Security Ontologies: Improving Quantitative Risk Analysis

Andreas Ekelhart*, Stefan Fenz*, Markus Klemen* and Edgar Weippl†
*Secure Business Austria — Security Research, Vienna, Austria
Email: {aekelhart, sfenz, mklemen}@securityresearch.at
†Vienna University of Technology, Vienna, Austria
Email: weippl@ifs.tuwien.ac.at

***Abstract*— IT-security has become a much diversified field and small and medium sized enterprises (SMEs), in particular, do not have the financial ability to implement a holistic IT-security approach. We thus propose a security ontology, to provide a solid base for an applicable and holistic IT-security approach for SMEs, enabling low-cost risk management and threat analysis. Based on the taxonomy of computer security and dependability by Landwehr [1], a heavy-weight ontology can be used to organize and systematically structure knowledge on threats, safeguards, and assets. Using this ontology, each threat scenario can be simulated with a different protection profile as to evaluate the effectiveness and the cost/benefit ratio of individual safeguards.**

I. Introduction

In the course of time, IT-security became a very diversified field of research. It is no longer limited to classical virus attacks; applied IT-security also has to consider physical attacks, random acts of nature, industrial espionage, etc. With the need to implement IT-security measures in world-wide corporate environments and the growing application scope, it becomes increasingly difficult for experts of different domains to understand each other and to use a precisely defined terminology. We thus need a security ontology to clarify the meaning and interdependence of IT-security relevant terms [2] which then can be used to facilitate qualitative risk analysis and decision processes. It seems not very efficient if an Asian employee is drafting a corporate-wide security policy, while his colleague in Russia is misinterpreting the policy, because the terms which were used are not explicitly defined. Some kind of agreed ontology can be used to avoid such inefficiencies.
The term 'ontology' can be defined in many different ways; the following one specifies the term adequately for our research activities.

'An ontology defines the basic terms and relations compromising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary.' [3]

Furthermore we distinguish lightweight and heavyweight ontologies. Lightweight ontologies include concepts, concept taxonomies, relationships between concepts, and properties that describe concepts [3]. Heavyweight ontologies add axioms and constraints to lightweight ontologies [3].
The security ontology is based on Landwehr's [1] [4] taxonomy; we extended it to form a heavyweight ontology. Landwehr's security and dependability classification [1] was enriched by domain specific concepts and attributes to incorporate enterprise infrastructure and role schemes.
The main goal of our current research activities is to provide a security ontology framework which unifies existing approaches like [5], [1], [4], and [6] to support IT-security risk analysis. The ontology 'knows' which threats endanger which assets and which countermeasures could lower the probability of occurrence, the potential loss or the speed of propagation for cascading failures. In addition, each infrastructure object and each countermeasure in the ontology can be annotated with various types of costs as well as benefits. By comparing various scenarios during a quantitative risk analysis, companies can choose their individual safeguard package which is based on both, a common security framework and the companys' security policy.

II. The Problem

Security is a crucial part for every company, but the approaches to security, including evaluation and implementation of safeguards, vary widely. All too often wrong decisions are made; they are caused by insufficient knowledge about the security domain, threats, possible countermeasures and the company infrastructure. There are two major reasons for this: First, security terminology is vaguely defined; this leads to confusion among experts as well as the people who should be counseled and served [2]. Without a shared terminology, especially in a complex domain like security, communication cannot be successful and is prone to errors. Ontologies are a viable solution because they allow not only the definition of terms but also define their relationship to each other. Reasoning on the generated knowledge opens further possibilities. However, at the moment ontologies are still not widely used in commercial applications.
Second, decisions are often made by managers who do not understand the depth and complexity of the underlying IT-infrastructure and therefore base their decisions more on intuition than on a thorough cost/benefit analysis. IT-security personnel are often not involved in the decision making process, and if they are, they have a hard time explaining the complex situation to the decision makers in a proper way.

Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS'07)
0-7695-2755-8/07 $20.00 © 2007 IEEE
1530-1605/07 $20.00 © 2007 IEEE

Third, the development of adequate security concepts and plans requires a thorough threat analysis. Most companies decide to base their analysis on existing security frameworks. Various IT-standards exist: the Baseline Protection Manual [5] for example is a comprehensive and well-developed approach to security. Companies can get certified if they meet a specified baseline. Drawbacks of this standard are the complexity (approximately 3000 pages) and insufficient risk analysis support - quantitative risk evaluation is not addressed. CobiT [7], an IT governance framework based on best practices, is complex to use and does not address security threats and safeguards in detail. The ISO 17799 standard [8] includes risk analysis and benchmarking but addresses security from a management perspective without going into operational details. All these frameworks lack in providing clear and simple visualization as well as a universal simulation environment which would allow managers to try different scenarios within the chosen framework.

Every security decision must consider the specific company environment. The employee responsible for security is often not aware of all relevant details of the infrastructure. To obtain the necessary information by interviewing colleagues or studying plans is time consuming and thus the risk analysis is usually based on an incomplete picture.

At a certain point the decision has to be made in favor of a specific measure. Time and monetary restraints are the key factors for decision making but it is a non-trivial task to understand costs and the consequences of various alternatives. We need an approach which helps to eliminate these problems and allows to simulate threats to corporate assets, taking the entire infrastructure into account. The damage over time and costs of affected assets should be visualized. Using this approach effective countermeasures and their costs can be calculated quickly and a subsequent decision will be based on objective criteria. In an additional run of the simulation, the benefits of the chosen countermeasures can be seen.

III. Security Ontology Framework

Our security ontology framework consists of four parts. The first part is based on the security and dependability taxonomy by Landwehr [1], the second part presents the underlying risk analysis methodology, the third part describes concepts of the (IT) infrastructure domain and the fourth part provides a simulation enabling enterprises to analyze various policy scenarios.

The ontology is coded in OWL (Web Ontology Language [9]) and the Protege Ontology Editor [10] was used to edit and visualize the ontology and its corresponding instances.

The following subsections describe the parts in more detail:

A. Security and dependability taxonomy

Figure 1 shows the *security and dependability taxonomy's* concept structure; for further information the paper [1] provides a detailed description. As Figure 1 shows, the taxonomy is designed in a very generic way, and so it may easily be extended with additional concepts. To populate

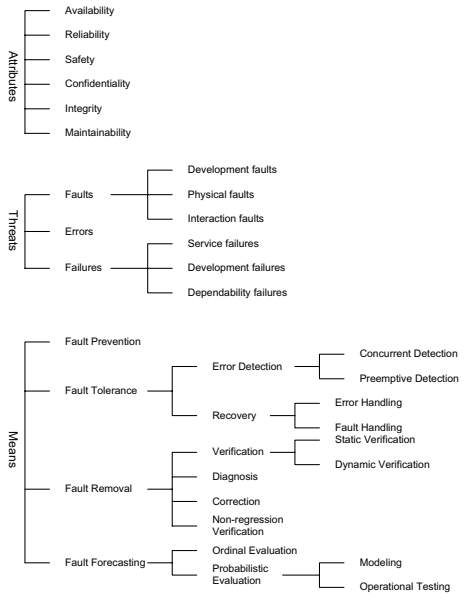

Fig. 1. Security and Dependability Taxonomy [1]

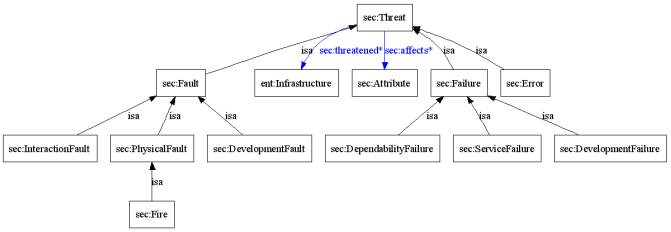

Fig. 2. Sub-tree 'sec:Threat'

the taxonomy, instances and dependencies were inserted. Figure 2 represents the sub-tree 'sec:Threat', which is part of the security ontology and was derived from Landwehr's taxonomy [1]. The sub-concept 'sec:Fire' was inserted as the first real concept; it is classified as a physical fault which belongs to the super-concept 'sec:Threat'. With 'sec:threatened' and 'sec:affects' the first dependencies were inserted; 'sec:threatened' describes that every instance of any 'sec:Threat' sub-concept threatens all instances of any 'ent:Infrastructure' sub-class. 'sec:affects' describes the fact, that every instance of a 'sec:Threat' sub-concept affects one or more instances of the 'sec:Attribute' concept.

