A Security Pattern-Driven Approach toward the Automation of Risk Treatment in Business Processes

Angel Jesus Varela-Vaca[1], Robert Warschofsky[2], Rafael M. Gasca[1], Sergio Pozo[1], and Christoph Meinel[2]

[1] Computer Languages and Systems Department,
Quivir Research Group
ETS. Ingeniería Informática, Avd. Reina Mercedes S/N,
University of Seville, Seville, Spain
{ajvarela,gasca,sergiopozo}@us.es
[2] Hasso-Plattner-Institute
Prof.-Dr.-Helmert Str. 2-3
14482 Potsdam, Germany
{robert.warschofsky,meinel}@hpi.uni-potsdam.de

Abstract. Risk management has become an essential mechanism for business and security analysts, since it enable the identification, evaluation and treatment of any threats, vulnerabilities, and risks to which organizations maybe be exposed. In this paper, we discuss the need to provide a standard representation of security countermeasures in order to automate the selection of countermeasures for business processes. The main contribution lies in the specification of security pattern as standard representation for countermeasures. Classical security pattern structure is extended to incorporate new features that enable the automatic selection of security patterns. Furthermore, a prototype has been developed which support the specification of security patterns in a graphical way.

Keywords: Business Process Management, Security, Pattern, Risk Treatment, Automation.

1 Introduction

New technologies have emerged into the business process scene providing applications to automate the generation of IT products based on business processes management. Most of the recognized business process products, such as Intalio BPM suite, Bonita Soft, BizAgi, and AuraPortal BPMS, are capable to generate automatically entire applications from definitions of business process diagrams and interactions with web services. These applications do not pay attention on security risks and less in the treatment of these. In general, security treatments are applied in a second thought. It is desirable to provide the business analyst with tools that enable the risk assessment of business process designs and also

the specification of necessary security countermeasures to apply. Nevertheless, in general the selection and configuration of security countermeasures is a human, manual, complex, time consuming and error-prone task that involves many security stakeholders p(managers and administrator). Security countermeasures can vary from technical controls to management controls; such as list of procedures, backup policies, and the specification of access-control policies). In order to understand the complexity of security countermeasures an example could be network Intrusion Detection Systems (IDS) where two different approaches [5] [9] apply machine learning methods for intrusion detection. The question is which specific approach is adequate for the requirements of the organization.

The main problems regarding to countermeasures are: (1) how to describe the countermeasures in business processes; (2) countermeasures are very heterogeneous; (3) countermeasures are described in natural language or informal way (4) in general the selection of countermeasures is carried out in manual way without criteria. A derived problem emerges from the complexity in the selection of a set of countermeasures that comply with several organizational constraints. This complexity increase when there exist multiple objectives to achieve such as reduce the return of investment (ROI) until certain value, reduce risks until ten percent, among other.

In this work, we propose a formalization based on security patterns templates in order to standardize the representation of security countermeasures in business processes. Furthermore, we propose an extension of these patterns with new features that enables to include organizational metrics and constraints for the automatic selection of countermeasures. In order to overcome the complexity of selection we propose to apply automatic algorithms based on artificial intelligence techniques. To support the proposals, OPBUS framework [14] has been improved by including a risk treatment stage which support the agile specification and automatic selection of security patterns as countermeasures.

The rest of the paper is organized as follows: in Section 2 security patterns, and the need to model and extend security patterns is detailed; in Section 3, an application example scenario is presented; in Section 4 the prototype developed is described; Section 5 gives an overview about related work in the domain of security patterns and model-driven security; the last section concludes this paper and outlines the future work.

2 Specifying Security Patterns

Security patterns [12] are widely recognized way for the description of security solutions. Security patterns are based on the idea of design patterns that has been introduced by Christopher Alexander in 1977: *"A pattern describes a problem which occurs over and over again our environment, and then describes the core of the solution to that pattern"*.

In general, security patterns [11] are defined in an informal way, usually using the natural language. Patterns are described in documents that have a specific structure: Name, Context, Forces, Problem, Solution. Other sections; such as Implementation details, Structure, Examples, can be incorporated to the security patterns in order to improve its information.

There exist relevant approaches where security patterns has been formalized such as [11] [12] [7]. The formalization in [12] is focused on the definition of ontologies to map security concepts within security patterns in order to enable search engines and query capabilities. In [7], security patterns have been formalized as profiles to automate the generation of security policies for SOA environments. In [11], a catalogue of security patterns is defined using natural language, and UML diagrams. Nevertheless, these formalizations are unsuitable to automate the application of security patterns.

