PATTERNS 2010

The Second International Conferences on Pervasive Patterns and Applications

November 21-26, 2010 - Lisbon, Portugal

ComputationWorld 2010 Editors

Ali Beklen, IBM Turkey, Turkey

Jorge Ejarque, Barcelona Supercomputing Center, Spain

Wolfgang Gentzsch, EU Project DEISA, Board of Directors of OGF, Germany

Teemu Kanstren, VTT, Finland

Arne Koschel, Fachhochschule Hannover, Germany

Yong Woo Lee, University of Seoul, Korea

Li Li, Avaya Labs Research - Basking Ridge, USA

Michal Zemlicka, Charles University - Prague, Czech Republic

PATTERNS 2010

Foreword

The Second International Conferences on Pervasive Patterns and Applications [PATTERNS 2010], held between November 21 and 26 in Lisbon, Portugal, targeted the application of advanced patterns, at-large. In addition to support for patterns and pattern processing, special categories of patterns covering ubiquity, software, security, communications, discovery and decision were considered. As a special target, the domain-oriented patterns cover a variety of areas, from investing, dietary, forecast, to forensic and emotions. It is believed that patterns play an important role on cognition, automation, and service computation and orchestration areas. Antipatterns come as a normal output as needed lessons learned.

We take here the opportunity to warmly thank all the members of the PATTERNS 2010 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to PATTERNS 2010. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the PATTERNS 2010 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that PATTERNS 2010 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the areas pervasive patterns and applications.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed the beautiful surroundings of Lisbon, Portugal.

PATTERNS 2010 Chairs:

Eva Blomqvist, STLab ISTC-CNR, Italy
Markus Goldstein, DFKI (German Research Center for Artificial Intelligence GmbH), Germany
Teemu Kanstren, VTT, Finland
Fritz Laux, Reutlingen University, Germany
Herwig Manaert, University of Antwerp, Belgium
Guenter Neumann, DFKI (German Research Center for Artificial Intelligence GmbH), Germany
Juan Pelaez, U.S. Army Research Laboratory, USA
Lei Shu, Osaka University, Japan
Zhenzhen Ye, iBasis, Inc., Burlington, USA
Michal Zemlicka, Charles University - Prague, Czech Republic

PATTERNS 2010

Committee

PATTERNS Advisory Chairs

Academia
Herwig Manaert, University of Antwerp, Belgium
Michal Zemlicka, Charles University - Prague, Czech Republic
Fritz Laux, Reutlingen University, Germany
Lei Shu, Osaka University, Japan

Research Institutes
Teemu Kanstren, VTT, Finland
Guenter Neumann, DFKI (German Research Center for Artificial Intelligence GmbH), Germany
Juan Pelaez, U.S. Army Research Laboratory, USA
Eva Blomqvist, STLab ISTC-CNR, Italy
Markus Goldstein, DFKI (German Research Center for Artificial Intelligence GmbH), Germany
Zhenzhen Ye, iBasis, Inc., Burlington, USA

PATTERNS 2010 Technical Program Committee

Junia Anacleto, Federal University of Sao Carlos, Brazil
Francesca Arcelli, University of Milano Bicocca, Italy
Li Bai, University of Nottingham, UK
Eva Blomqvist, STLab ISTC-CNR, Italy
Dave Bustard, University of Ulster-Coleraine, UK
Jean-Charles Créput, Universite de Technologie de Belfort Montbeliard (UTBM), France
Angélica de Antonio, Universidad Politécnica de Madrid (UPM), Spain
Alessandra Scotto di Freca, University of Cassino, Italy
Kamil Dimililer, Near East University, Turkey
Petre Dini, Concordia University, Canada / IARIA, USA
Jürgen Ebert, Universität Koblenz-Landau, Germany
Eduardo B. Fernandez, Florida Atlantic University - Boca Raton, USA
Francesco Fontanella, Università degli Studi di Cassino, Italy
Harald Gjermundrod, University of Nicosia, Cyprus
Markus Goldstein, DFKI (German Research Center for Artificial Intelligence GmbH), Germany
Carmine Gravino, University of Salerno - Fisciano, Italy
Yann-Gaël Guéhéneuc, École Polytechnique de Montréal, Canada
Mohamed Farouk Abdel Hady, University of Ulm, Germany
Jon Hall, The Open University, UK
Chih-Cheng Hung, Southern Polytechnic State University, USA
Shareeful Islam, Technische Universität München, Germany
Hermann Kaindl, TU-Wien, Austria
Teemu Kanstren, VTT, Finland