To provide useful knowledge for simulating threats to corporate assets, the ontology has yet to be extended with additional concepts describing the (IT) infrastructure and personnel structure.

B. Quantitative Risk Analysis

The obvious first step prior to installing any security safeguards is to perform a security risk analysis such as described in Peltier [11] or Pipkin [12]. The basic idea is to first enumerate all assets including their values and to determine threats to them. The second step is to estimate the

probability that a threat will occur and the damage it will cause. Based on these estimates the top risks can be identified. Threat Modeling [13], [14] takes a slightly different approach. Starting with a complex threat (e.g. fraudulent payment with a credit card) possible attack paths are identified (e.g. forging a credit card, stealing the card) and are recursively refined.

Qualitative approaches to security risk assessment use classes such as high, medium, low of probabilities and damages; in quantitative risk analysis precise numeric values are needed.
A simple measure to determine the financial risk is the Annual Loss Expectancy (ALE).

$$ALE = \sum_{i=1}^{n} I(O_i)F_i$$

$\{O_1, O_2, \ldots, O_n\}$	set of harmful outcomes,
$I(O_i)$	impact of outcome i in USD,
$F_i\,\{O_1, O_2, \ldots, O_n\}$	frequency of outcome i.

More generally, risk can be formally defined as a set O of outcomes and the likelihood L of their occurrence [15].

$$Risk \equiv \{(L_1, O_1), \ldots, (L_i, O_i), \ldots, (L_n, O_n)\}$$

Raiffa [16] laid the fundamentals on decision theory which were refined by Howard in 1966 [17]. Soo Hoo [18] provides a definition: *At its core, decision analysis is a reductionist modeling approach that dissects decision problems into constituent parts: decisions to be made, uncertainties that make decisions difficult, and preferences used to value outcomes.*

Implementing safeguards will most likely decrease either the probability or the damage of threats. A very simple model compares the reduction of the ALE with the cost for each safeguard. While this model is easy to understand and use it has obvious shortcomings. (1) Some safeguards will be more efficient if employed together. For instance file system encryption is best combined with BIOS passwords to prevent booting other operating systems. (2) Other safeguards lose efficiency when combined because they are substitutes: for instance, a packet and an application-level firewall. (3) Yet other safeguards cannot be combined at all. Two desktop virus scanners scanners cannot be installed on one computer because they interfere with each other, usually rendering the entire system instable. Soo Hoo [18] addresses these issues in his model. Safeguards are combined to packages or policies. For a specific policy each safeguard is either used or not used. The following table contains all the definitions required to understand the model.

B_i	Bad event, $i = \{1, 2, 3, \ldots, n\}$
S_j	Safeguard, $j = \{1, 2, 3, \ldots, m\}$
$I_k(S_j)$	Binary function, that returns 1 if safeguard S_j is included in policy P_k, else 0
$F_0(B_i)$	Frequency of occurrence of the bad event B_i with no safeguards in place
$D_0(B_i)$	Damage of the bad event B_i with no safeguards in place
$E_f(B_i, S_j)$	% reduction of the frequency of occurrence of B_i with S_j in place
$E_d(B_i, S_j)$	% reduction of the damage of B_i with S_j in place
ALE_k	Annual Loss Expectancy with policy P_k

$$\text{Benefit}_k = ALE_0 - ALE_k,\ \forall k : k \in \{1, 2, 3, \ldots\}$$

$$ALE_k = \sum_{i=1}^{n} \Big\{ F_0(B_i)D_0(B_i) \prod_{j=1}^{m} [(1 - E_f(B_i, S_j)I_k(S_j)) \\ (1 - E_d(B_i, S_j)I_k(S_j))] \Big\}$$

Soo Hoo's approach to the annual loss expectancy calculation provides — in theory — a proper framework. Using it in real world applications brings up some problems regarding the concrete calculation of the ALE. It is very difficult to estimate the % reduction of the frequency of occurance ($E_f(B_i, S_j)$) and the % reduction of the damage ($E_d(B_i, S_j)$). In most cases these numbers are based on expensive expert knowledge which is not availiable to small and medium sized enterprises.

