Specifying and Verifying Meta-Security by Means of Semantic Web Methods

Joaquín Borrego-Díaz[1], Antonia M. Chávez-González[1], José Luis Pro-Martín[2], and Virginia Matos-Arana[1]

[1] Dept. of Computer Science and Artificial Intelligence – University of Seville, Spain
{jborrego,tchavez}@us.es, vma_1990@hotmail.com
[2] Modinem S.L., Seville, Spain
jlpro@modinem.com

Abstract. In order to achieve a systematic treatment of security protocols, organizations release a number of technical briefings for describing how security incidents have to be managed. These documents can suffer semantic deficiencies, mainly due to ambiguity or different granularity levels of description and analysis. Ontological Engineering (OE) is a powerful instrument that can be applied for both, cleaning methods and knowledge in incident protocols, and specifying (meta)security requirements on protocols for solving security incidents. We also show how the ontology built from security reports can be used as the knowledge core for semantic systems in order to work with resolution incidents in a safe way. The method has been illustrated with a case study

1 Introduction

A key dimension in Security for Information Systems (SIS) is the document generation and management. Reports on incidents, protocols and information on systems play a structural role in the SIS paradigm. The uniform view of SIS in an organization provides robust strategies and secure solving methods. However, as it is said in [10], currently the reports generally describe information security policies by a mix of *professional opinion*, staff experience, technology manufacturer advice and external security standards or regulations.

It could happen these reports are useful only for members of the organizations (which share the same implicit knowledge about this) or new paradigms forces them to conciliate management methods.

SIS has evolved from a technical discipline to a strategic concept. The world's growing dependence on a powerful but vulnerable Internet – combined with the disruptive capabilities of cyber attackers – now threatens national and international security. In [6] the influence factors in this particular case of security incidents is summarized, showing the complexity and hardness of the problem. The potentially vast number of disparate information sources makes their management complex and time-consuming (see also [10]). Although such knowledge

⋆ Partially supported by *Excellence project* TIC-6064 of *Junta de Andalucía*, co-financed with FEDER founds.

may be consolidated by individual organizations, it is typically kept "in-house" and the interoperability among different organizations could be a challenge.

Semantic Web Technologies (SWT) can provide a unified view to solve the above-mentioned problems. On the one hand, the attempt to formalize the information described in the reports allows to emerge the knowledge. On the other hand, SWT naturally solve interoperability problems. That is, the consensus effort to represent document knowledge by means of ontologies and data forces the engineer to achieve the sound understanding of ideas, represented by means of concepts, properties and axioms of the ontology. Thus the problem of understanding the structure of concepts to anticipate potential failures may be solved by the combined work of Knowledge engineers and security experts.

Ontological Engineering provides tools to analyze important features as consistency, compliance with current Security Standards, and fidelity to the intended model [1]. The latter is about the sound representation of some concepts, this means, whether the specification represents the intentions of security experts and there are not axioms nor properties clearly incompatible with real concepts. Therefore, the ontology-based approach enables the definition of security concepts and their dependencies in an understandable way for both, humans and software agents [11]. Beside consistency and complexity, the absence of representational anomalies is mandatory [1].

In this paper we focus the interest in reports on incident protocols and security requirements. It has been selected as running example the document set published by Spanish INTECO–CERT institution[1]: *INTECO's identification and report of security incident for strategic operators* [3] and *The operator console. A Basic Guide to Critical Infrastructure Protection* [4]. The first one aims to be a guide intended to serve as a manual for action reporting and management related to Critical Infrastructure and Strategic Operators incidents through the INTECO-CERT. The second one describes the actions that operators have to perform in order to provide an effective and efficient response to security incidents. The documents provide a standardized protocol for both, effectively solve and document security incidents in a SIS scenario.

Aim of the paper. The aim is to show how to use SWT to analyze and repair security reports. It is based on the construction of an ontology from information contained in the documents, showing how the construction of the ontology itself allows to detect potential conflicts in protocols, documentation and classification.