Christian Kruschitz, University of Klagenfurt, Austria
Richard Laing, The Robert Gordon University, Aberdeen, UK
Fritz Laux, Reutlingen University, Germany
Gyu Myoung Lee, Institut Telecom, Telecom SudParis, France
Haim Levkowitz, University of Massachusetts Lowell, USA
Chendong Li, University of Connecticut, USA
Chung-Horng Lung, Carleton University - Ottawa, Canada
Herwig Manaert, University of Antwerp, Belgium
Cristina Marrocco, University of Cassino, Italy
Paul Marshall, The Open University, UK
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Murali Medidi, Boise State University, USA
Gerrit Meixner, German Research Center for Artificial Intelligence (DFKI)- Kaiserslautern, Germany
Ivan Mistrík, Independent Consultant. Heidelberg, Germany
Guenter Neumann, DFKI (Deutsches Forschungszentrum fuer Kuenstliche Intelligenz GmbH), Germany
Javier Ortega-Garcia, Universidad Autonoma de Madrid, Spain
Christian Percebois, University of Toulouse, IRIT, France
Juan Pelaez, U.S. Army Research Laboratory, USA
Agostino Poggi, Università degli Studi di Parma, Italy
Mar Pujol, Universidad de Alicante, Spain
Caludia Raibulet, University of Milano, Italy
Yonglin Ren, University of Ottawa, Canada
Joel Rodrigues, Instituto de Telecomunicações, University of Beira Interior, Portugal
Lei Shu, Osaka University, Japan
Vladimir Stantchev, Berlin Institute of Technology, Germany
Horia-Nicolai Teodorescu, "Gheorghe Asachi" Technical University of Iasi / Romanian Academy, Romania
Laurent Wendling, University Paris Descartes (Paris V), France
Reuven Yagel, Jerusalem College of Engineering, Israel
Zhenzhen Ye, iBasis, Inc., Burlington, USA
Michal Zemlicka, Charles University - Prague, Czech Republic

PATTERNS 1: Software patterns

A Formal Language of Pattern Compositions

Ian Bayley, Oxford Brookes University, UK
Hong Zhu, Oxford Brookes University, UK

PLOP: A Pattern for Learning Objects for Programming

Luis Alberto Alvarez-Gonzalez, Universidad Austral de Chile, Chile
Valeria Henríquez, Universidad Austral de Chile, Chile
Erick Araya, Universidad Austral de Chile, Chile
Fabiola Cárdenas, Universidad Austral de Chile, Chile

Pattern Catalog for Capability Diagnostics and Improvement of Service-oriented Enterprise Architectures

Alfred Zimmermann, Reutlingen University, Germany
Eckhard Ammann, Reutlingen University, Germany
Fritz Laux, Reutlingen University, Germany

Towards A Taxonomy of Dynamic Invariants in Software Behaviour

Teemu Kanstrén, VTT, Finland

PATTERNS 2: Basics on patterns I

Towards a Common Pattern Language for Ubicomp Application Design

René Reiners, Fraunhofer FIT, Germany

Detection of Generic Micro-architectures on Models

Cédric Bouhours, IRIT - MACAO team, France
Hervé Leblanc, IRIT - MACAO team, France
Christian Percebois, IRIT - MACAO team, France
Thierry Millan, IRIT - MACAO team, France

A Benchmark Platform for Design Pattern Detection

Francesca Arcelli Fontana, University of Milano Bicocca, Italy
Marco Zanoni, University of Milano Bicocca, Italy
Andrea Caracciolo, University of Milano Bicocca, Italy

Tissue Classification from CT of Liver Volumetric Dataset Using 3D Relational Features

Wan Nural Jawahir Hj Wan Yussof, Albert-Ludwigs-Universität Freiburg, Germany
Hans Burkhardt, Albert-Ludwigs-Universität Freiburg, Germany

PATTERNS 3: Basics on patterns II and Security patterns

Highlighting the Essentials of the Behaviour of Reactive Systems in Test Descriptions Using the Behavioural Atomic Element

Lars Ebrecht, DLR - German Aerospace Center, Germany
Karsten Lemmer, DLR- German Aerospace Center, Germany

CUX Patterns Approach: Towards Contextual User Experience Patterns

Marianna Obrist, HCI & Usability Unit, ICT&S Center, University of Salzburg, Austria
Daniela Wurhofer, HCI & Usability Unit, ICT&S Center, University of Salzburg, Austria
Elke Beck, HCI & Usability Unit, ICT&S Center, University of Salzburg, Austria
Manfred Tscheligi, HCI & Usability Unit, ICT&S Center, University of Salzburg, Austria

A new pattern template to support the design of security architectures

Santiago Moral-García, University Rey Juan Carlos, Spain
Roberto Ortiz, Group S21sec Gestión S.A., Spain
Santiago Moral-Rubio, BBVA, Spain
Belén Vela, University Rey Juan Carlos, Spain
Javier Garzás, University Rey Juan Carlos, Spain
Eduardo Fernández-Medina, University of Castilla-La Mancha, Spain