Thus we implement a security ontology which is capable of calculating realistic values for damage and frequency of occurrence. In the first step we develop a model to calculate the damage of a linear spreading threat such as fire. The model assumes that the threat starts in a certain place (e.g. room 0202) and spreads linearly. The damage calculation per room is based on the following model:

$$D_{r,n}(B_i) = \sum_{i=1}^{n} \frac{t}{100} * A$$

B_i	Bad event, $i = \{1, 2, 3, \ldots, n\}$
$D_{r,n}(B_i)$	Damage of Room r in time n for bad event B_i
n	Elapsed time; $n \leq t$
t	Point in time of full asset damage
A	Value of all assets located in a certain room

If e.g., one room is burned down (in time t), the fire spreads to the next room and so the total damage increases over time. Horizontal and vertical spreading is considered. Without any safeguards in place for the currently considered

room, the elapsed time will equal the point in time of full asset damage. Concrete values for the elapsed time are gained from the security ontology. After every bordering room is processed we obtain the total damage sum over all rooms:

$$DA(B_i) = \sum_{i=1}^{r} D_{r,n}(B_i)$$

$DA(B_i)$	Damage of all rooms without safeguard damage for bad event B_i
$D_{r,n}(B_i)$	Damage of Room r in time n for bad event B_i
r	Current room

Ignoring safeguards for the damage calculation of each room, $DA(B_i)$ corresponds to $D_0(B_i)$ (Damage of the bad event B_i with no safeguards in place) in Soo Hoo's model with the difference that $DA(B_i)$ is calculated by taking the actual infrastructure on a very detailed level into account. The information is taken from the corresponding ontology instances (see Section III-D).

Safeguards, which are activated by corresponding sensors, can be applied in our model. The calculation of the ALE with safeguards happens dynamically: if a threat is detected by a sensor the corresponding safeguard is started; the properties of each safeguard are described in the ontology. E.g. the fire and its further spreading can be stopped by a distinguisher within a certain timeframe. Not only the threat itself is causing damage, also safeguards (e.g. active water extinguisher in computer labs) can cause an enormous damage. We considered this circumstance with a safeguard damage parameter which describes the damaging factor of an activated safeguard:

$$DS_r(B_i) = D_{r,n}(B_i) + (A - D_{r,n}(B_i)) * SD_i * I_S$$

$DS_r(B_i)$	Total room damage with safeguards in place for room r for bad event B_i
A	Value of all assets located in a certain room
$D_{r,n}(B_i)$	Damage of Room r in time n for bad event B_i
SD_i	Damage parameter of safeguard i
I_S	Binary parameter: = 0 if $n <$ than safeguard reaction time = 1 if $n \geq$ than safeguard reaction time

After calculating the damage caused by a certain threat, we need the corresponding probability of occurrence to be able to calculate the annual loss expectancy. Our security ontology provides the user with threat specific probabilities of occurrence. The probabilities depend on certain circumstances (e.g. the probability of a fire will be less if there is a ban on smoking) and can be adapted by the user if needed.
Now we are able to calculate the ALE, which is based on a very detailed and company-specific data material:

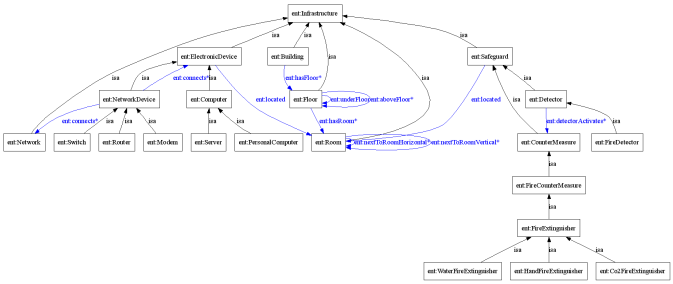

Fig. 3. Sub-tree 'ent:Infrastructure'