2 Semantic Features of Security Documentation

A detailed analysis of the SIS documents must be performed from different points of view. It is necessary to distinguish between classification (identification of incidents) of SIS elements and the description level of security protocols (for reporting or solving incidents). The representation of these features should

[1] Acronym of spanish Incident Response Center Security
`http://www.inteco.es/home/national_communications_technology_institute/`

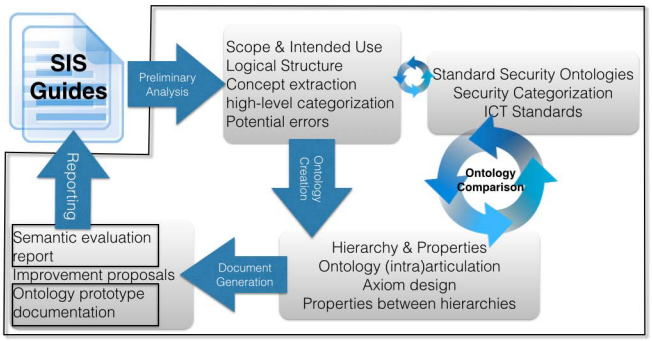

Fig. 1. Strategy applied to SIS documents

to provide essential elements (classes and particular individuals) for the ontology. The modular nature of the ontology should allow to extend aor modify these elements without a general reconsideration of ontological commitments. To achieve this modularity, the top levels of the ontology have to conciliate both points of view, whilst low level classes will represent a set of particular elements (usable actions, specific protocols, a set of possible identifications and classifications, etc.). Identification and protocol descriptions have different ontological nature although they share some common features which allow to articulate the ontology in two sub-hierarchies.

It would possible to specify identification and resolution protocols by means of standard service ontologies (e.g. OWL-S or WSMO). In this case a specific flowchart-based ontological description of protocols is selected. The reasons for this choice are justified by the particular features of SIS:

• Description (at operator level) is simpler than standard service ontologies.
• The representation of protocols is very similar to their natural (graphical) descriptions in documents, making them easily understandable.
• It provides a concise semantic description of the protocols which does not add complexity to reasoning services.
• Because of natural mapping between actions and ontological elements, the addition of new actions/description elements does not require SWT experts.

2.1 Strategy for Knowledge Recovery and Representation

The strategy for ontology extraction consists of several stages (see Fig. 1):

1. Preliminary analysis
   - To state the scope and intended use of SIS document.
   - Document analysis. Ontology engineers analyze the logical structure of the document and isolate main concepts used within.
   - To determine the ontological nature of different concepts. Elaboration of a first categorization (possibly by building several hierarchies).

- To find potential ambiguities or deficiencies in elements to be included in the ontology.

2. Ontology creation:
   - Hierarchies and properties implementation. Ontology articulation.
   - Design of axioms (classes specification) for the key concepts.
   - Study of relationships between the former subhierarchies.
3. Comparison of different (sub)ontologies with standard security ontologies.
4. Semantic evaluation report (with improvement proposals).

Each step requires some discussion on the features of critical concepts. The applicability of the ontology as semantic reference of future SIS systems has to be taken into account. Due to the lack of space, only the main steps are described in the paper, specially those where ontological analysis is relevant.

2.2 Representability of Security Issues

The proposed bottom-up approach is the natural choice because it is not intended to build a (other) security ontology. It aims to build an implicit ontology hidden in report documentation within an organization. The other approach, the adoption of a pre-existent security ontology to formalize and clarify the SIS documentation, does not seem a sound approach for these goals: Such an ontology usually describes an approach to SIS report/classification that can be incompatible with the implicit knowledge in the concrete organization. It have usually been built on security information resources, and, since these kind of resources have not been designed to fit ontological structures, several deficiencies of representation arise. In [5], authors detect a number of representational problems when enriching a security ontology with Information security:

P1: No concepts for some kind of vulnerabilities
P2: Vague connections between threats and controls
P3: No relationships between threats
P4: Inconsistent granularity of information
P5: Redundancy and overlapping of information

The bottom-up extraction of the ontology aid to solve most of the abovementioned problems for a particular organization (problems P1,P2,P4,P5) while problem P3 rests explicit posed (to be solved by SIS experts). It is worthy to note that the adaptation of a general security ontology for this task is hard to automate, because some criteria for revision cannot be fully formalized.

3 Strategy for Incident Report and Identification (IRD)

This section is devoted to comment the main conclusions of the application of the above described strategy to IRT documents [3,4].