A Pattern Collection for Privacy Enhancing Technology

Cornelia Graf, CURE- Center for Usability Reserach & Engineering, Austria
Peter Wolkerstorfer, CURE- Center for Usability Reserach & Engineering, Austria
Arjan Geven, CURE- Center for Usability Reserach & Engineering, Austria
Manfred Tscheligi, CURE- Center for Usability Reserach & Engineering, Austria

PATTERNS 4: Patterns at work

Multiple Pattern Matching

Stephen Fulwider, University of Central Florida, USA
Amar Mukherjee, University of Central Florida, USA

Definition and Reuse of Analysis Patterns for Real-Time Applications

Hela Marouane, MIRACL_ISIMS, Tunisia
Saoussen Rekhis, MIRACL_ISIMS, Tunisia
Rafik Bouaziz, MIRACL_ISIMS, Tunisia
Claude Duvallet, LITIS, France
Bruno Sadeg, LITIS, France

Luis Alberto Alvarez-Gonzalez

PLOP: A Pattern for Learning Objects for Programming

Eckhard Ammann

Pattern Catalog for Capability Diagnostics and Improvement of Service-oriented Enterprise Architectures

Erick Araya

PLOP: A Pattern for Learning Objects for Programming

Francesca Arcelli Fontana

A Benchmark Platform for Design Pattern Detection

Ian Bayley

A Formal Language of Pattern Compositions

Elke Beck

CUX Patterns Approach: Towards Contextual User Experience Patterns

Rafik Bouaziz

Definition and Reuse of Analysis Patterns for Real-Time Applications

Cédric Bouhours

Detection of Generic Micro-architectures on Models

Hans Burkhardt

Tissue Classification from CT of Liver Volumetric Dataset Using 3D Relational Features

Andrea Caracciolo

A Benchmark Platform for Design Pattern Detection

Fabiola Cárdenas

PLOP: A Pattern for Learning Objects for Programming

Claude Duvallet

Definition and Reuse of Analysis Patterns for Real-Time Applications

Lars Ebrecht

Highlighting the Essentials of the Behaviour of Reactive Systems in Test Descriptions Using the Behavioural Atomic Element

Eduardo Fernández-Medina

A new pattern template to support the design of security architectures

Stephen Fulwider

Marianna Obrist

CUX Patterns Approach: Towards Contextual User Experience Patterns

Roberto Ortiz

A new pattern template to support the design of security architectures

Christian Percebois

Detection of Generic Micro-architectures on Models

René Reiners

Towards a Common Pattern Language for Ubicomp Application Design

Saoussen Rekhis

Definition and Reuse of Analysis Patterns for Real-Time Applications

Bruno Sadeg

Definition and Reuse of Analysis Patterns for Real-Time Applications

Manfred Tscheligi

CUX Patterns Approach: Towards Contextual User Experience Patterns
A Pattern Collection for Privacy Enhancing Technology

Belén Vela

A new pattern template to support the design of security architectures

Peter Wolkerstorfer

A Pattern Collection for Privacy Enhancing Technology

Daniela Wurhofer

CUX Patterns Approach: Towards Contextual User Experience Patterns

Marco Zanoni

A Benchmark Platform for Design Pattern Detection

Hong Zhu

A Formal Language of Pattern Compositions

Alfred Zimmermann

Pattern Catalog for Capability Diagnostics and Improvement of Service-oriented Enterprise Architectures

A New Pattern Template to Support the Design of Security Architectures

Santiago Moral-García[1], Roberto Ortiz[2], Santiago Moral-Rubio[3], Belén Vela[1], Javier Garzás[1, 5],
Eduardo Fernández-Medina[4]

*(1) Kybele Group. Dep. of Computer Languages and Systems II. University Rey Juan Carlos,
Madrid (Spain).
{santiago.moral, belen.vela, javier.garzas} @urjc.es
(2) S21secLabs. SOC. Group S21sec Gestión S.A, Madrid (Spain).
r.ortizpl@gmail.com
(3) Dep. Logical Security. BBVA, Madrid (Spain).
santiago.moral@grupobbva.es
(4) GSyA Research Group. Dep. of Information Technologies and Systems.
University of Castilla-La Mancha, Ciudad Real (Spain).
eduardo.fdezmedina@uclm.es
(5) Kybele Consulting, Madrid (Spain).
javier.garzas@kybeleconsulting.com*

Abstract—**The vast majority of current security patterns are oriented towards the production of security mechanisms, such as secure access systems or secure authentication systems. This type of patterns may be extremely useful for those security engineers who work on the production of this kind of mechanisms for large companies (Oracle, Microsoft, IBM, Google, Cisco, etc.), but they cannot be applied by a wide sector of security engineers who work in the development of security architectures. This is owing to the fact that these patterns do not consider aspects of the real complex system in which they will be installed. In order to complement security patterns and make them more applicable to security architecture design environments, in this paper we will propose a new description template of security patterns. The solution provided by this new template is oriented towards the architecture and technologies that should be used to design security architectures in real complex systems.**

Keywords: information security engineering; security architectures; security technologies; security patterns; real environments.