In our approach, the security patterns are utilized for the selection of treatments for certain risks. Hence, security patterns have to be equipped with information that enable the evaluation how much adequate is a pattern among others. Furthermore, in the selection security patterns should comply with certain business objectives (cost, acceptable risk levels) pre-stated. We propose the modelling of security pattern as shown in Table 1. The table presents, regarding to classic security patterns, four new sections: Security goal, Security intention, Risk type and Attributes. This extension is based on the UML Profile for Modeling QoS and Fault Tolerance (hereinafter UML profile) [1]. In the next sections, security pattern structured is detailed.

Table 1. Template of extended security patterns

Name	Description	is a label that identifies and summarize of the pattern	
	UML QoS & FT Profile	::QoSValue	
Security Intentions	Description	is a label to describe security intentions to cover	
	UML QoS & FT Profile	::QoSCharateristic	
Security Goals	Description	list of indicates the security goals to fullfil	
	UML QoS & FT Profile	::QoSCharateristic	
Context	**Risk Type**	Description	indicates el type of risk of the pattern
		UML QoS & FT Profile	::RiskAssessment ::Treatment ::TreatmentOption
	Attributes	Description	describes the attributes concerning to the context related to risk
Forces	Description	describe the constraint that exist in the business process, they affect the problem	
	UML QoS & FT Profile	::QoSConstraint	
Problem	Description	indicates the problem that occurs within the context	
	UML QoS & FT Profile	::QoSCharateristic	
Solution	Description	indicates the solution for the problem within the context	
	UML QoS & FT Profile	::QoSCharateristic	

2.1 Security Goals and Intentions in Security Patterns

Security patterns as countermeasure specify well-known solutions to common security problems, hence there should exist a direct relation between countermeasures and the security goals the pattern is enforcing. In fact, security pattern templates as shown in [12] this relation is fuzzy. In other approaches, such as the approaches in [7] [6], the authors define a relation with security intentions attaching security intentions in the problem section. Although, a security intention could enforce various security goals this relation is also fuzzy. For instance, in [7], the authors define the security patterns: 'Secure Pipe' and 'Information Protection' as intention. However it is not clear which security goals the pattern is enforcing. We propose the enhancement of security pattern templates with security goal and intention concepts in order to provide with new criteria for the selection of security patterns.

For a better understanding of the problem and due to the heterogeneity of all concepts, we have formalized all concepts by means of an ontology as shown in Figure 1. The ontology includes various examples to illustrate each concept. For instance, we can observe how risk concepts in the context are related to threats and vulnerabilities. Threats and vulnerabilities represent problems from the point of view of a security pattern. There exist various databases, such as NIST Vulnerability Database [3], and Common Weakness Enumeration (CWE) [2], that provide information referring to technological vulnerabilities. Identifiers utilized by CWE or NIST can be adopted for the specification of security patterns in our approach. For instance, Figure 1 shows a SQL injection attached the identifier CWE-89. This is used by CWE dictionary to identify a specific SQL-Injection vulnerability. In addition, CWE provides a particular section (Common Consequences) to indicate which security goal is affected. In the same way, other databases such as CRAMM database can be adopted by means of defining the alignment of its concepts.

2.2 Risk Type and Attributes for Selection

Following the UML Profile, there exist four categories of risk treatments: avoid, reduce, transfer and retain. Security patterns as countermeasures should be categorized inside of one of the four categories. This classification could be useful to rule out countermeasures for in-suitable risks and to do more efficient searches. We therefore propose to include a risk type property in order to support the classification of the pattern. For instance, an organization has decided to transfer one risk making outsourcing, if there exist a database with one hundred security patterns splitted in four different categories, it would be desirable that rule out several patterns for its category and focus on the patterns which category is transfer.

On the other hand, metrics are necessary in order to evaluate which countermeasure is better than others. We propose to incorporate a new section within the context to gather metrics that enable the measurement of the security pattern. In general, these attributes are related to metrics that the organization

