A Systematic Security Analysis of Information Systems

Roberto Ortiz, Santiago Moral-Rubio
Dep. Information Security
BBVA Group
Madrid, Spain
r.ortizpl@gmail.com;
santiago.moral@bbva.com

Javier Garzás
Kybele Group
Rey Juan Carlos University
Madrid, Spain
javier.garzas@urjc.es

Eduardo Fernandez-Medina
GSyA Research Group
University of Castilla- La Mancha
Ciudad Real, Spain
Eduardo.FdezMedina@uclm.es

***Abstract*—The integration of security into software development processes through methodologies guarantees that these developments are controlled, planned and verified at all stages. It is thus possible to avoid unexpected errors whilst improving the quality and security of the system produced. These methodologies can be enriched with the use of security patterns that compile the knowledge of security experts in a documented and structured manner, providing us with a systematic means to solve recurring problems. In this paper we shall summarize pattern-based security methodology in order to support both the construction of secure information systems and the maintenance of the level of the security attained, upon which we are currently working. We shall also provide an in–depth study of the analysis stage, showing the elements of which it is composed, such as the input and output artifacts, together with the main roles and activities that participate in it.**

Keywords-Security; Security Methodology; Secure Systems Analysis; Security Patterns; Secure Information Systems

I. INTRODUCTION

In recent years, the majority of attacks against organizations aim to exploit the vulnerabilities caused by a poor design and development of the functionalities given to information systems (IS) [1]. The need to build secure IS has therefore arisen, and this situation has encouraged the scientific community to research the integration of security into IS development processes [2-4]. This integration is established through development methodologies since they offer, from the first stages, a systematic, planned, controlled, verifiable and detailed process that will avoid the appearance of uncontrolled security errors, thus mitigating the possible risks associated with the implementation of new functionalities in an IS [5]. The main advantage of these processes is also based on the fact that they are decomposed into elementary tasks in which each task is identified by a procedure that defines how to carry it out, the most appropriate actors for its implementation and the tools and techniques needed in each one of them [6].

Given that the purpose of security methodologies consists of systematizing the process of providing specific solutions that solve security problems, thus minimizing the impact of the attacks against IS, and bearing in mind that the majority of problems take place in a similar way in different contexts, the generic solutions to these problems can be expressed as patterns [7]. These patterns will provide the methodology with great value, because they offer validated, tested and reusable solutions, whilst simultaneously compiling the knowledge of security experts [8].

Various proposals currently follow this approach, i.e., offer a systematic process for the construction of secure IS by using patterns. For example, in [11-13], the authors apply security patterns through a secure IS development method based on hierarchical architectures whose layers define the scope of each security mechanism. The main advantages of these works, which are the evolution of the same approach, are the following: in each stage, they offer the user guidelines to indicate where to apply and how to select the appropriate security pattern to satisfy the functional requirements or restrictions involved in each stage; and, they offer guidelines to identify vulnerabilities and threats in the system, along with selecting the patterns with which to mitigate them at each architectural level and at each development stage. According to these authors, one of their future works will be the implementation of this proposal in real environments. In [14], the authors put forward a method with which to integrate security patterns into a software engineering process. This proposal helps experts to close the breach between the abstract solution described in the pattern and the implementation proposed in the application. The cataloguing of different roles and the use of tools that support the systematic process make this proposal a valuable approach for real and complex organizations. However, the complexity and dynamism of these kinds of entities require an in–depth study of the detailed definition of the additional specific security tasks that are parallel to the software development in order to achieve secure IS.

After analyzing some of the proposals existing in the literature that is focused on the development of secure IS through the use of patterns, we believe that it is necessary to enrich this type of methodologies with a real and practical approach that encourages their use in a simple and systematized manner at the time of creating secure IS within real and complex organizations. This enrichment can be achieved through the detailed specification of the subjacent activities of each of the proposed stages, the elements involved and the roles taking part in each of them. It would thus be possible to provide security engineers with step by step guidelines when they confront new projects in which the existing IS within an organization need to be modified, thus mitigating the

 ISBN: 978-1-61208-158-8

errors and threats during the first stages of development of these IS and specifying the most appropriate security techniques with which to perform each of the activities and thereby maintaining the security level achieved.