I. INTRODUCTION

Organizations currently require to guarantee availability, integrity and confidentiality of their assets [16]. In view of the fact that the realization of this task should consider the constant evolution of the organization's setting [27], we should specifically consider the variation between people, technologies, risks, processes, volumes of information, business strategies, etc. Therefore, there is a need to adapt the organization to all these changes in order to attain the objective of guaranteeing the fundamental security properties for its assets [20]. It is not easy for an organization to evaluate its level of risk and adapt itself to permanent changes. It is therefore vital for it to seek support from a security architecture [3] in order to mitigate the impact of these changes and thus minimize the risks associated with each of them.

The concept of security architecture can be defined as the practice of applying a structured, coordinated, rigorous method with the intention of discovering an organization's structure, bearing in mind human resources, business processes and technologies, i.e., all the elements that are involved in the organization to provide its systems with security and thus ensure the safety of its assets [19]. Security architectures are installed with the intention of minimizing the risks associated with the use of information technologies as well as optimizing an organization's business processes and strategies. If this objective is to be achieved, it is necessary to establish a set of technological infrastructure controls with which to identify the security mechanisms that are needed to define the system's security.

The security mechanisms used in security architectures are artifacts which have been designed to detect problems, prevent risks or make immediate corrections in order to avoid any undesirable events which may make security vulnerable [26].

After carrying out a systematic review of the literature related to security patterns, we have found out that the vast majority of patterns which are currently in use are focused on supporting the construction of new security mechanisms [9, 24, 28]. These patterns are a useful support for those engineers who work on developing security mechanisms which are the basic elements of an architecture [22, 7]. However, it is difficult to apply most of them to those work environments that are focused on the analysis and design of security architectures, since they do not consider the details of installing the solution in real complex systems [9, 28, 18]. We understand a real complex system to be all those elements that

are involved in an organization, i.e., human resources, business processes and technologies.

We have therefore detected the need to discover structured solutions in the form of patterns, or the evolution of existing security patterns, to support information security engineers in the analysis and later design of security architectures which are used in an organization's real complex systems.

If security patterns are to be applicable to the sector of security engineers who design secure architectures in real systems, and confidentiality, integrity and availability of the organization's information assets are to be ensured, then it is necessary to resolve a series of lacks which have been detected. These solutions are shown as follows:

- Detailing the information assets, which the deployment of the pattern attempts to ensure, and the level of criticality of these assets.
- Detailing what an organization is protecting with the installation of the pattern.
- Including the deployment details in a real environment, bearing in mind the architecture and technologies that should be used to develop the solution in a satisfactory manner.
- Carrying out a qualitative analysis of the most important technological aspects with regard to the proposed solution (memory consumed, processing capacity, etc.).
- Bearing in mind different countries' rules and regulations with regard to the information assets that they wish to conserve. It may be that a solution which is legal in one country is not legal in another.

The lacks detected in current security patterns have led us to the belief that it is necessary to define a new description template of security patterns with which to resolve these limitations. This new template is characterized by the fact that it includes all the aspects which are necessary for a simple and reusable definition of security architectures. The definition of this template provides a step by step description of the architecture's design, and is linked to the necessary security requirements in relation to the criticality of the assets to be protected, known incidents, the systems involved in the solution, the necessary volumetric, and other variables associated with the environment such as the complexity of deployment, the use and maintenance of the solution, the regulations of the country in which the solution will be installed, and associated costs.

The remainder of this paper is organized as follows. Section II provides a description of the goodness of security patterns and shows related works in order to represent these patterns. Section III presents a new description template of security patterns. Section IV states our general conclusions with regard to the approach, and puts forward our future work.

II. SECURITY PATTERNS

A security pattern describes a recurrent security problem which arises in a specific context, and provides a well tested generic scheme as a solution to that problem [12]. One of the main advantages of patterns is that they combine experience in

the design of information system [10], thus making them more efficient. Patterns are a literary format with which to capture the knowledge and experience of security experts, resulting in a structured document in the form of a template to which the security experts' knowledge is transferred [21].

The first authors to propose security patterns were Yoder and Barcalow in 1997 [29]. The number of security patterns which have been published has increased considerably since then [22, 11, 30].