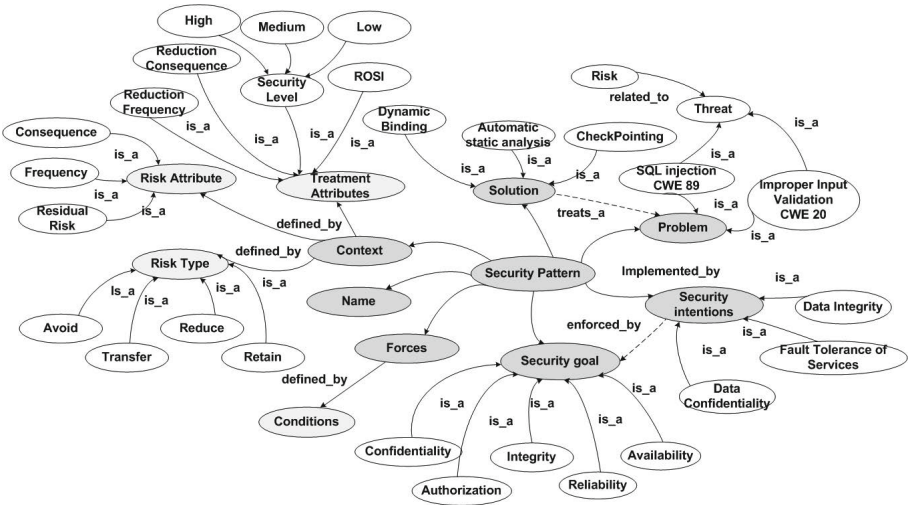

Fig. 1. Ontology for problem, solution, security goals, and security intentions

need to carry out an evaluation of countermeasures in monetary terms; such as annual cost, cost of implementation, annual loss expectancy (ALE). Otherwise, from the point of view of the risk management it would be useful to introduce metrics related to the level of risk allowed, and security level (High, Medium and Low), risk reduction, impact, priority of application of the pattern, etc. In Figure 1 various examples of risk attributes are given. For instance, an organization has detected a vulnerability related to SQL injections since the information introduced in a web form is not validated. In most of the cases, SQL injections are prevented by using information input validation. Other mechanisms such as data analysis methods can exist as well. Different mechanisms are adequate however which one is the best for the organization.

2.3 Forces and Selection of Security Patterns

As mentioned earlier, forces section indicates conditions in the context for the application of the solution. These conditions reflects restrictions of application relative to the context, thus to the metrics previously defined in the context. An example of constraint could be for instance $NeutralizationSQLInjection.Cost \leq TotalCost$ that indicates the cost of the implementation of 'Neutralization of SQL Injection' must be less than the total cost. These concepts (Cost and TotalCost) has previously been defined in the context. Another example could be related the level of security achieved in the application of this pattern, *SecurityLevel=Medium*.

The selection aim consist of finding a list of security patterns (Figure 2) that comply with the objectives specified in the business process, the constraint of application indicated in the forces of security patterns, and treat risks in the activities that need to be treated. This selection could be carried out in manual

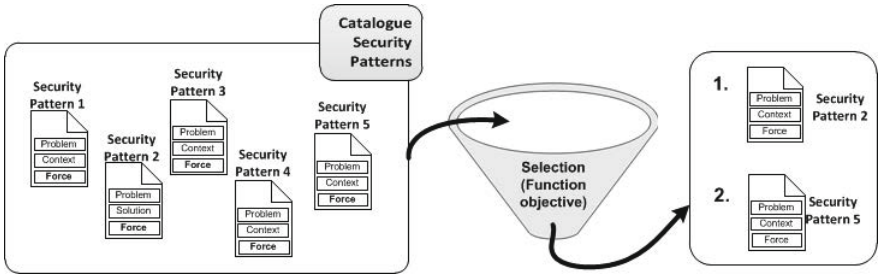

Fig. 2. Selection process of security patterns

way or in automatic way by means of certain reasoning technique. As proof of concept, an approach for the automatic selection of countermeasures could be an algorithm that receives as input: a business process, and a catalogue of countermeasures (set of security patterns). The business process composed by a set of activities (that need to be treated), risks associated to activities and the business process, a set of threats, and objectives of the business process. Firstly, the catalogue is reduced by means of security goal and risk type stated in the business process. After that, the algorithm strives for the selection of security patterns that treat risks of activities, comply with cost of the business process, and satisfy the constraints in the pattern. Finally, the algorithm returns a set of security patterns associated to activities of the business process.

2.4 Problem and Solution Items in Security Patterns

Security patterns already include a section to describe problems within the context. In [12], the authors use the problem section to describe the problem by natural language. However, in [7], the problem section is utilized to specify the security intention as a objective to achieve and not to describe problem. For our propose, the problem section is related to a specific threat or vulnerability.

