vulnerabilities to take effect, such as improper storage of combustible materials or inadequate smoking policies. In the figure, only a partial set of predefined vulnerabilities is illustrated to enhance readability. It is essential for an organization to know and understand possible vulnerabilities prior to taking effective controls. It has to be kept in mind that for each element a description in natural language is given to enhance the understanding of the concepts for a human reader. For each vulnerability a corresponding control is presented as a mitigation measure and guides the organization in the vulnerability assessment step. Since controls are modeled on a highly granular level, in most cases the controls are straight forward: for example, if there are no fire resistant doors, the control would be to install appropriate fire resistant doors. The proposed controls are derived from best-practice guidelines, which can be seen on the left hand side. If controls demand physical implementation, such as the *Safety Doors* control, concrete implementation options are provided. In the case of the *Safety Doors* control, the installation of a safety door, compliant to EN1634 or a similar security rating, is required. Referring to the information gained from the knowledge model, it appears that the Safeguard 2.21 in the German IT Grundschutz Manual [11] requires the implementation of a smoking policy that corresponds to the control modeled in our framework. Due to the granular nature of controls, it is also possible that more than one information security standard control is mapped to a defined control (e.g. the *Fire Protection Regulation* control corresponds to the German IT Grundschutz Manual element *GSHB S 1.6* and *ISO 27001 A.9.1.4*). To summarize the example so far, an organization is aware of possible threats as well as vulnerabilities and mitigation controls including concrete implementation proposals by using our framework. Furthermore, compliance with existing information security standards is implicit. In addition, the framework expresses threat chains: the knowledge of threat consequences, for example that fire can be caused by lightning and that fire triggers asset damage and injury, is of paramount importance in considering threat impacts and consequently in business continuity management as well. For reasons of flexibility and reusability, no role, infrastructure, or data concepts are directly affected by the threat of fire in this framework; instead reusable subsequent threats, such as asset damage (affects assets) and injury (affects roles) have been introduced. We refer to `http://securityontology.securityresearch.at/aurum` for further information on a concrete application of the security ontology in the field of information security risk management.

After describing the ontology structure and the incorporated knowledge, it is to be shown how the structure was evaluated to ensure that it is generic enough to cover the entire information security domain and concrete enough to provide human beings and machines with information security domain knowledge to support the information security risk management process.

5. EVALUATION

According to [39], informal and formal competency questions have been used to evaluate our ontology with the help of a team of experienced information security professionals. Since most ontology evaluation approaches, as described in [9], [29], or [10], are concerned with selecting the most appropriate ontology from a set of existing ontologies, the approach by [39] has been adopted to create an evaluation methodology which is able to check an ontology against its initial requirements. Therefore, the following evaluation phases have been conducted: (1) domain expert evaluation of the ontology structure including concept definitions, relations, and formal axioms, (2) identification of informal competency questions based on best-practice guidelines and domain expert interviews, (3) creation of formal competency questions based on the informal competency questions identified in Step 1, and (4) evaluation (conducted by domain experts) of the formal competency question result sets.

As domain experts are central to the ontology evaluation methodology, a team of eight information security professionals was put together. All members have been working and researching for years in the information security domain. In short, all team members have qualified themselves as experts. They are working in industrial projects related to information security, two of them give information security related lectures at the Vienna University of Technology, seven team members hold information security certificates such as CISSP or CISA, eight members hold an academic degree in the field of computer sciences, and two team members hold a PhD degree related to information security. Although this is neither a significant nor representative group of experts, it helped improving the ontology structure as

well as the modeled information security knowledge.

5.1 Ontology Structure

At first, the goals, capabilities, and the structure of the security ontology have been introduced to the evaluation team, to ensure that the team disposes of the same information level regarding the security ontology. The first evaluation round was promising, since a few weaknesses were identified: (1) no supplier role in the external role concept, (2) no possibility to model intangible assets such as reputation or power of an organization, (3) no possibility to rate the impact on a certain asset with regard to the time factor (it may be a medium impact if the asset is down for three hours but a high impact if the asset is down for a day), (4) no possibility to incorporate more granular organizational units such as departments, (5) it is not possible to rate the information security awareness level of roles or persons, (6) no incorporation of legal issues or threats against the environment, and (7) no business continuity standards are incorporated into the security ontology.