C. Ontology maintenance

The goal of this approach is to allow a detailed quantitative risk analysis, without requiring too much expertise and time. Experts are only needed for the framework creation (at the first two levels). Three levels exist:

1) Defining the concepts of the ontology (including infrastructure, personnel, roles and disasters concepts), must be obviously done by experts.
2) Based on these concepts domain experts have to provide individuals (e.g. disasters, safeguards) and their attributes (e.g. spread time, damage, probability of occurrence).
3) At this stage a solid framework exists which can be utilized by companies. Their task is it to model their company. This process will be done by the IT administrator or in further versions automatically based on building plans and the information extracted by automatic IT infrastructure exploring tools. Minor customization and extension can be done by company staff or external experts.

Filling the ontology is supported by user friendly and intuitive forms. The application itself, which accesses the ontology knowledge and executes the simulation to calculate risk management ratios, is provided by us.

D. Infrastructure concepts

Figure 3 shows the infrastructure part of the security ontology. The building with its corresponding floors and rooms can be described using the infrastructure framework. To map the entire building plan exactly to the security ontology, each room is described by its position within the building. The ontology 'knows' in which building and

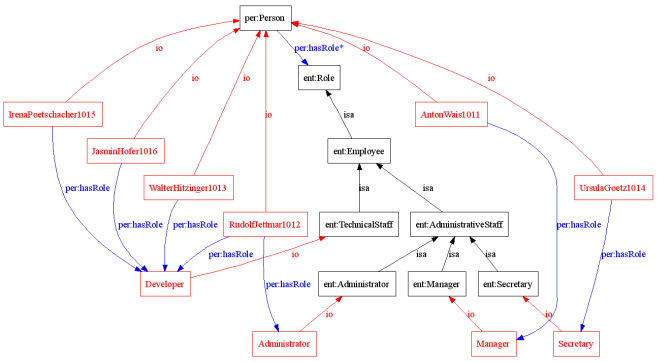

Fig. 4. Sub-tree 'per:Person' and 'ent:Role'

on which floor a certain room is located. The attributes 'ent:nextToRoomHorizontal' and 'ent:nextToRoomVertical' describe the exact location of each room. Each instance of 'ent:ElectronicDevice' and 'ent:Safguard' is located in a certain room. Of course a room can contain more concepts. The current ontology uses a flexible and easily extendable structure; additional concepts can be included without effort. The concept 'ent:Safeguard' is subdivided into 'ent:CounterMeasure' and
'ent:Detector', which are used to model detectors (fire, smoke, noise, etc.) and its corresponding countermeasures (fire extinguisher, alarm system, etc.).
Beside 'ent:Infrastructure', the concepts 'ent:Role' and 'per:Person' ensure that both technical and personnel structures can be mapped to the current ontology.

E. Person and role concepts

The concept 'per:Person' enables the ontology to map natural persons. Figure 4 represents the role concept for assigning certain roles to natural persons. Several instances of 'per:Person' were created to assign different roles to them. The current ontology considers only sub-concepts of 'ent:Employee'; additional roles can be added easily, if needed.

After describing the security ontology, the next section presents a practical example and makes the benefit for corporations clear.

IV. Example

In this section we provide a simplified example of how a company would use the aforementioned security ontology to model an IT infrastructure. It must be stressed out that the underlying cost models are kept simple and some factors, such as additional revenues which may be generated by implementing new safeguards, are left aside in order to maintain clarity.