Phases of incident response: According to [3], the description of the main phases in incident response and mitigation of risk are (see Fig. 2, from [3]): Identification (classification), contention and mitigation, evidence preservation

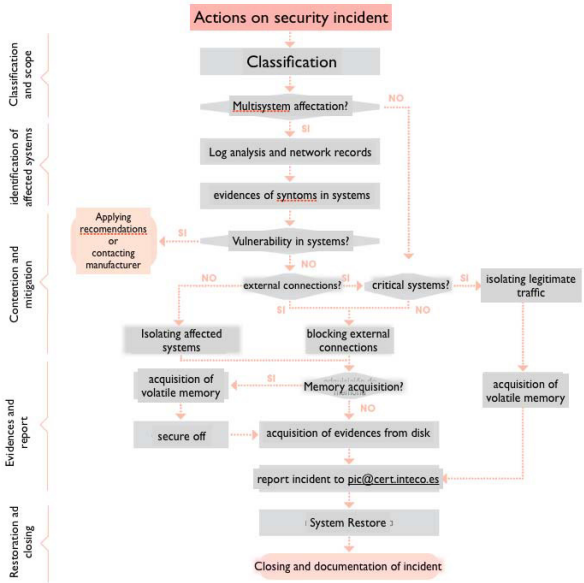

Fig. 2. Flowchart of action in a security incident according [3]

and legal considerations, documentation and recovery. The elements in these phases have different nature. On the one hand, classification and identification have static nature while actions correspond to protocols (non complex plans).

Static dimension versus Dynamic dimension: Preliminary analysis of documents show that two ontological dimensions are combined. The first one refers to (statical) identification of main elements. The importance of this dimension in SIS documents is due to solving/repairing/mitigation methods that strongly depend on the secure identification. Despite that, it is hard to state the complex relationship among different categories. SIS documents often enumerate elements appearing in a particular organization. The methods often depend on such classification. However refinements of categorization aid to specify the methods.

The second dimension is about the description of dynamic elements of SIS scenarios, as for example protocols and methods. The description of the protocol is more precise than risk identification. This observation suggests to define precise flowchart-based subontologies to describe them.

Features of Descriptional ontological level: The semantic description SIS has the great advantage of allowing to compare the INTECO-CERT approach to risks with other related classifications and/or ontologies, in order to evaluate its soundness. Particularly interesting is to consider its relationship with the following six general categories of information technology risk [14]. Note that concept mapping between these general categories and INTECO-CERT categories provides useful insights to enrich the description of action classes related with them. The relationship among both categories is depicted in Fig.3 . The relationship

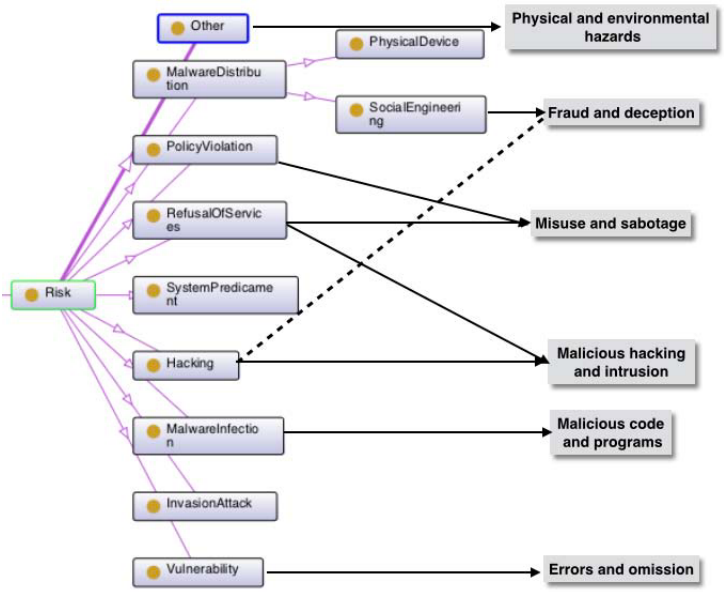

Fig. 3. Descriptional class `Risk` and its relationship with categories from [14]

is rough and it has to be understood as a set of incipient refinements of the ontology. It is interesting to highlight some of them:

- *Malicious code and programs*: The concept contains `MalwareInfection`. Thus, ontology could be expanded by adding classes to prevent risks. It requires protection at the individual and system level.
- *Malicious hacking and intrusion*: contains `Hacking` and `InvasionAttack`. However, INTECO classification also considers malicious hacking whitout intrusion (`RefusalOfService`).
- *Fraud and deception*: Description in [14]:
  Various forms of attacks in the form of spoofing, masquerading, or salami attacks have been used to do damage to privacy. Social engineering is often an effective means to obtain illegal access.
  First paragraph of the description corresponds to `SocialMalware` and part of `Hacking` while the second one correspond to `SocialEngineering`. In this case ontology is more specific than category from [14].
- *Misuse and sabotage*: Closely related with `Vulnerability`. It also contains `PolicyViolation`. The first class is one of the underspecified concepts in INTECO-CERT. The original category from [14] represents the resources that can be misused, or vandalized through unauthorized access.
- *Errors and omissions*: Closely related with `Vulnerability`. According to [14], this category assumes accidental (software) errors, to include unintended destruction of files or data, as well as routing or transmission errors. This also includes programming errors. Thus it seems that `Vulnerability` class has not a good level of granularity in INTECO document.