We are therefore working on a methodology with which to build secure IS supported by patterns whose main objective is to offer security engineers a systematic process to be used together with the traditional software development methodologies. This will permit the construction of secure IS or maintain the security level attained in an organization's IS.

A first version of this has been published in [22], and its main characteristics are: it is based on the same stages as the classic development cycles in which we present the input and output artifacts that represent the information that is produced, modified or used for a process; the main roles taking part in each activity; and, the detailed activities of which each of the stages is composed. Another of the main contributions of this methodology is that it is focused on a central axis, which is the criticality of the assets to be protected. The use of security patterns is a fundamental contribution of our methodology since they provide structured, validated and reusable security knowledge, offering guidelines for the construction and evaluation of secure IS [9]. Finally, we would like to emphasize that this methodology is in the process of being implemented in a financial entity, and that interesting results are being obtained, which will allow us to refine, test and validate it.

In this proposal, we shall present a summary of the aforementioned methodology, whose new features consist of the in–depth study of the analysis stage in which we detail the input and output artifacts, the main roles taking part in this stage and the main activities of which it is composed. We shall support our presentation with graphical charts that represent the formalization of this methodology in SPEM (Software & Systems Process Engineering Metamodel) version 2.0 [10].

The remainder of this paper is organized as follows: Section II shows a summary of the aforementioned methodology. Section III provides a detailed description of the analysis stage. Section IV shows the current state of the application of our methodology to a real organization in the banking sector. We finish with some conclusions in Section V.

II. OVERVIEW OF THE SECURE SYSTEM DEVELOPMENT METHODOLOGY

In this section, we shall present a summary of the methodology on which we are working.

This methodology is intended to be used in parallel with and in addition to traditional software development methodologies with the purpose of strengthening the IS development process, along with being able to guarantee security against attacks and threats that place the confidentiality, availability and integrity of the assets located in these systems in jeopardy. It is also based on methodologies such as the Unified Process [15] in which a development and implementation process is carried out in an iterative and incremental manner. The advantage of this type of processes lies of the fact that we can perform successive refinements to identify risks and security critical errors during the first stages by using test mechanisms during each one of these stages to obtain a final effective and optimum solution. The structure of this systematic process follows the classic software development cycle in stages, and the main characteristics of each stage are as follows:

Analysis Stage: Set of activities centered on the achievement of security requirements according to a proposed business model. After the achievement of these requirements, a feasibility study is carried out that is focused on the assets to be protected. In this study, we analyze the risks and threats that may affect the organization's IS, and the technical possibilities with which to tackle the proposed solution in the form of security architecture.

Design Stage: Set of activities centered on the design of the final security technological solution that mitigates the risks and threats detected in the previous stage. Here, we select the structural technological elements of which the IS will be composed. We also plan the process of construction of this system in detail, and identify the tasks and personnel that will be in charge of carrying them out to obtain the proposed security architecture. All this design is focused on the criticality of the organization's assets that must be protected.

Construction Stage: The IS proposed in the previous stage will be built in this stage, and the necessary maintenance security patterns are simultaneously defined in order to guarantee the reliability and maintainability of the model built.

Test Stage: After integrating the system hardware and software components, it is necessary to guarantee their correct functioning and that they satisfy the needs indicated in the previous stages before delivering the system to the end user. The test security patterns that will show the security tests to be carried out in the future in the IS built are also defined in this stage.

Maintenance Stage: Set of specific activities that are periodically executed to guarantee that the level of security attained has not diminished over time with the appearance of new threats or security risks, or new needs not only of the end user but also of the organizational environment.