A. The Company

For the purpose of clarity, we chose a small company with six employees. The company is set on two floors (1st and 2nd floor) of a 5-floor building in the center of a small town. On the first floor (Figure 5), there is one office, a storage

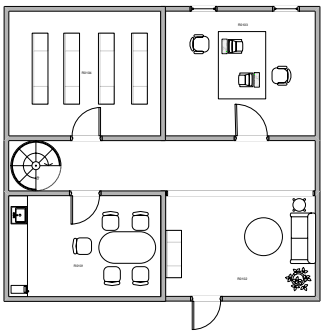

Fig. 5. Floor I

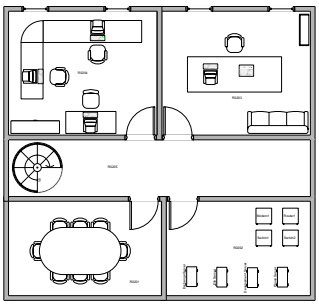

Fig. 6. Floor II

room and a lunchroom. The server room, a meeting room and two more offices are located on the second floor (see Figure 6).
The following listing shows the allocation of relevant (IT) infrastructure elements:

- First floor - Office room (R0103): 2 PC's
- First floor - Storage room (R0104): data media (archived)
- Second floor - Server room (R0202): 4 Servers, 1 Router, 2 Switches, 1 Modem
- Second floor - Office room (R0203): 1 PC, 1 Notebook
- Second floor - Office room (R0204): 3 PC's

The infrastructure is mapped on the sub-tree 'ent:Infrastructure' (compare Figure 3). The following listing gives an example for an OWL definition which describes a certain PC with its attributes:

```
<ent:PersonalComputer rdf:ID="Pc4">
 <ent:deliveryTime rdf:datatype="http://www.w3.org/2001/
 XMLSchema#int">3</ent:deliveryTime>
 <ent:assetCost rdf:datatype="http://www.w3.org/2001/
 XMLSchema#int">1500</ent:assetCost>
 <ent:outageCost rdf:datatype="http://www.w3.org/2001/
 XMLSchema#int">0</ent:outageCost>
 <ent:antiVirus rdf:datatype="http://www.w3.org/2001/
 XMLSchema#boolean">false</ent:antiVirus>
 <ent:hasOs rdf:datatype="http://www.w3.org/2001/
 XMLSchema#string">WinXpPro</ent:hasOs>
 <ent:located rdf:resource="#R0204"/>
</ent:PersonalComputer>
```

The concept 'ent:PersonalComputer' with its concrete instance 'Pc4' has the attributes 'ent:deliveryTime', 'ent:assetCost',

'ent:outageCost', 'ent:antiVirus', 'ent:hasOs and 'ent:located'. If this or any other instance will be destroyed by a certain disaster, the ontology 'knows' how long it takes to get a new one, how much it costs, where it is located and how much the outage costs per day. Apart from 'ent:antiVirus' and 'ent:hasOs', all attributes are inherited from super-concept 'ent:Infrastructure'.

B. The Disaster

After describing the company with its infrastructure, the current subsection defines the disaster, which will hit our software company.
The event of fire, as a physical threat scenario, was chosen. The simulation should show the amount of damage in the course of time and in consideration of the fire source. A certain room can be defined as the fire source; the speed of propagation without any countermeasures will be 5 minutes per floor and 5 minutes per room. Every infrastructure element is assigned to a certain room. In the case of fire all infrastructure elements within a room will be destroyed completely. The outage costs per room correspond to the outage costs sum of all destroyed elements, which are located in the room. It is possible to assign countermeasures to any room. These safeguards can lower the probability of occurrence and the speed of propagation in the case of fire.
The following OWL code-snippet shows an example of the countermeasure element 'ent:WaterFireExtinguisher':

```
<ent:WaterFireExtinguisher
 rdf:ID="WaterFireExtinguisher0102">
 <ent:located rdf:resource="#R0102"/>
 <ent:assetCost rdf:datatype="http://www.w3.org/2001/
 XMLSchema#int">500</ent:assetCost>
 <ent:deliveryTime rdf:datatype="http://www.w3.org/2001/
 XMLSchema#int">5</ent:deliveryTime>
 <ent:startTime rdf:datatype="http://www.w3.org/2001/
 XMLSchema#float">0.0</ent:startTime>
 <ent:extinguishingTimeRoom rdf:datatype="http://
 www.w3.org/2001/XMLSchema#int">1
 </ent:extinguishingTimeRoom>
</ent:WaterFireExtinguisher>
```

We can see that this extinguisher is located in room R0102 and will start, when switched on, immediately. Instance 'WaterFireExtinguisher0102' will extinguish the room within one minute. The attribute 'ent:startTime' is important for countermeasures which are not activated automatically (e.g. hand fire extinguisher).