In our particular use of security pattern as countermeasures, solutions are related to solve security risks concerning threats or vulnerabilities. That is, the solution field of security pattern template has been used to gather information about particular solutions to certain kind of vulnerabilities. For example, a solution such as 'Automatic static analysis' treat the vulnerability of 'CWE-20 - Improper Input Validation'. This solution has been obtained from the CWE database. CWE state three different types of detection methods: automated static analysis, manual static analysis and fuzzy. These methods could be used as identifiers to specify solutions in the security patterns at this level. The identifiers used in this field are general and do not describe explicit mechanism. In the same way, we can define our own identifiers according to the solutions that are not picked up in the CWE database. For instance, Figure 1 shows a solution named 'Checkpointing' concern to the 'Fault Tolerance Services' intention.

Although, this solution is not in CWE database. However, it can be applied in business processes to enforce the reliability even the integrity as demonstrated in [13]. For a better understanding, the description of security patterns and the automatic selection of these are illustrated in an example in section below.

3 Illustrative Example

An example scenario where a hosting organization would like to provide a service to publish entries in a blog system. The organization has decided to automate this using business process management systems and developed a business process model which contains three basic activities: (1) Login, to enter username and password information to authenticate customers; (2) Request, to request information (title and content) about a new entry for blogs; (3) Publish: the entry information is registered in the database and published in a blog system.

The organization wishes to increase the security where the priority is customers integrity and confidentiality. In manual way, a security analysis of the system can detect possible vulnerabilities referring to: (1) Passwords integrity and confidentiality due to use an insecure channel in public networks; and (2) Data integrity and confidentiality due to an insecure channel in public networks.

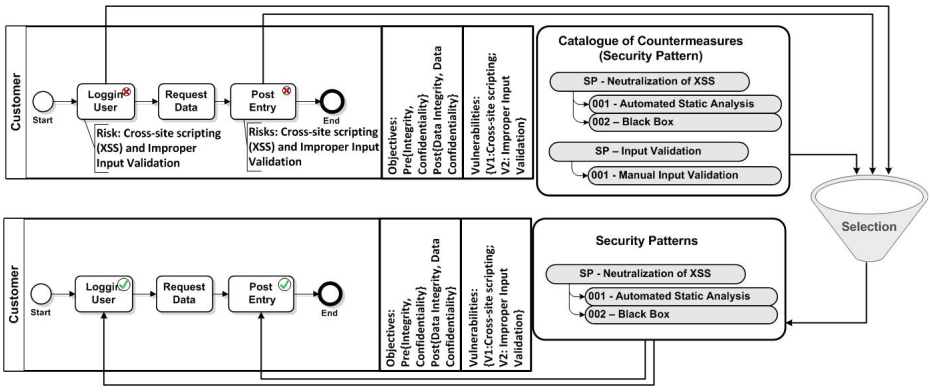

Fig. 3. Example of automatic risk treatment of a business process

The organization wish to automate the customer security by including countermeasures in the business processes. In order to ensure a security-aware development of business process a risk assessment is done through OPBUS [14]. After diagnosis, OPBUS indicates which vulnerabilities and activities have to be treated as shown by red marks in Figure 3. In that case vulnerabilities 'V1' concerns to 'Cross-site scripting (XSS)' vulnerability and 'V2' concerns to 'Improper input validation' produces a risk in activities 'Loggin User' and 'Post Entry' that exceed the risk levels allowed. Hence, the exploiting of vulnerabilities or the materialization of threats can produce effects that cannot be assumed and they has

to be treat. After a first filter, OPBUS provide two types of security patterns to treat these vulnerabilities as shown in Figure 3. These countermeasures are defined based on CWE database which indicates some detection methods for this kind of vulnerabilities. Detection methods are mechanisms to prevent the threat of vulnerabilities. In this particular case countermeasures are stated as follows: (1) CWE-79 solutions: Automated static analysis based on data flow analysis, and Black box based on automated test; (2) CWE-20 solution: Manual input validation.

Table 2. Neutralize XSS based on data analysis

Name		
Neutralization of XSS		
Problem		
CWE-79: Improper Neutralization of Input During Web Page Generation		
Security Intention		
Data integrity		
Security goal		
Integrity,Confidentiality		
Context		
Risk Type		
Reduce		
Attributes		
Risk Reduction Freqpuency (RRF)	[30,60]%	[50,70]%
Risk Reduction Consequence (RRC)	[10,30]%	[10,30]%
Annual Number Attacks (ANA)	30	30
Cost per Attack (CA)	[60-100]	[60-100]
Cost Solution (CS)	[500-600]	[900-1000]
Forces		
$ANA * CA + CS < TotalCost$ $CS < 600$		
Solution		
Automated Static Analysis - Data Flow Analysis		Black Box - Testing

CWE-79 also provides a long list of potential mitigation even specifying implementation details to mitigate the effects of this vulnerability. Following descriptions of CWE database, two examples of security patterns related to the mitigation of cross-site scripting have been defined as shown in Table 2. A set of attributes (ANA, CA, CS, RRF, RRC) and constraints have been included in attribute and forces sections respectively.