– *Physical and environmental hazard*: It is out of the scope of `Risk` class of [3].

Ontological analysis of this kind of relationships among categorizations can be used in other parts of the ontology, by using another related security ontologies. Even it can induce to distinguish between safety and security, in order to refine ontology in some SIS scenarios [13]. A more detailed risk classification and description needs the formal inclusion of *damage* concept. This inclusion would force to refine risk categories, as in [9]. Also, it is interesting to refine concepts about cyber attacks from [6]. In this way the inclusion of *target* concept allows the introduction of new mitigation strategies at dynamic level.

Dynamic ontological level. One of the INTECO–CERT/CNPIC tasks is the response to security incidents reported as occurring in Critical Infrastructures by users of this service, ensuring that the relevant information is stored. The description of the process follows the scheme shown in Fig. 2 from [3], which can be fairly represented using the flowchart representation. Although there exists other ontological representations of flowcharts[2], as it was already mentioned, a specific sub ontology is designed to manage these critical elements in SIS.

Dynamic dimension of semantic analysis of SIS guides consists of flowchart based representation of protocols. The version of this basic concept on the ontology is depicted in Fig. 4. A singular feature of the ontology is the identification between `AtomicAction` and `FlowchartAction` classes. This non orthodox equivalence is the result of a group discussion among authors. Ontological distinction between action and representation of the action within flowcharts is discarded. In this way action class is used in both levels.

Although there are multiple variants of flowcharts (Petri nets, ASM charts and so on), we can consider the simplest one, with only two types of nodes (boxes): *Action boxes* and *Decision boxes*. The first ones contain a set of actions that the user should execute in that state, therefore an action box must have one and only one output path. They are represented as class `ActionBox` in our ontology. The second ones are *Decision boxes* where the inner text is a condition to be verified. The next current state depends on the value at which the condition may be evaluated. This kind of nodes can have multiple output paths. Decision boxes are modeled by class `DecisionBox` in our ontology. Fig. 5 it is shown the hierarchy of classes of our sub-ontology. It can be seen that `ActionBox` and `DecisionBox` are subclass of a more generic concept that we have called `InnerClass` (representing the internal nodes of a flowchart). In this way some restrictions on the classes can be added:

$$\texttt{ActionBox} \sqsubseteq (= 1\ \texttt{hasOutputPath.Path}),$$
$$\texttt{DecisionBox} \sqsubseteq (\geq 1\ \texttt{hasOutputPath.Path})$$

As it is shown in Fig. 2 some kinds of flowcharts have two special nodes. Those that don't have an input path (i.e. input degree in the graph is equal zero) and those that don't have an output path (i.e. ouput degree is zero). This nodes are represented in our flowchart ontology thanks to `StartBox` and `EndBox` classes,

[2] e.g. `www.daml.org/ontologies/183`, `www.daml.org/ontologies/306`, `bioportal.bioontology.org`