III. SECURE SYSTEM ANALYSIS

This section will show details of the analysis stage, describing its input and output artifacts together with the main roles and activities taking part in it.

The main objective of this stage is to carry out an iterative and incremental process to detect security risks that may affect the organization if the proposed business model is implemented, along with an in-depth analysis of the impact that it could have on the organization's IS.

1) Artifacts: The concept of artifact will be used as a piece of information that is produced, modified or used by a process [16]. We shall now detail the artifacts that will

 ISBN: 978-1-61208-158-8

be involved in this stage. Figure 1 shows a UML diagram containing each of these artifacts and the relations between them.

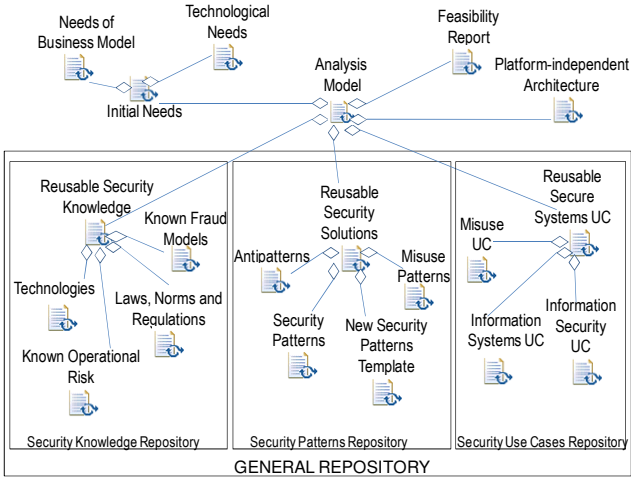

Figure 1. Analysis Stage Artifacts.

- *Initial Needs*: the input artifact for this stage, which defines the initial needs and requirements that the stakeholders need that are covered by the system. It will be composed of: the *Needs of Business Model* artifact, which specifies the business functional requirements, i.e., the initial needs required by the customer; and the *Technological Needs* artifact, which is the translation of the customer needs into technical requirements that will be used to obtain the needs at the technological infrastructure level, which must be supported by the organization's IS that cover the functional needs, the security needs and the needs of the organizational environment.
- *General Security Repository* artifact (Input/ Output): This is the most relevant artifact, both in this activity and in the whole methodology in general. It consists of an innovative element that allows this methodology to be used in real cases within complex organizations. This is owing to the fact that it is composed of several security specific repositories that collect the accumulated knowledge on this matter in different ways (real cases databases, security patterns and security use cases (UC) extracted from real IS). This artifact can also be reusable, thus minimizing the effort needed by the engineers in charge of this task to obtain validated, tested and secure IS. It will always be updated with the feedback from each of the new projects analyzed. The *General Security Repository* is formed of the following artifacts:
  - *Security Knowledge Repository*: artifact composed of different databases that compile advanced knowledge in the field of information security. This includes the *Known Fraud Models,* which is a specific repository that contains information related to known technological fraud events in the sector in which the organization operates, together with compensatory measures used to mitigate the attacks. This database will be fed by experts in the field of fraud and technological crimes based on their own experience and knowledge, and on other real data sources such as OWASP and SANS [17, 18]. The business model needs, the technical needs and the functional needs will be associated with the known fraud models to perform a study of the threats that the implementation of the model implies; *Known Operational Risk*, which is formed of a knowledge database with operational risk models and compensatory measures in relation to this kind of risk. We shall verify whether the proposed business model includes the risk of losses resulting from a lack of adaptation or from a failure in the processes, personnel or internal systems, or as a result of external events; *Laws, Norms, Regulations* collected as a repository that contains the legal restrictions that may be imposed by the country in which the organization's IS are located, the regulatory restrictions of the sector in which the organization operates and the organization's own rules in each particular project. This information will be used to carry out a study to certify that the business model does not breach any of these aspects; and finally, *Security Technologies*, which are grouped as a repository that will be verified with the business needs to define, which products/ technologies are the most appropriate to cover the needs and security risks of the proposed business model.
  - *Reusable Patterns Repository*: The artifact will include: patterns similar to *Misuse Patterns*, which relate possible attacks or misuses to the security measures that mitigate them [7]; *antipatterns* [19, 20]: *security patterns* with the structure shown in [21] that contain three levels of solutions for a specific security problem; and, *traditional security patterns* [8]. These patterns will be used to link the business model requirements, the associated security problems and the misuses that can be derived to solutions, which have already been validated and tested.
  - *Security UC Repository*: a reusable artifact that represents IS use case diagrams (UC, actors and relations), describing their behavior and capturing the requirements needed to develop a secure IS. The purpose of this artifact is to provide an IS overview through UC diagrams, capturing the main security characteristics of this kind of systems. It will be composed of other artifacts defined in the repository, which are validated and tested solutions that will assist us to improve and reduce the time and effort needed in the analysis stage. This artifact will in turn be composed of the *Reusable Secure System UC* diagrams artifact