Since the expert team originates from several information security domains, such as business continuity management or multi-objective decision support regarding control selection, different requirements had to be dealt with, which were not always in the information security risk management domain. Therefore, each change request was evaluated due to its necessity in the field of information security risk management support: (1) a supplier concept was created as an external role since suppliers are fundamental roles in any organization, (2) the asset concept was split into two subconcepts, namely a tangible (movable and immovable asset) and an intangible asset, and under the intangible asset concept the concepts power and reputation were modeled, which enable to model further threat consequences such as loss of reputation in the case of a data loss, (3) the idea of incorporating dynamic threat processes was dropped, since the threat impact on a given asset is statically derived regarding confidentiality, integrity, and/or availability, (4) since it is possible to model more granular organizational units such as departments under the existing organization concept, the current structure was kept, (5) although rating the information security awareness level of roles or persons could be useful in some cases, this attribute was not incorporated to ensure a clear, non-bloated ontology structure, (6) since threats such as lawsuits are not directly related to information security, those threats were not incorporated into the security ontology, and (7) the current focus is designing the ontological structure to support information security risk management and, therefore, the idea of incorporating business continuity standards was shelved.

After the elimination of the identified weaknesses, the security ontology was evaluated in a second evaluation round. Since the evaluation team did not find any further weaknesses, the evaluation process continues with the formulation of informal competency questions.

5.2 Informal Competency Questions

Since we want to support the entire information security risk management process the domain expert team developed competency questions according to the generically defined information security risk management phases (1) system characterization, (2) threat and vulnerability assessment, (3) risk determination, (4) control identification, and (5) control evaluation and implementation. The following competency questions define the requirements concerning the information security risk management support capabilities of the security ontology:

- Assuming threat T: (a) Which threats are generally relevant to information security? (b) Which threats are relevant to the considered organization? (c) Does the considered threat T have a deliberate or accidental threat source? (d) Does the considered threat T have a human or natural threat origin? (e) Which assets are threatened by threat T? (f) Which security attributes are affected by threat T? (g) Which vulnerabilities are exploited by threat T?
- Assuming vulnerability V: (a) Which controls can be used to mitigate vulnerability V? (b) What is the severity of vulnerability V?
- Assuming control C: (a) Which controls have already been implemented in the organization? (b) What is the physical location of the implemented controls? (c) To which standard control does control C correspond? (d) Which vulnerabilities could be mitigated by control C? (e) Which assets are required to implement control C? (f) How is control C classified in terms of its control type? (g) Up to what extent protect existing control implementations a given asset?
- Assuming threat probability TP: (a) Which threat probabilities increase if threat T becomes effective? (b) Which threats are relevant for the probability of threat T? (c) What is the a priori threat probability of threat T for a given location?
- Assuming impact I: (a) What is the impact I for a given threat T? (b) What is the impact I, if asset A goes down? (c) Which assets are required by asset A?
- Assuming control recommendation CR: (a) Which controls are recommended to mitigate a given threat? (b) Which controls are recommended to mitigate a given vulnerability? (c) Which controls are recommended to protect a given asset? (d) What are potential control implementation possibilities to mitigate a given threat? (e) Which controls are required to protect a given security attribute?

5.3 Formal Competency Questions

After the definition of informal competency questions by the information security professional evaluation team, these questions were transformed into formal competency questions to extract the corresponding result sets from the security ontology. Due to the different granularity level of the informal competency questions three different technologies have been utilized to extract the knowledge: (1) for questions which address inference issues the ontology reasoner Pellet 1.5.1[3] was used, (2) for those questions which incorporate the usage of instances SPARQL [41] was used to query the OWL implementation of the security ontology, and (3) since SPARQL has some limitations regarding its expressive power, the Protege OWL API[4] was used as well to encode

[3]Pellet: `http://pellet.owldl.com/`

[4]Protege OWL API: `http://protege.stanford.edu/plugins/owl/api/`

the formal competency questions in algorithmic structures. To enhance the readability some of the formal competency questions are represented as pseudo-code. Since a listing of all formal competency questions would go beyond the scope of this paper, this section presents only selected competency question implementations.

5.3.1 Does the considered threat has a human or natural threat origin?

Since the information whether a threat has a human or natural threat origin is encoded in the formal description of each threat concept, reasoning engines are able to extract the answer to the stated question.

The Protege ontology editor was used in combination with the Pellet ontology reasoner to classify those threats which have a human threat origin. To provide the reasoner with the necessary input the concept *sec:HumanThreat* (sec:HumanThreat $\sqsubseteq$ sec:Threat), including restriction $\exists$ sec:threatOrigin.sec:HumanThreatOrigin was created. According to the human threat concept specification the reasoner was thus able to extract the corresponding threat concepts (e.g. break in, data tampering, or fire fighting) to explicitly classify them as human threats. Threats with a natural threat origin were extracted by creating the concept *sec:NaturalThreat* (sec:NaturalThreat $\sqsubseteq$ sec:Threat), including restriction $\exists$ sec:threatOrigin.sec:NaturalThreatOrigin.