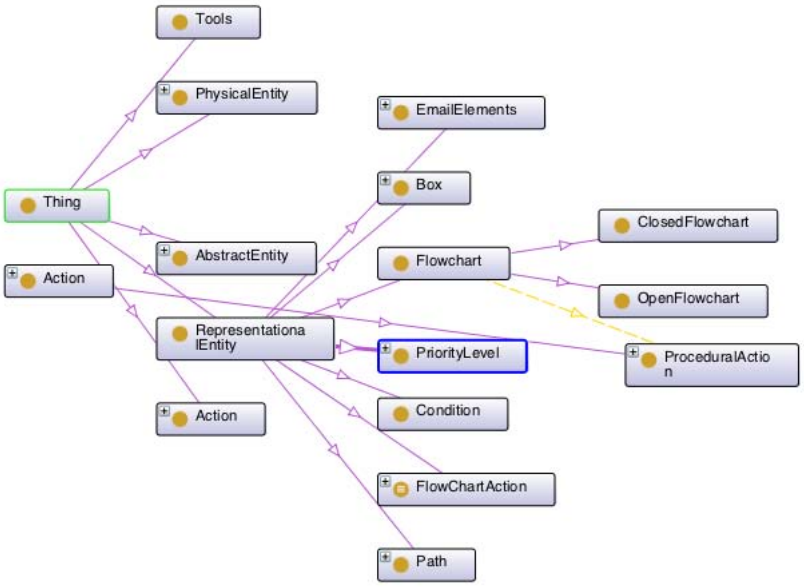

Fig. 4. Flowchart as basic element in representational dimension of the ontology

respectively. We can enforce these constraints making these classes subtypes of `OutputPathBox` and `InputPathBox`:

$$\texttt{InputPathBox} \sqsubseteq \exists\ \texttt{hasInputPath.Path},$$
$$\texttt{OutputPathBox} \sqsubseteq \exists\ \texttt{hasOutputPath.Path}$$

Thus, an instance of `InnerBox` must inherit both restrictions:

$\texttt{InnerBox} \sqsubseteq \exists\ \texttt{hasInputPath.Path}$, $\texttt{InnerBox} \sqsubseteq \exists\ \texttt{hasOutputPath.Path}$

Some other key concepts and classes of this ontology (but not shown in Fig.5) are `Condition` and `Path` with the usual associated semantics.

The stage of (internal) ontology articulation allows to build semantic bridges among the above sub-ontologies. In fact, descriptional and dynamic ontologies share concepts of common ontological nature. This step produces the refinement of the high level of the ontology.

With respect to ontology population, two main kinds of individuals for ontology population can be extracted from documents (protocols and incidents). Revising population methods for security ontologies also suggests the need of extending the information of the document.

4 Logical Specification of Meta-Security in IRD

Specification of the ontology opens the possibility of including constraints that would be included in the SIS documentation (in natural language). Some of them

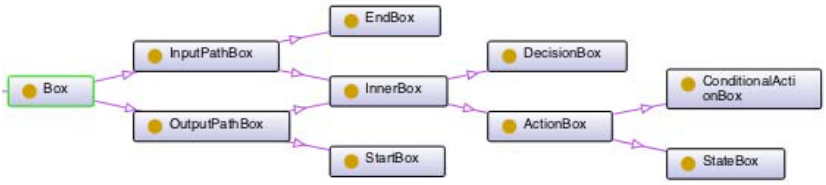

Fig. 5. Flowchart box element class

would allow to monitorize integrity/safety constraints. For example, the system only considers as *detected incident* one for which it has an evidence:

$$\texttt{Detection} \equiv \exists \texttt{hasEvidence}.\texttt{Evidence}$$

Likewise, flowchart semantic specification allows to instantiate protocols, making each one a complete and consistent representation of a security method. In particular, only flowcharts representing approved methods can be included:

$$\texttt{FlowChart} \sqsubseteq (\geq 1\ \texttt{represents}.\texttt{Action})$$

where $\texttt{Action} \equiv \texttt{AtomicAction} \sqcup \texttt{ProceduralAction}$. The absence of classification of an incident is prevented by a restriction axiom on the property `originIn`:

$$\texttt{Incident} \sqsubseteq (= 1\ \texttt{originIn}.\texttt{Risk})$$

5 Related and Future Work

The paper shows how the construction of ontologies from security reports - instead of selecting a standard security ontology- habilitates the use of formal methods that insure their safety, by clarifying process and descriptions. As it has been already commented in the introduction, it is not the goal to build (another) ontology on security, neither it is a goal to reproduce a standard method to extract one ontology from a document. The aim is to exploit the ontology construction itself to clarify and revise security reports. Therefore, the key is the application of SWT steps from the document information.

The evaluation of the overall process depends on two key stages that, because of lack of space, have not been discussed in this paper. On the one hand, since the process aims to debug and clarify security reports by means their specifications and ontologies, the evaluation of the method has to be based on the feedback from the report author. On the other hand, the secondary product (the ontology itself) is evaluated by comparing it with standard ontologies on both the same scope and the intended use. The soundness of the new ontology is useful to revise the report itself. However, the ontologies built from standard security descriptions are very useful to enhance the behavior of multi-agent-systems for security issues (see e.g. [8]). Likewise, the tight relationship between the knowledge contained in the report and the performed one allows to use reasoning services. This feature needs of a refined classification of different reasoning services that will be described in a next paper.