A great heterogeneity exists between the different descriptions found in each of the security patterns published [21, 15, 2, 13, 17]. This is because the authors who describe the security patterns that have been discovered have historically used different description templates to represent them. The most frequently used templates are those proposed by the Gang of Four [14], which have been adapted to describe security patterns, the template proposed by Buschmann et al. [4], the template proposed in the SERENITY project [23], and that proposed by Alexander [1]. Apart from these, other templates for the description of patterns have also been published, but their use is not massively extended yet. One example of these is that proposed in [25], in which the security patterns are represented as events calculus. Recent years have seen the proposal of other types of more specific security patterns, such as attack patterns [8] or misuse patterns [13].

As shown in [17], although the various authors who describe security patterns do not use a standardized description template, the majority of the description templates of these patterns have the following trio of elements in common: the context in which the pattern has been discovered; the security problem that is attempted to be resolved within the context put forward; and the forces that affect the solution. The solution is conditioned by the associated forces, and these are expressed through UML diagrams which model this solution [13].

In order to resolve the lacks detected in current security patterns and thus support information security engineers when analyzing and designing organizations' security architectures, we propose a new description template of security patterns. The template proposed below is intended to be an easy-to-use guideline which will allow both experts and non-experts in security to access a structured and methodical document with which to resolve security problems in the real complex systems of the organizations in which they work.

III. A NEW DESCRIPTION TEMPLATE OF SECURITY PATTERNS

In this section, we shall set out the new description template of security patterns, explaining its characteristics and the contribution that it will make to the scientific community in the field of security. We shall then go on enumerating and detailing each of the description elements of the proposed template.

A security pattern focused on the development of security architectures describes a valid generic path that assists security engineers in making analysis and designing decisions when

confronting the development of a secure architecture, which will resolve a real security deficiency in an information system. In order to obtain the maximum applicability within an organization, the proposed solution is oriented towards the architecture and technology that must be used in that organization to guarantee the security of the information assets associated with the deficiencies that we intend to resolve.

The new template will be described with the description elements from the description template proposed by Buschmann et al. [4] and the template proposed in the SERENITY project, used in [5], together with new description elements which are necessary to provide security experts and non-experts with a template to support the design of security architectures.

One of the main contributions of this approach is that the proposed solution provides security engineers with three complementary levels or *viewpoints:* platform independent level, platform specific level and product dependent level. This solution model manages to separate the implementation of the system's functionality specification over a platform in a specific technology. This allows differentiating the functionality that the system must satisfy and the technologies that could be implemented to develop the solution. Security engineers can also visualize the evolution of the solution from abstract models to real implementations in the complete system.

Figure 1 (below) shows a graphical representation of the solution levels.
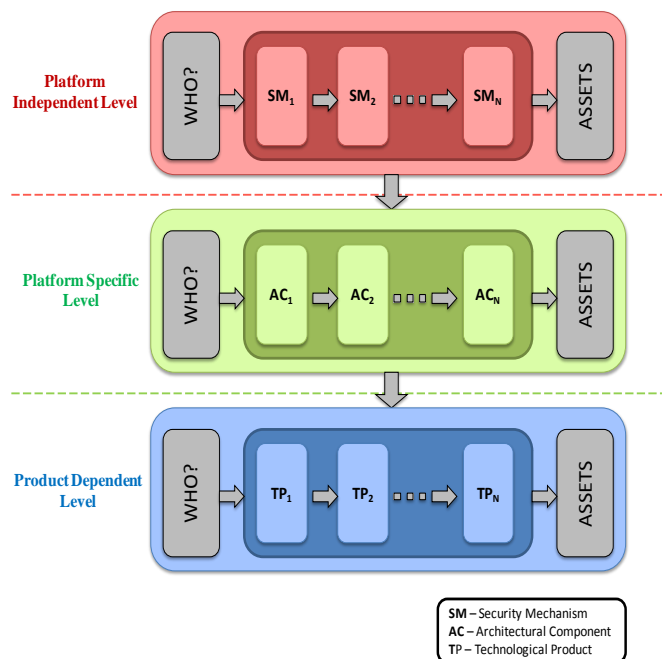

Figure 1. Abstraction levels of the solution.

As the figure above shows, all security systems must consider which information assets they intend to protect and who will have access to them.

We shall now provide a short description of each of the

abstraction levels shown in Figure 1, and how the transformations through which to move from one level to the following should be carried out, illustrating the new elements needed to be incorporated or considered.

Platform Independent Level: this level provides a description of the security functionalities that the system should have, independently of its technological characteristics and implementation details. More specifically, a conceptual description of the security mechanisms that should be incorporated into the system is provided, along with the type of relationship that exists among them. The elements that should appear at this level are security patterns which are oriented towards the development of security mechanisms. A good guideline which can be used as a basis for discovering the type of patterns that are necessary is the guideline developed by Schumacher et al. in [22].