5.3.2 Which assets are threatened by a given threat?

Since the actual risk of a threat has to be determined, besides its probability also its potential impact is needed to be known. Algorithm 1 incorporates the OWL API and is used to determine the threatened assets of the considered organization. The *GetRelated(x, y)* function returns an array filled with the concepts which are connected to concept x via relation y (e.g. sec:Data for *GetRelated(sec:DataLoss, sec:threatens)*). The *GetInstances(x)* function returns an array filled with all instances located in concept x (e.g. Room0101 and Room0102 for *GetInstances(ent:Section)*). Finally, R returns an array of concrete assets, owned by the given organization and threatened by the given threat.

Algorithm 1 Assets threatened by a given threat

```
1: T ← given threat
2: O ← given organization
3: R ← null
4: RCL ← GetRelated(T, sec:threatens)
5: for i ← 0 to RCL.Length do
6:    A ← GetInstances(RCL[i])
7:    for i ← 0 to A.Length do
8:       if A[i].ent:ownedBy == O then
9:          R.Add(A[i])
10:      end if
11:   end for
12: end for
13: return R
```

5.3.3 Which vulnerabilities are exploited by a given threat and which controls can be used to mitigate the vulnerabilities?

If a threat is threatening crucial assets of the considered organization, it has to be known which of the existing vulnerabilities the threat exploits and how these vulnerabilities can be mitigated by appropriate controls to reduce the risk to an acceptable level. First of all, the subsequent SPARQL statement queries the vulnerabilities which are associated by relation *sec:exploits* with the asset loss threat. Note that the asset loss threat is just an example and that the vulnerabilities of each threat can be revealed in the same way.

```
SELECT ?vulnerability
WHERE {sec:AssetLoss sec:exploits ?vulnerability}
```

Since one vulnerability of asset loss is the unavailability of fallback equipment, the following query reveals the associated controls.

```
SELECT ?control
WHERE {sec:NoFallbackEquipmentAvailable
sec:mitigatedBy ?control}
```

With the appropriate control concept on hand, the organization is now able to derive the control implementation descriptions to mitigate the corresponding vulnerability in the context of a given asset.

5.3.4 Up to what extent protect existing control implementations a given asset?

This competency question is one of the most crucial ones in the information security risk management process, as it determines whether existing control implementation are adequate for a given asset or not. Algorithm 2 is used to determine which threats, vulnerabilities, controls, and existing control implementations are relevant to the considered asset. *GetThreats(x)* returns those threats which directly affect asset x. IC contains an array of existing control implementations protecting the considered asset. NC contains an array of those controls which are not implemented in the given organization in the context of the given asset. *Protects(x, y)* returns true if control implementation x protects asset y. Therefore, the function reveals concept z which is connected via the *sec:vulnerabilityOn* relation to the corresponding vulnerability of control implementation x. To protect asset y, control implementation x has to be implemented at concept z which is connected depending on the vulnerability type to asset y. If it is an administrative vulnerability, *sec:vulnerabilityOn* points at *ent:Organization* and asset y has to be connected via the *ent:ownedBy* relation to the considered organization z to be protected by control implementation x which is valid for organization z (e.g. a non-smoking policy). In the case of a technical vulnerability, *sec:vulnerabilityOn* could also point at *ent:Section* and asset y has to be connected via the *ent:locatedIn* relation to the considered section z to be protected by control implementation x which is also located in section z (e.g. an automatic fire extinguishing system).

5.4 Result Sets

By the implementation and the subsequent execution of the formal competency question set, each competency question resulted in a data set, which is evaluated by the security professional expert team in this evaluation step. For the evaluation of the security ontology, a fictive organizational model has been used as input, thus the evaluation team concentrated on the result set regarding the returned type of knowledge and not to its completeness. Due to the high degree of complexity, not all formal competency questions have been answered with simple ontology queries. Nevertheless, it could be shown that the ontology is able to answer such complex questions, even if an external calculation is required.