 ISBN: 978-1-61208-158-8

that defines use case diagrams for secure IS that have been built to define common scenarios and behaviors associated with this kind of systems. This reusable artifact defines generic UC diagrams, which have been built or defined in other developments and, which are useful for this application because they contain common aspects that do not vary from one IS to another. These diagrams could also be merged with other more complex UC diagrams to represent the final IS. The *Reusable Secure System UC* artifact represents the reusable use cases, the actors and the relations between them, in order to obtain a secure IS. It is formed firstly of the *Information System UC artifact* that represents different use cases defined within an IS, which could be new use cases defined for a specific IS or reusable use cases from the repository and, which represent common functionality and technological requirements for this kind of IS. It is secondly formed of the *Information Security UC* artifact, which is similar to the previous artifact but with the difference that it captures security aspects from IS. It is in turn composed of *Security UC* and *Misuse UC,* which show security behaviors in this kind of environments, identifying possible threats and attacks against the IS itself or against the assets that must be protected, in addition to defining appropriate security requirements with which to mitigate them.

- *Analysis Model* (Output): Set of elements that are the result of the execution of the different activities in this stage. This artifact will contain the summary of the tasks developed, i.e., the initial requirements, the technological and security needs, the possible risks, threats, and legal restrictions, the security patterns identified, along with the misuse patterns, and antipatterns that are associated with the proposed business model. In addition, as output elements with own entity within this artifact, we can find the *Feasibility Report,* which is an output artifact that certifies the performance of the feasibility analysis carried out by those in charge of IS security within the organization. It presents the elements of the analysis model and its aim is to be evaluated by the relevant departments that must decide whether or not to implement the proposed business model; and finally, the *Platform- Independent Architecture* output artifact [21], which contains a high level architecture that provides a description of the security functionalities that the IS should have, independently of its technological characteristics and implementation details. More specifically, it is a conceptual description of the security mechanisms that should be incorporated into the organization's IS according to the proposed model, together with the type of relations that exist between them to guarantee the security of the organization's IS.

2) Main Roles: We shall now specify the main roles that will take part in the analysis stage, along with the functions to be developed by them (see Figure 2 in SPEM 2.0). We would like to stress that some of these roles can be executed by the same person or group of people in certain organizations.