Platform Specific Level: the solution should be defined at this level, detailing the architecture or platform to which it will be applied. It is also necessary to set out how the necessary security mechanisms should be situated, through the presentation of an optimum security architecture with which to resolve the problem, independently of the technology used to protect the organization's systems. Given that security problems have repercussions on specific technological architectures, the same platform independent model can be instantiated N times, since it corresponds with different technological architectures. The security mechanisms described at the independent level become architectural components at this level.

Product Dependent Level: it is necessary to install the platform specific model into a specific architecture at this level, to implement it with technological products that are already available. Each of the architectural components can, therefore, be transformed into N technological products. The technological products must be valid products made by known manufacturers in the security industry. The final solution may vary significantly depending on the technologies used. This level should be independent of the information system's technological conditions. This view of the solution is very practical since it shows the user the different technologies that already exist on the market and that are oriented towards resolving the given problem.

This manner of structuring the solution provides a clear example of the steps that must be followed to implement the pattern, signifying that both experts and non-experts can understand the solution and know how to deploy it in a real system.

A further implicit property of this description template is its associated *decision path*. This element is of great assistance when selecting the most appropriate pattern with which to resolve a determined problem. The following five levels have been proposed in the decision path in order to classify the patterns that are associated with a discovered security deficiency:

1) What is the state of the information, programs or configurations that need to be protected? The possible states are the following:

a) Stored: These are found in a data base.

b) Transit: Through a transfer to another company or service. There is a movement of information.

c) Access: The information is being accessed.

2) Who accesses the information that we wish to protect? The people who can access the information are:

a) The organization's internal users.

b) External users or customers.

c) Computing staff during their work. This type of user is special since he can access data, applications and systems without using the security mechanisms which have been designed in the applications utilized by the final users.

3) How is the information accessed? or *What is the means of access?* In short, the information can be accessed in the following manners:

a) Directly: By accessing the data directly without any limitations on the use that is made of them.

b) Through an application: By applying business logic to the use, through which the information is shown.

4) Where is the information accessed from? It is basically accessed from two places:

a) Within the organization, i.e., all the technological spheres that are governed by the same security policies.

b) Outside the organization: where it is not possible to ensure the fulfillment of the same security policies that appear in the organization in which the assets are located.

5) Who manages the means used to access the information that needs to be protected?

a) The person responsible for security who will use the pattern and will be legally authorized to manage the systems' security.

b) Any other person who does not belong to the organization or does not have legal authorization to manage the system's security.

This *decision path* can be used to verify what type of problem, in general terms, will be resolved with the pattern discovered, i.e., two security patterns that respond identically to the same path resolve problems of the same nature, and could thus be alternatives to the same problem.

With regard to the elements described in the template, it is also necessary to emphasize that they do not describe the security vulnerabilities that may affect the information system in which the solution is installed. This is owing to the fact that new vulnerabilities frequently appear and the pattern must be constantly modified. We consider that the technologies themselves should be updated each time a new vulnerability is encountered, and that in this case it should be the manufacturer who updates them, or the security administrator who incorporates new rules into the security technologies used, if the impact of these vulnerabilities is to be minimized. This new template of security patterns therefore considers that vulnerabilities appear in all technologies on a permanent basis,

and this concept forms a part of the pattern's considerations. The greater a technology's exposure to public networks, the higher its level of weakness. All security architectures will therefore be designed by bearing in mind that critical vulnerabilities repeatedly appear in all technologies.

The template proposed for the description of security patterns focused on the design of security architectures will be shown as follows. We must emphasize that this template is used to evolve existing security patterns, since it maintains the same base structure as their description, and it is only necessary to add the new elements that are proposed. The template that is proposed consists of the following elements:

A. Name

The pattern's name should represent the problem that it is attempting to resolve. This name must also be unique within the sphere of this type of patterns.

B. Context

The context provides a generic description of the setting, both at user level and system level, and includes the conditions under which the described pattern should be applied.

C. Problem

This describes the situation which has led to the necessity to apply a series of security mechanisms in order to obtain an optimum solution, and it basically details the reasons for the problem. It should also indicate the following questions:

- Which assets need to be protected? Information, programs and/or configurations.
- What are we protecting ourselves from? Information leaks, massive attacks, etc.
- Which security properties do we intend to conserve? Confidentiality, integrity, availability, auditability and/or non-repudiation.

D. Known incidents

It consists of a description of real cases of known security incidents, in relation to the problem posed that the implementation of the pattern intends to resolve. These incidents can be easily located on the Internet on specialized sites [6], which collect this type of events and specify when they occurred, how they occurred and what their impact was.