During the first evaluation round the evaluation team identified the following shortcomings: (1) assets provide no

Algorithm 2 Determining existing and missing control implementations

```
1: A ← given asset
2: IC ← null
3: NC ← null
4: TL ← GetThreats(A)
5: for i ← 0 to TL.Length do
6:   VL ← GetRelated(TL[i], sec:exploits)
7:   for j ← 0 to VL.Length do
8:     CL ← GetRelated(VL[j], sec:mitigatedBy)
9:     for k ← 0 to CL.Length do
10:      IL ← GetRelated(CL[k], sec:implementedBy)
11:      for l ← 0 to IL.Length do
12:        CI ← GetInstances(IL[l])
13:        if CI != null then
14:          for m ← 0 to CI.Length do
15:            if Protects(CI[m], A) == true then
16:              IC.Add(IL[l], 'implemented by:
                 '.CI[m])
17:            else
18:              NC.Add(IL[l], 'not implemented')
19:            end if
20:          end for
21:        else
22:          NC.Add(IL[l], 'not implemented')
23:        end if
24:      end for
25:    end for
26:  end for
27:  PTL ← GetRelated(TL[i], sec:canBeConsequenceOf)
28:  for n ← 0 to PTL.Length do
29:    Line 6 to Line 30 with PTL[n] for TL[i]
30:  end for
31: end for
32: return IC
33: return NC
```

information on the maximum tolerable period of disruption and recovery period objective, (2) the currently implemented three-point Likert scale may be insufficient for effectiveness or requirement ratings, and (3) there is no information on the complexity of control implementations.

These issues have been addressed as follows: (1) since an attribute for the importance of each resource has already been incorporated, the idea of implementing more specific attributes, such as the maximum tolerable period of disruption was dropped; instead the existing qualitative rating scale was refined by providing different scale definitions for confidentiality, integrity, and availability, (2) additional scales were incorporated which enable a more granular rating of effectiveness and requirements, and (3) the complexity of control implementations is implicitly given by the formal recommended control implementation representations and the associated asset costs.

6. CONCLUSION

Based on the analyzed risk management approaches, existing literature, and risk management specific requirements, the ontology comprises the concepts threat, vulnerability, and control to represent the information security domain knowledge. Besides these core concepts, also concepts and relations necessary to formally describe the organization and its assets have been incorporated. While the formal description of the core concepts is mainly based on hermeneutic research, the formal description of the non-core concepts relies also on already existing taxonomies such as the United Nations Standard Products and Services Code. The analyzed information security knowledge sources have been mapped to a great extent to the security ontology.

To enrich the knowledge model with concrete information security knowledge we analyzed several best-practice guidelines and information security standards regarding their acceptance, completeness, availability, and knowledge representation. Finally, the German IT Grundschutz Manual has been superimposed on the security ontology and more than 500 information security concepts and 600 corresponding formal axioms have been integrated into the ontological knowledge base. The main challenges at the knowledge integration have been the differences regarding both knowledge models and the inconsistent knowledge granularity of the German IT Grundschutz Manual.

To extend the existing knowledge model, we plan to map and integrate further information security standards, such as the French EBIOS. This will not only guarantee for an enriched set of threats, vulnerabilities and controls, but also opens new possibilities with regard to certification support. More and more companies carry out certification initiatives to gain a competitive advantage or to comply with legal regulations. With our knowledge model at hand, organizations not only have a formal reference representation to understand which controls have to be implemented to fulfill a specific certification standard, but also automated reasoning on the current certification status, based on the organizational model, is part of future research. Furthermore, we are currently working on novel tools which utilize the ontological knowledge base to facilitate automatic risk management support.

7. ACKNOWLEDGMENTS

This work was supported by grants of the Austrian Government's FIT-IT Research Initiative on Trust in IT Systems under the contract 813701 and was performed at the Research Center Secure Business Austria funded by the Federal Ministry of Economics and Labor of the Republic of Austria (BMWA) and the City of Vienna.

8. REFERENCES

[1] M. Aime, A. Atzeni, and P. Pomi. AMBRA: automated model-based risk analysis. In *QoP '07: Proceedings of the 2007 ACM workshop on Quality of protection*, pages 43–48, New York, NY, USA, 2007. ACM.

[2] M. Alavi and D. E. Leidner. Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, 25(1):107–136, 2001.

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

[4] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, January 2003.

[5] F. Baader, I. Horrocks, and U. Sattler. *Mechanizing Mathematical Reasoning*, volume 2605/2005 of *Lecture Notes in Computer Science*, chapter Description Logics as Ontology Languages for the Semantic Web, pages 228–248. Springer Berlin / Heidelberg, 2005.

[6] R. Baskerville. Information systems security design methods: Implications for information systems development. *ACM Computing Surveys*, 25(4):375–414, December 1993.

[7] M. Bishop. *Computer security - art and science*. Addison Wesley, 2003.