The main difference between both countermeasures are the values in the attributes. Attributes included within context can be stated by a single value and intervals when there is no certainty of the value. The main question is which countermeasure is better for the organization. After selection, OPBUS has proposed 'Neutralization of XSS' as solution to treat the problems identified in the activities. If we observe carefully the main differences between both countermeasures are the cost of implementation (CS) and the risk reduction in terms of frequency (RRF). For instance, 'RRF' in the first countermeasure has a interval of [30,60]%, and the second one has a interval of [50,70]%. Taking into consideration the constraints included in forces section: $CS < 600$ and

$ANA * CA + CS < TotalCost$. It is not clear which countermeasure is better in plain sight a comparative process is necessary. However, in other cases with a large number of vulnerabilities the list of countermeasure could multiply and to select the adequate countermeasures become a very complex task. The complexity in the selection of adequate countermeasure makes it mandatory to include mechanisms that enable the automatic selection treatment.

After applying the security patterns the model can be re-diagnosed obtaining residual risks. The results of these residual risks are lower than the previous. Figure 3 shows the results of the diagnosis after including security patterns as countermeasures.

4 OPBUS Prototype

Security patterns has been integrated as part of the OPBUS model. The model is composed of three main sub-models: risk model, business process model, and security policy model. Security patterns have been modelled as part of the security policy model, and according to the structure listed on Table 1. In general, security countermeasures are gathered by means of security policies. The risk model provide an extension of business process models, and security policy model is used to define policies by means of security patterns as specific countermeasures. Security patterns can be defined as security constraints that enforce certain security goals within a security policies. As we can observe in Figure 4 risk model includes 'Treatment' and 'Countermeasure' by means of security patterns to enable the specification of risk treatments into threat scenarios.

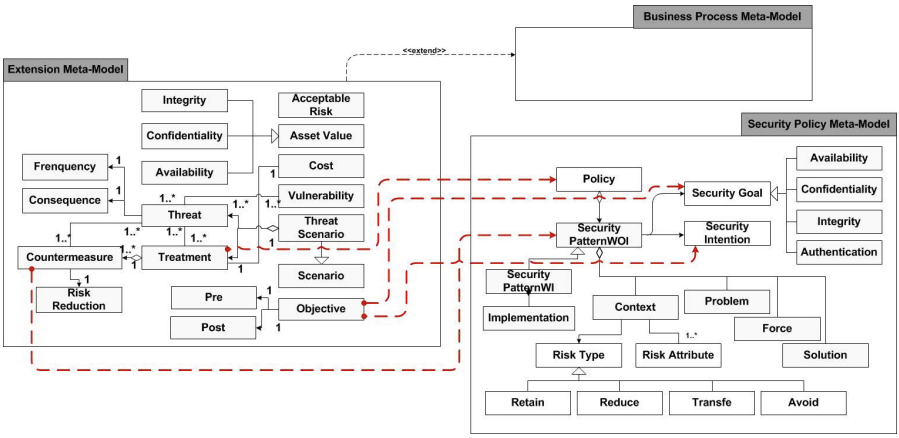

Fig. 4. Business process extensions

A prototype has been developed as part of the OPBUS tools [4]. The prototype have been developed as Eclipse plug-in which integrates a business process modeller that enable the automatic risk assessment of the model. The main novelty is to support the specification of security patterns. A specific properties

tab has been enable in order to set up the security pattern attributes. Security patterns can also be linked to threats as countermeasures related to certain vulnerabilities.

5 Related Work

Various approaches have emerged in the context of business process management in order to bridge the gap between security and business domains [15] [8] [7]p [10]. Most of these initiatives are focused on providing new domain-specific languages (DSLs) for the annotation of assets, security requirements, and threats into the business process models. Nevertheless, most of the studies avoid the introduction of risk treatments.

In [8], Menzel et al. propose an approach to adequate the generation of security requirement of authorization and trust in accordance with specifics risk levels. In the same way, in [15], Wolter et al. provides security annotations for graphical business processes that enable the set up of authorization directly over the model. However, these approaches are only focused on the definition of authorization and trust in service-based environments.