E. Decision Path

This element should describe all general levels of the state of the assets that need to be protected (previously described). This will make it possible to determine which pattern should be used to resolve a specific security problem. The objective of this descriptive element is to be able to develop a methodology based on security patterns, on the basis that the pattern's definition itself develops its own path in the decision tree.

F. Solution

This element describes the solution in accordance with the scenario and the problem being considered. This solution must be expressed at three different abstraction levels, as previously

shown. It is first necessary to set out the solution for a platform independent level, showing the security mechanisms that must be used and the relationship that exists among them. This first level is then transformed into a second level, called platform specific level, which refers to the technological architecture proposed to resolve the given problem. The second level is finally transformed into a third level, called product dependent level, which shows a proposal for the technologies that can be used to implement the solution proposed by the described pattern. These technologies must be considered trustworthy by the Security Engineering sector.

Once these three levels have been developed, the solution should be complemented with a UML sequence diagram that is oriented towards the product dependent level, and that shows and describes in detail what the sequence of optimum processes to carry out the solution is.

G. Considerations

It is necessary to carry out a qualitative analysis of the solution in relation to the critical parameters found in the real complex system: a) storage; b) memory consumed; c) frequency with which the systems, technologies and applications are patched up; d) process capacity; e) complexity for final user; f) complexity for security/systems administrator; g) complexity of log management; h) broadband consumed; i) complexity for massive use of solution; j) cost of installing solution; and k) solution fulfillment guarantees. It is necessary to decide whether each of these aspects is qualitatively altered in a Null (0), Low (1), Medium (2) or High (3) manner when deploying the solution in a real information system.

These decisions will assist in the evaluation of whether or not the implementation of the solution is appropriate for the organization's current situation. This is particularly true when considering the cost parameters and fulfillment conditions since excessive costs and an inability to ensure the fulfillment of the solution might be the main cause of any solution being rejected.

H. Rules and Regulations

If the adoption of a predefined solution in the form of a pattern in a real environment is desired, it is necessary to consider the regulations of the country in which the solution is intended to be installed, with regard to the information activities that need to be protected. We must also bear in mind the rules associated with these regulations which must be fulfilled by the proposed solution for it to be correct both juridical and legally. For example, Argentina does not permit the movement of information related to people who reside in that country and a solution which does not fulfill this regulation could not, therefore, be installed.

I. Benefits

A short description of a solution's goodness with regard to the sphere and specific context in which the pattern is developed.

J. Consequences

This element describes the consequences of adopting a pattern as a solution in a real information system. An analysis of the risks that the organization runs if it does not adopt this solution must also be carried out. To do this, it is necessary to describe the following consequences:

- Negative consequences of adopting the solution.
- Consequences of not adopting the solution.

K. Alternatives

The majority of security deficiencies can be resolved in different ways, and this section should therefore describe other solutions that can be used to resolve the considered problem. These alternatives may differ from the pattern described at the technological level, at the architectural level or even in the security mechanisms used to guarantee the information assets that are at risk.

IV. CONCLUSION

In this paper we have presented a new description template of security patterns. To do this, we have provided a brief introduction to security patterns and their related works which put forward pattern description templates. We have then set out the reasons why security patterns focused on designing security architectures are necessary.

Existing security patterns are currently focused on supporting security engineers in the construction of security mechanisms. This type of patterns can rarely be applied by those security engineers who are dedicated to the analysis and later design of security architectures in real systems. This limited applicability results from the fact that current patterns: a) do not contemplate the impact of the systems involved in the solution; b) do not define the assets that must be protected; c) do not classify these assets according to their criticality; d) do not consider the restrictions involved in applying them in the different countries where we may wish to install the solution; e) do not consider the complexity of deployment, use and maintenance of the solution by the engineers in charge of them; f) do not define the reason why it is necessary to protect the assets; g) do not consider the impact of parameters on the system in which the solution will be installed; and h) do not put forward a real use case to provide both experts and non-experts in security with an example with which they can compare their problem. All of the aforementioned reasons led us to the belief that it was necessary to state a new description template of security patterns oriented towards resolving the need to obtain structured, valid and reusable solutions with which to support information security engineers in the analysis and design of security architectures in real complex systems.

We are currently working on the description of new security patterns focused on designing security architectures. We are also attempting to refine existing security patterns to make them applicable to the design of security architectures. Finally, we are defining a use methodology for this security patterns to allow both experts and non-experts in security to apply

security to their systems in an easy, rapid and optimum manner.

ACKNOWLEDGEMENTS

REFERENCES

[1] C. Alexander, S. Ishikawa, and M. Silverstein "A Pattern Language: Towns, Buildings, Constructions" Oxford University Press, 1977.