[8] M. Bishop. What is computer security? *IEEE Security and Privacy*, 1(1):67–69, 2003.

[9] J. Brank, M. Grobelnik, and D. Mladenić. A survey of ontology evaluation techniques. In *SIKDD 2005 at Multiconference IS 2005*, 2005.

[10] C. Brewster, H. Alani, S. Dasmahapatra, and Y. Wilks. Data driven ontology evaluation. In *International Conference on Language Resources and Evaluation*, 2004.

[11] BSI. IT Grundschutz Manual, 2004.

[12] V. K. Chaudhri, B. E. John, S. Mishra, J. Pacheco, B. Porter, and A. Spaulding. Enabling experts to build knowledge bases from science textbooks. In *K-CAP '07: Proceedings of the 4th international conference on Knowledge capture*, pages 159–166, New York, NY, USA, 2007. ACM.

[13] DCSSI. EBIOS - Section 2 - Approach. February 2004.

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

[15] ENISA. Risk management: implementation principles and inventories for risk management/risk assessment methods and tools. Technical report, European Network and Information Security Agency, June 2006.

[16] T. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Humam-Computer Studies*, 43(5-6):907–928, 1995.

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

[18] I. Horrocks, P. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[19] ISO/IEC. ISO/IEC 27001:2005, Information technology - Security techniques - Information security management systems - Requirements, 2005.

[20] P. Jrvinen. Research questions guiding selection of an appropriate research method. In *Proceedings of the 8th European Conference on Information Systems, Trends in Information and Communication Systems for the 21st Century, ECIS 2000, Vienna, Austria, July 3-5, 2000*, 2000.

[21] C. Jung, I. Han, and B. Suh. Risk analysis for electronic commerce using case-based reasoning. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 8:61–73, 1999.

[22] M. Karyda, T. Balopoulos, L. Gymnopoulos, S. Kokolakis, C. Lambrinoudakis, S. Gritzalis, and 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)*, pages 1033–1037, Washington, DC, USA, 2006. IEEE Computer Society.

[23] S. Kesh and P. Ratnasingam. A knowledge architecture for it security. *Communications of the ACM*, 50(7):103–108, 2007.

[24] A. Kim, J. Luo, and M. Kang. Security ontology for annotating resources. In *OTM Conferences (2)*, pages 1483–1499, 2005.

[25] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55, 1932.

[26] L. A. F. Martimiano and E. dos Santos Moreira. An OWL-based security incident ontology, 2005.

[27] D. J. McManus and C. A. Snyder. Synergy between data warehousing and knowledge management; three industries reviewed. *International Journal of Information Technology and Management*, 2(1-2):85–99, 2003.

[28] NIST. An Introduction to Computer Security - The NIST Handbook. Technical report, NIST (National Institute of Standards and Technology), October 1995. Special Publication 800-12.

[29] C. Patel, K. Supekar, Y. Lee, and E. Park. Ontokhoj: a semantic web portal for ontology searching, ranking and classification. In *WIDM '03: Proceedings of the 5th ACM international workshop on Web information and data management*, pages 58–61, New York, NY, USA, 2003. ACM Press.

[30] T. Peltier. *Information Security Risk Analysis*. Auerbach Publications, Boca Raton, Florida, 2001. ISBN: 0-8493-0880-1.

[31] PITAC. Cyber security: A crisis of prioritization - report to the president. Technical report, President's Information Technology Advisory Committee, February 2005.

[32] V. Raskin, C. F. Hempelmann, K. E. Triezenberg, and 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*, pages 53–59, New York, NY, USA, 2001. ACM Press.

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

[34] R. Shirey. RFC 2828 - internet security glossary, May 2000.

[35] S. Smith and E. Spafford. Grand challenges in information security: Process and output. *IEEE Security & Privacy*, 2(1):69–71, 2004.

[36] G. Stoneburner, A. Goguen, and A. Feringa. Risk management guide for information technology systems. NIST Special Publication 800-30, National Institute of Standards and Technology (NIST), Gaithersburg, MD 20899-8930, July 2002.

[37] D. Straub and R. Welke. Coping with systems risk: Security planning models for management decision making. *MIS Quarterly*, 22(4):441–469, December 1998.

[38] United Nations. United Nations Standard Products and Services Code, 2006.

[39] M. Uschold and M. Grninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.

[40] W3C. OWL - web ontology language, February 2004.

[41] W3C. SPARQL - query language for RDF, 2007.

[42] M. Whitman. Enemy at the gate: threats to information security. *Communications of the ACM*, 46(8):91–95, 2003.