C. The Simulation

The framework for our threat analysis has been explained in the preceding sections, now we present a tool called 'SecOntManager' which processes the ontology knowledge to simulate threats. This prototype handles IT costs and poses as a proof of concept. Further threat effects as well as infrastructure components can be added easily due to the generic structure.
In our example the management wants to know what impact fire would have on the infrastructure, what countermeasures exist and what their benefits are. For this purpose we show two program runs, one against the unprotected company and another including safeguards.

- The first program run without countermeasures: 'SecOntManager' offers an intuitive graphical user interface, shown in Figure 7. A threat and a corresponding starting point have to be chosen before a simulation can be started. We decide for the threat type fire and the server room (room0202) as the origin of fire. The program run produces a detailed log file which shows how the fire spreads from room to room and the damage it causes. An abridgment of this file can be seen in the following listing:

```
Current Room: <http://secont.com/secont.rdf#R0203>
        http://secont.com/secont.rdf#Pc7: 0
        used by Person:
         http://secont.com/secont.rdf#AntonWais1011
        Salary: 3000
        Total outage costs of infrastructure
         component / 5 min: 6
        Total damage costs of infrastructure
         component: 2000
        Recovery time and costs: 4 days: 2000
        ...
Search Detectors:
        Detector found: /
        Countermeasure activated: /
```

  Each room is processed completely before neighboring rooms are searched and at the end of the simulation all occurring costs are visualized in a line chart (see Figure 7). The time axis unit is set to minutes. Four curves, reflecting different cost categories, exist: the blue curve visualizes the damage. In the example the damage costs rise very fast due to the speed of fire - within 30 minutes every room was destroyed. By zooming in, displaying only the first 30 minutes, we can see how the damage evolves. After every room is set to condition 'burned down' no further damage can occur. The red line represents the outage costs, taken from assigned outage costs of infrastructure components and employee's costs. Outage costs rise constantly in the simulation until recovery. The green curve shows recovery costs. Replacement and setup times for destroyed components are taken into consideration. When components are available, connected outage costs decrease, visually spoken, the red line flattens. Additional installation costs lift the recovery costs upon damage. When every component is recovered, the pre-threat state is reached and outage costs do not rise anymore. Furthermore the total of all costs is reflected by the yellow curve. Fire costs amount to 79.634€ and it takes at least five days to recover from the damage.

- Second program run with countermeasures enabled: We now concentrate on reducing the damage by installing safeguards. SecOntManager offers to install fire suppression systems in the building. We decide for pre-action pipes in the entire building. Necessary detectors and fire extinguishers are added to rooms in the OWL file.

Fig. 7. SecOntManager: No countermeasures

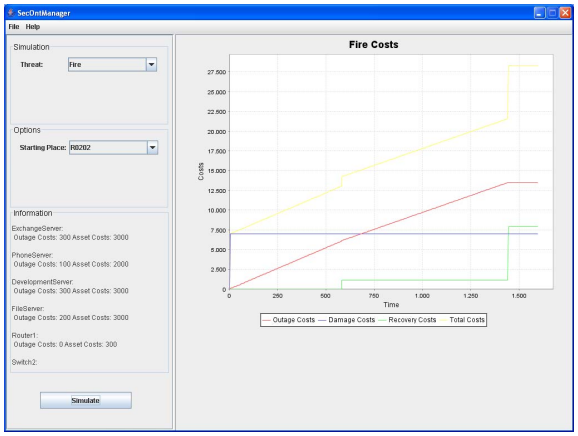

Fig. 8. SecOntManager: Scenario with pre-action water pipe installation

Their costs amount to 7.200€. Running the simulation produces the cost chart in Figure 8. As can be seen the total damage decreased drastically to 28.329€. After installing safeguards, the fire can not spread anymore; it is detected and extinguished shortly after breakout. Nevertheless costs and recovery times are still very high. The reason is that water extinguishers have a high damage factor concerning electronic devices and we have chosen the server room as place of fire origin. SecOntManager also offers CO2 fire extinguishers for locations with high electronically damages. Replacing the water extinguisher by a more expensive C02 extinguisher the total costs are reduced to 10.934€, which are mostly outage costs of one server which caused the fire. By adding a redundant server the outage time and costs could be cut to zero.