[2] Z. Anwar, W. Yurcik, R. E. Johnson, M. Hafiz, and R. H. Campbell "Multiple design patterns for voice over IP (VoIP) security" in Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International, 2006.

[3] A. Barth, C. Jackson, and C. Reis "The Security Architecture of the Chromium Browser" Technical Report 2008.

[4] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. "Pattern-oriented software architecture: A system of patterns" Wiley, 1996.

[5] A. Cuevas, P. El Khoury, L. Gomez, and A. Laube "Security Patterns for Capturing Encryption-Based Access Control to Sensor Data" in SECURWARE '08. Second International Conference on Emerging Security Information, Systems and Technologies, 2008, pp. 62-67.

[6] "DATALOSS db - Open Security Fundation", http://datalossdb.org/, 2010

[7] E. Fernandez "Security Patterns and Secure Systems Design" in Dependable Computing, 2007, pp. 233-234.

[8] E. Fernandez, J. Pelaez, and M. Larrondo-Petrie "Attack Patterns: A New Forensic and Design Tool" in Advances in Digital Forensics III, 2007, pp. 345-357.

[9] E. Fernandez, H. Washizaki, N. Yoshioka, A. Kubo, and Y. Fukazawa "Classifying Security Patterns" in Progress in WWW Research and Development, 2008, pp. 342-347.

[10] E. B. Fernández "Security patterns and secure systems design" ACM Southeast Regional Conference 2007.

[11] E. B. Fernandez and J. L. Ortega-Arjona "The Secure Pipes and Filters Pattern" in DEXA '09. 20th International Workshop on Database and Expert Systems Application, 2009, pp. 181-185.

[12] E. B. Fernandez, J. C. Pelaez, and M. M. Larrondo-Petrie "Security Patterns for Voice over IP Networks" in ICCGI 2007. International Multi-Conference on Computing in the Global Information Technology, 2007, pp. 33-33.

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

[14] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides "Design Patterns: Elements of Reusable Object Oriented Software" Addison Wesley, 1995.

[15] J. Garzás and M. Piattini "Object Oriented Microarchitectural Design Knowledge" IEEE Software, pp. 28-33, 2005.

[16] D. M. Kienzle, M. C. Elder, D. Tyree, and J. Edwards-Hewitt "Security patterns repository, version 1.0" 2006.

[17] S. Moral-Garcia, R. Ortiz, B. Vela, J. Garzás, and E. Fernández-Medina "Patrones de Seguridad: ¿Homogéneos, validados y útiles?" in RECSI XI Tarragona, Spain, submit accepted.

[18] R. Ortiz, S. Moral-García, S. Moral-Rubio, B. Vela, J. Garzás, and E. Fernández-Medina "Applicability of Security Patterns" The 5th International Symposium on Information Security (IS'10 - OTM'10), 2010 - submit accepted.

[19] "OSA - Open Security Architecture", http://www.opensecurityarchitecture.org/cms/index.php, 2010

[20] D. G. Rosado, C. Gutiérrez, E. Fernández-Medina, and M. Piattini "Security patterns and requirements for internet-based applications" Internet Research: Electronic Networking Applications and Policy, 2006.

[21] M. Schumacher "B. Example Security Patterns and Annotations" in Security Engineering with Patterns, 2003, pp. 171-178.

[22] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad "Security Patterns: Integrating Security and Systems Engineering" Wiley, 2006.

[23] "Serenity Project - System Engineering for Security & Dependability", www.serenity-project.org, 2010

[24] M. Solinas, E. B. Fernandez, and L. Antonelli "Embedding Security Patterns into a Domain Model" in DEXA '09. 20th International Workshop on Database and Expert Systems Application, 2009, pp. 176-180.

[25] G. Spanoudakis, C. Kloukinas, and K. Androutsopoulos "Towards security monitoring patterns" in Proceedings of the 2007 ACM symposium on Applied computing Seoul, Korea: ACM, 2007.

[26] W. Stallings "Network security essentials: applications and standards", Prentice Hall, 2007.

[27] C. Steel, R. Nagappan, and R. Lai "Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management", Prentice Hall ed., 2005.

[28] H. Washizaki, E. B. Fernandez, K. Maruyama, A. Kubo, and N. Yoshioka "Improving the Classification of Security Patterns" in DEXA '09. 20th International Workshop on Database and Expert Systems Application, 2009, pp. 165-170.

[29] J. Yoder and J. Barcalow "Architectural Patterns for Enabling Application Security" Fourth Conference on Patterns Languages of Programs (PLoP'97), 1997.

[30] K. Yskout, T. Heyman, R. Scandariato, and W. Joosen " An inventory of security patterns" Katholieke Universiteit Leuven, Department of Computer Science 2006.