- *Project Manager*: Role in charge of leading the project development with specific knowledge of management, and whose responsibility it is to coordinate the different security groups to obtain the performance of the Project. He will organize and supervise the Analysis Stage.
- *Security Requirements Engineer*: Role in charge of the collection of the requirements according to the proposed business model. He must be able to translate the business model needs into the technological language, extracting the main security issues that the performance of the Project implies.
- *Risk Analyst***:** In charge of leading and organizing the risk analysis related to the proposed business model. He must coordinate this task by, on the one hand studying whether the operative risk analysis will be able to detect this type of risk linked to the achievement of the business model and, on the other hand, managing the preventive analysis that will be supported by the security expert in the field of fraud and technological crimes, in addition to managing the legal analysis that will be supported by the Legal Consultant in this case.
- *Fraud Analyst*: Person in charge of leading the analysis to avoid fraud and possible technological crimes associated with the proposed business model. This stakeholder should provide current knowledge regarding fraud tendencies, new attacks, and compensatory measures to mitigate them.
- *Legal Consultant*:. Support personnel for the risk analyst who will carry out the evaluation of the risks that are inherent in the legal material with regard to existing laws, rules and regulations concerning privacy and information security.
- *Security Analyst*: Person in charge of leading and coordinating the analysis of the security requirements obtained by the *Security Requirements Engineer*. He will also be in charge of analyzing the security threats, misuses, etc., along with describing the technological security needs of the solution.
- *Security Expert*: Role specialized in determined security fields that will help the security team in very specific tasks in which the permanent members of this team do not have the necessary knowledge. He will give advice about necessary new products, proposed technologies with which to carry out the solution model, specialized tools for specific tasks, etc.
- *Security Architect*. In charge of designing the security technical architecture according to the technological and security requirements with the purpose of guaranteeing the security of the organization's IS. The

 ISBN: 978-1-61208-158-8

infrastructure designed will be implemented later, in the following stages of the methodology.

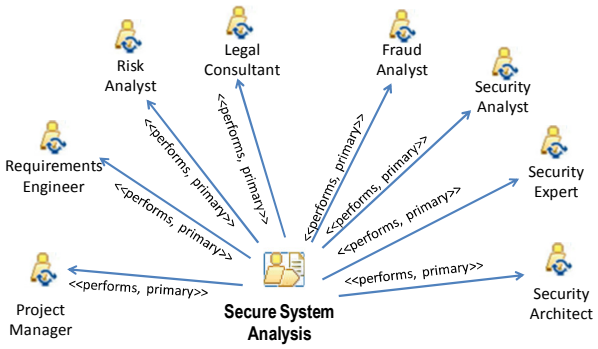

Figure 2. SPEM 2.0. view of Stakeholders

3) Analysis Stage activities: The aforementioned internal artifacts are produced in this stage. In some cases, they are the output artifacts of some activities and the input artifacts of others. The Analysis Model artifact that will serve as an input artifact for the Design stage and will certify the thorough security analysis of the proposed model will eventually be composed of all these artifacts. The main activities that will be carried out in this stage are detailed below:

- Identifying Security Systems UC: An analysis of the initial needs of the proposed business model is performed in this activity (see Figure 3). Once these needs are known, we identify the assets that must be protected and we carry out the risk analysis according to known fraud cases, operational risk associated with this business model, internal rules of the organization, regulations in the sector in which the organization operates, and laws that may affect the solution depending on where the organization's IS are located. This analysis is performed to identify the *Security UC* and *Misuse UC* that apply to the business model, and these UC can be collected in the *Security UC Repository* or can be defined by the user when a new project is analyzed. The repository will be fed back with the UC of the business model analyzed.

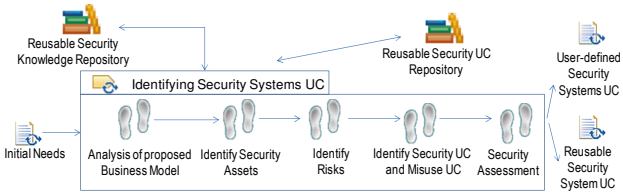

Figure 3. Activities to identify Security Systems UC in SPEM 2.0.