V. Conclusion

Increasingly, businesses require accurate security concepts and plans to protect themselves and their clients against various threats, including physical attacks, acts of nature beyond of human control and industrial espionage. Establishing an all-encompassing IT-security concept demands in-depth knowledge of existing threats, the company infrastructure and possible countermeasures. We propose an ontology-based approach to model companies combining security- with business-domain knowledge. The ontology guarantees a shared and accurate terminology — and when using OWL to represent it also guarantees portability. Knowledge of threats and corresponding countermeasures are integrated into the ontology framework. Moreover, we implemented a prototype capable of simulating threats against the modeled company by processing the knowledge contained in the ontology. 'SecOntManager' visualizes the damage caused by specific threats, outage costs and the recovery time. Running the program with added safeguards shows their benefits and offers objective data for decision making which safeguards to implement and to avoid installing countermeasures that are not cost-effective. An enhanced prototype with advanced risk analysis support will take failure probability into account, and will be developed in pilot installations with partner companies.

Acknowledgements

This work was performed at Secure Business Austria, a competence center that is funded by the Austrian Federal Ministry of Economics and Labor (BMWA) as well as by the provincial government of Vienna.

References

[1] A. Avizienis, J.-C. Laprie, B. Randell, and C. E. Landwehr, "Basic concepts and taxonomy of dependable and secure computing." *IEEE Trans. Dependable Sec. Comput.*, vol. 1, no. 1, pp. 11–33, 2004.

[2] M. Donner, "Toward a security ontology," *IEEE Security and Privacy*, vol. 1, no. 3, pp. 6–7, May/June 2003. [Online]. Available: http://dlib.computer.org/sp/books/sp2003/pdf/j3006.pdf

[3] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, *Ontological Engineering*, 1st ed. London: Springer, 2004.

[4] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, "A taxonomy of computer program security flaws," *ACM Comput. Surv.*, vol. 26, no. 3, pp. 211–254, 1994.

[5] "It-grundschutzhandbuch," http://www.bsi.de/gshb/deutsch/download/GSHB2004.pdf, 2004.

[6] "eclass," http://www.eclass.de/, 2006.

[7] "Cobit," http://www.isaca.org/, 2006.

[8] "Iso17799," http://www.iso.org/, 2006.

[9] "Owl web ontology language," http://www.w3.org/TR/owl-features/, 2004.

[10] "The protege ontology editor and knowledge acquisition system," http://protege.stanford.edu/, 2005.

[11] T. R. Peltier, *Information Security Risk Analysis Boca Raton*. Boca Raton, Florida: Auerbach Publications, 2001.

[12] D. Pipkin, *Information Security*. Prentice Hall, 2000.

[13] M. Howard and D. LeBlanc, *Writing Secure Code*, 2nd ed. Microsoft Press, 2002.

[14] F. Swiderski and W. Snyder, *Threat Modeling*. Microsoft Press, 2004, iSBN 0735619913.

[15] H. Kumamoto and E. Henley, *Probabilistic Risk Assessment and Management for Engineers and Scientists*, 2nd ed. New York: Institute of Electrical and Electronics Engineers, Inc, 1996.

[16] H. Raiffa and R. Schlaifer, "Applied statistical decision theory," Harvard University, Boston, 1961.

[17] R. Howard, "Decision analysis: Applied decision theory," in *Proceedings of the Fourth International Conference on Operational Research*, D. Hertz and J. Melese, Eds. Wiley-Interscience, 1966, pp. 55–71.

[18] K. J. S. Hoo, "How much is enough? a risk-management approach to computer security, working paper," Consortium for Research on Information Security and Policy (CRISP), June 2000, http://iis-db.stanford.edu/pubs/11900/soohoo.pdf. [Online]. Available: http://iis-db.stanford.edu/pubs/11900/soohoo.pdf