There exists a number of security ontologies with different features and scope (see [2] for a general vision of the field). In general, security ontologies are built in the traditional fashion in OE, whilst our approach is the re-use of OE methods to validate reports in IRD framework. Of course, an information security ontology should define the most important security issues and concepts and the relationships between them. Thus reports, as these analyzed here, have to describe such elements. Therefore, OE extraction methods will produce ontologies which can be comparable with the former ones. By means of the comparison, it can be estimated the soundness of the report, to induce refinements or reparations.

For example, it is interesting to compare the ontology with the Security Ontology (SO)[3] from [7]. Both ontologies complement each other with features as risk identification (from ours to the SO) and countermeasures analysis (from the SO to ours). Although it is evident that the countermeasure ontology from SO is richer than [3], it can consider that it is useful as addenda of the document.

As future work, it is very interesting to recover knowledge from security reports by applying Formal Concept Analysis (FCA)[12]. By using FCA it would be possible to extract hidden concepts from the protocols which are susceptible to be considered by the authors of security documents in the organization.

References

1. Aranda-Corral, G.A., Borrego-Díaz, J.: Mereotopological Analysis of Formal Concepts in Security Ontologies. In: Herrero, Á., Corchado, E., Redondo, C., Alonso, Á. (eds.) Computational Intelligence in Security for Information Systems 2010. AISC, vol. 85, pp. 33–40. Springer, Heidelberg (2010)
2. Blanco, C., Lasheras, J., Valencia, R., Fernández, E., Toval, A., Piattini, M.: A Systematic Review and Comparison of Security Ontologies. In: Proc. 3rd. Int. Conf. on Availability, Reliability and Security, pp. 813–820. IEEE Computer Society (2008)
3. Díaz-Vico, J., Fírvida-Pereira, D., Lozano-Merino, M.A.: Identification and reporting of security incidents for strategic operators. A basic guide for the protection of critical infrastructures. National Institute of Communication Tecnologies
4. Díaz-Vico, J., Fírvida-Pereira, D., Lozano-Merino, M.A.: The Operator Console. A Basic Guide to Critical Infrastructure Protection. National Institute of Communication Tecnologies
5. Fenz, S., Ekelhart, A.: Formalizing information security knowledge. In: Proc. 4th Int. Symp. on Inf. Comp. & Comm. Security, ASIACCS 2009, pp. 183–194. ACM (2009)
6. Geers, K.: Strategic Cyber Security. NATO Cooperative Cyber Defence Centre of Excellence (2011)
7. Herzog, A., Shahmehri, N., Duma, C.: An Ontology of Information Security. Int. J. Information Security and Privacy 1(4), 1–23 (2007)
8. Herrero, A., Navarro, M., Corchado, E., Julián, V.: RT-MOVICAB-IDS: Addressing real-time intrusion detection. Future Generation Comp. Syst. 29(1), 250–261 (2013)

[3] `http://www.ida.liu.se/~iislab/projects/secont/`

9. Kim, W., Jeong, O.-R., Kim, C., So, J.: The dark side of the Internet: Attacks, costs and responses. Inf. Syst. 36(3), 675–705 (2011)
10. Mace, J.C., Parkin, S., van Moorsel, A.: A collaborative ontology development tool for information security managers. In: Proc. 4th Symp. Comp. Human Inter. for the Management of Information Technology, 10 pages. ACM (2010)
11. Pereira, T., Santos, H.: An Ontology Based Approach to Information Security. In: Sartori, F., Sicilia, M.Á., Manouselis, N. (eds.) MTSR 2009. CCIS, vol. 46, pp. 183–192. Springer, Heidelberg (2009)
12. Sarmah, A., Hazarika, S.M., Sinha, S.K.: Security Pattern Lattice: A Formal Model to Organize Security Patterns. In: Proc. 19th Int. Conf. on Database and Expert Systems Application (DEXA 2008), pp. 292–296. IEEE Computer Society (2008)
13. Sadvandi, S., Chapon, N., Piètre-Cambacédès, L.: Safety and security interdependencies in complex systems and SoS: challenges and perspectives. In: Complex Systems Design and Management, pp. 229–241. Springer, Heidelberg (2012)
14. Smith, G.E., Watson, K.J., Baker, W.H., Pokorski, J.A.: A critical balance: Collaboration and security in the IT-enabled supply chain. Int. J. Production Research 45(11), 2595–2613 (2007)