- Identifying Security Patterns: An association between the *Security UC* and the *security patterns* that solve the security needs specified in these use cases will be carried out. We shall also analyze the antipatterns associated with the Misuse UC to avoid security risks that may affect the solution in later stages. We shall additionally identify the technological requirements that will be mapped with the template presented in [21] to obtain the *Platform- Independent Architecture* output artifact. Finally, we shall create the *Analysis Model* that will be refined to check that there are no new security risks or technological needs, to eventually use it as an input artifact in the Design stage. Figure 4 shows this set of activities.

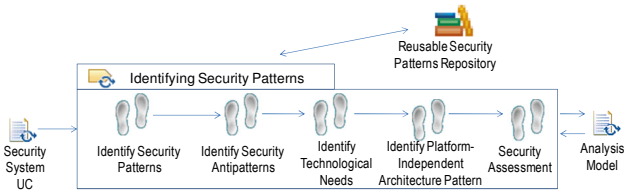

Figure 4. Activities to identify Security Patterns in SPEM 2.0

IV. DISCUSSION: CURRENT STATE OF THE METHODOLOGY

In real organizations' IS whose complexity increases daily, security aspects continue to be non-functional requirements within the software development process. This situation signifies that security aspects are, in most cases, detected in the final stages of IS construction or even when the IS is already working, thus increasing the cost and time spent on modifying the IS produced.

The experience of implementing a systematic process, which is parallel and additional to the software development process in a real organization is providing interesting results. First of all, we have observed that the different teams taking part in IS development are more and more interested in being advised by security teams in the first stages of the process, to be guided in relation to how to design the IS to avoid security threats and inherent risks. This is owing to the fact that this collaboration becomes an objective, foreseen and reliable participation, which encourages the other groups to involve the security team in all the changes that occur in the organization. Time and costs are thus saved because these changes are made during the first stages of the project, and the organization is benefited. We should also mention other benefits such as: Homogeneity between the means of working in the organization and the IS built, not only in the main headquarters but also in the acquired external entities; efficiency at the time of confronting new projects because a systematic process is available to manage each of the steps involved in building or maintaining a secure IS; cataloguing of all the provided security solutions as patterns, whose main value is to allow a fast localization and modification of IS against a threat or a suffered or foreseen risk, or its agile optimization; and, a great exportability of the means of working to any new entities acquired.

V. CONCLUSION AND FUTURE WORKS

Our research line is centered on developing a methodology for the construction of secure IS based on

 ISBN: 978-1-61208-158-8

patterns with the aim of helping security engineers in the creation of secure IS or in the maintenance of the security level attained within the IS of a real and complex organization. The systematic process, which we are working on is based on traditional development methodologies, including their key stages and dividing each of these stages into clearly defined activities that will guide engineers when adding security to an IS. In each of the stages, we show the input and output artifacts that represent the initial elements of each stage and the results, which we expect from each one of them, together with the ideal roles to develop each activity. The use of security patterns provides us with agility when solving security problems because these kinds of solutions compile the knowledge of security experts and are already validated and tested solutions that solve common security problems.

In this work, we have shown a summarized general overview of each of the stages of the methodology, and we have provided an in-depth study of the analysis stage, detailing its input and output artifacts, the roles taking part in it and the main activities of which it is composed. This presentation is supported by the formalization of the methodology in a metamodelling language (SPEM 2.0.), which has been validated and approved by the scientific community.

Finally, we should like to emphasize that the methodology proposed herein is being used in the implementation stage in a large financial entity, and this is providing us with interesting results that will help us to refine and validate it.

In future works, we shall carry out an in-depth study of the remaining stages, in addition to presenting practical examples that will certify their use in a real and complex organization. We are also working on another line consisting of building a tool to support the whole process which will serve to control each of the activities, artifacts, and people taking part in the construction of a secure IS within a real and complex organization.

ACKNOWLEDGMENT

This research has been carried out in the framework of the following projects: MODEL-CAOS (TIN2008-03582/TIN) financed by the Spanish Ministry of Education and Science, SISTEMAS (PII2I09-0150-3135) and SERENIDAD (PEII11-0327-7035) financed by the “Viceconsejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha” and the FEDER, and BUSINESS project (PET2008-0136) financed by the "Ministerio de Ciencia e Innovación”, Spain.