Concerning to security pattern, in [7], Menzel provide a new application for security patterns to automate the generation of security policies in SOA environments. Menzel provides different security patterns in order to fulfill certain security intentions such as User Authentication, Identity Provisioning, Data Confidentiality and Data Authenticity. Furthermore, security patterns has been formalized by means of ontologies. The main problem in the approach is the coupling to the solution. That is, security model, formalization and security patterns have been defined to support the specification of solutions only valid to SOA environments.

6 Conclusion

We have proposed to incorporate a formalization based on security pattern templates in order to represent security countermeasures in business process. For this reason, we have been modelled, extended and adapted security patterns templates to support it in the risk assessment of the business processes. Security pattern has been equipped with new sections concern to security goals, security intentions, risk type, and risk attributes. Furthermore, security patterns have been interconnected with real concepts belong to database of vulnerabilities such as NIST or CWE in order to do more compatible the approach. In other respects, transformations are available in the framework to translate security pattern into certain security configurations.

To the best of our knowledge, this is the first proposal that define a specific format to define countermeasures to automate the risk treatment in business processes. To future work, we propose to define algorithms to automate the selection of optimal countermeasures following security pattern templates proposed in this paper. Furthermore, we propose to extend the prototype including these algorithms for the automatic selection of countermeasure.

Acknowledgements. This work has been partially funded by the Department of Innovation, Science and Enterprise of the Regional Government of Andalusia project under grant P08-TIC-04095, by the Spanish Ministry of Science and Education project under grant TIN2009-13714, and by FEDER (under the ERDF Program). The authors would like to thank Dr. Michael Menzel and Hasso-Plattner-Institute for the valuable information that has contributed to develop the ideas in this paper.

References

1. UML Profile for Modeling QoS and Fault Tolerance Characteristics and Mechanisms (2009), http://www.omg.org/spec/QFTP/1.1
2. Common Weakness Enumeration (2011), http://cwe.mitre.org/index.html
3. NIST National Vulnerability Database (2011), http://nvd.nist.gov/
4. OPBUS tools (2012), http://estigia.lsi.us.es/angel/OPBUS/
5. Corchado, E., Herrero, L.: Neural visualization of network traffic data for intrusion detection. Applied Soft Computing 11(2), 2042–2056 (2011)
6. Menzel, M., Warschofsky, R., Meinel, C.: A pattern-driven generation of security policies for service-oriented architectures. In: 2010 IEEE International Conference on Web Services (ICWS), pp. 243–250 (July 2010)
7. Menzel, M.: Model-driven Security in Service-oriented Architectures. Ph.D. thesis. Hasso-Plattner - University of Potsdam (2010)
8. Menzel, M., Thomas, I., Meinel, C.: Security requirements specification in service-oriented business process management. In: International Conference on Availability, Reliability and Security (ARES), pp. 41–48. IEEE Computer Society (2009)
9. Mrutyunjaya, Abraham, A., Das, S., Patra, M.R.: Intelligent Decision Technologies 5(4), 347–356 (2011)
10. Rosemann, M., zur Muehlen, M.: Integrating risks in business process models. In: 16th Australasian Conference on Information Systems (ACIS 2005), Paper 50, pp. 1–10 (2005)
11. Schumacher, M. (ed.): Security Engineering with Patterns - Origins, Theoretical Models, and New Applications. LNCS, vol. 2754. Springer, Heidelberg (2003)
12. Schumacher, M., Fernandez-Buglioni, E., Hybertson, D., Buschmann, F., Sommerlad, P.: Security Patterns: Integrating Security and Systems Engineering. John Wiley and Sons, Ltd (2006)
13. Varela-Vaca, A.J., Gasca, R.M.: OPBUS: Fault Tolerance against integrity attacks in business processes. In: 3rd International Conference on Computational Intelligence in Security for Information Systems, CISIS 2010 (2010)
14. Varela-Vaca, A., Gasca, R., Jimenez-Ramirez, A.: A model-driven engineering approach with diagnosis of non-conformance of security objectives in business process models. In: 2011 Fifth International Conference on Research Challenges in Information Science (RCIS), pp. 1–6 (May 2011)
15. Wolter, C., Menzel, M., Schaad, A., Miseldine, P., Meinel, C.: Model-driven business process security requirement specification. Journal of Systems Architecture - Embedded Systems Design 55(4), 211–223 (2009)