REFERENCES

[1] S. T. Halkidis, N. Tsantalis, A. Chatzigeorgiou, and G. Stephanides: "Architectural Risk Analysis of Software Systems Based on Security Patterns," IEEE Transactions on Dependable and Secure Computing 5, pp. 129-142, 2008.

[2] R. E. Andrew, A. P. Moore, L. Bass, M. Klein, and F. Bachmann: "Security and Survivability Reasoning Frameworks and Architectural Design Tactics," SEI, 2004.

[3] J. Jürjens: "Secure Systems Development with UML," Springer-Verlag, 2004.

[4] H. Mouratidis and P. Giorgini: "Integrating Security and Software Engineering: Advances and Future Vision," IGI Global, 2006.

[5] R. Pressman: "Software Engineering: A Practitioner's Approach," McGraw-Hill Science/Engineering/Math, 2004.

[6] T. Roberts: "Why can't we implement this SDM?," IEEE Software 16, pp. 70 - 71, 1999.

[7] E. B. Fernandez, N. Yoshioka, and H. Washizaki: "Modeling Misuse Patterns," International Conference on Availability, Reliability and Security (ARES '09), pp. 566-571, 2009.

[8] M. Schumacher, E. B. Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad: "Security Patterns: Integrating Security and Systems Engineering," 2006.

[9] R. Ortiz et al.: "Applicability of Security Patterns," The 5th International Symposium on Information Security (IS'10 - OTM'10), Crete, Greece, 2010.

[10] OMG: "Software & Systems Process Engineering Meta-Model Specification (SPEM) 2.0," 2008.

[11] E. B. Fernandez, M. M. Larrondo-Petrie, T. Sorgente, and M. VanHilst: "Chapter 5, A methodology to develop secure systems using patterns," Integrating security and software engineering: Advances and future vision, IDEA Press, pp. 107-126, 2006.

[12] E. B. Fernandez: "Security Patterns and A Methodology to Apply them," Security and Dependability for Ambient Intelligence, pp. 37-46, 2009.

[13] E. B. Fernandez et al.: "Using security patterns to develop secure systems," H. Mouratidis (ed.): Software Engineering for Secure Systems: Industrial and Research Perspectives, pp. 16-31, 2009.

[14] F. Sanchez-Cid and A. Maña: "SERENITY Pattern-Based Software Development Life-Cycle," 19th International Workshop on Database and Expert Systems Application (DEXA '08), pp. 305-309, Turin, 2008.

[15] P. Kruchten: "The Rational Unified Process: An Introduction," Addison-Wesley, Boston, 2000.

[16] D. G. Rosado, E. Fernández-Medina, J. López, and M. Piattini: "Analysis of Secure Mobile Grid Systems: A systematic approach," Information and Software Technology, 2010.

[17] The Open Web Application Security Project (OWASP) <http://www.owasp.org> 01.04.2011

[18] SANS - Computer Security Training, Network Research & Resources <http://www.sans.org> 05.04.2011

[19] M. Kis: "Information Security Antipatterns in Software Requirements Engineering," 9th Conference of Pattern Languages of Programs, 2002.

[20] J. Král and M. Zemlicka: "Popular SOA Antipatterns," Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, pp.271-276, Athens, Greece, 2009.

[21] S. Moral-García et al.: "A New Pattern Template to Support the Design of Security Architectures," The Second International Conferences of Pervasive Patterns and Applications, Lisbon, Portugal, 2010.

[22] R. Ortiz, S. Moral-Rubio, J. Garzás, and E. Fernández-Medina: "Towards a Pattern-Based Security Methodology to Build Secure Information Systems," 8th International Workshop on Security in Information Systems, Beijing, China, 2011.

 ISBN: 978-1-61208-158-8