An Engineering Process and Modelling Framework for Development of Secure Systems

Jose Fran. Ruiz[1], Marcos Arjona[2], Antonio Maña[3], Carsten Rudolph[4] and Janne Paatero[5]
Fraunhofer SIT[1, 3], University of Malaga[2, 5] and RUAG Switzerland Ltd. [4]
GERMANY[1,4], SPAIN[2,3] and SWITZERLAND[5]

jose.ruiz.rodriguez[1],carsten.rudolph[4]{@sit.fraunhofer.de}; marcos[2],amg[3]{@lcc.uma.es};
janne.paatero[5]@ruag.com

ABSTRACT

This paper presents a novel Security Engineering Process for the creation of security-enhanced system models. The process offers a language for the definition of a domain-specific security knowledge language, the creation of security artefacts using the previous architecture and the use of these artefacts in a system model for fulfilling its security requirements and assurance. It makes security fit naturally in the systems by interleaving security into the initial architecture and system description. The process offers also solutions for the security properties by means of Security Patterns (a new type of patterns developed in the process) and Security Building Blocks. The Security Engineering Process and its Framework has being applied successfully to several and different domains (metering devices, emergency scenarios, set-top boxes, etc.) and is currently being expanded to work with cloud computing scenarios. To illustrate our process we use a mobile command post scenario where we apply the process for the creation of a security-enhanced model.

1.0 INTRODUCTION

The engineering and development of complex security-sensitive systems, is becoming increasingly difficult due to the need to address aspects like heterogeneity (of application domains, requirements, threats, regulations, solutions, etc.), dynamism and runtime adaptation needs, and the high demands for security and privacy of the users and agencies involved in scenarios where these systems work (natural disasters, military, etc.). In particular security issues require a lot of very specialized knowledge, which depends on particular application domains and evolves in a very dynamic way. Threats, security-properties, advanced security mechanisms and even seemingly simple solutions such as small security protocols are difficult to understand with all their consequences and do not show the same behaviour and properties when applied in different environments and with slightly differing configurations. Frequently, new threats, attacks and weaknesses are found. Therefore, security technology is subject to frequent changes. These characteristics make the development of security-sensitive systems hard because the security knowledge required for dealing with such a huge variety of situations cannot be sufficiently covered by the average development team. Further, most of the times, these systems must use externally developed components that have a complex set of characteristics (especially in relation to their security features). Proper specifications and assurance for security properties is often not available for components either directly integrated (e.g. software libraries) or remotely accessed via clearly defined interfaces.

Modelling is one of the most important activities in the process of engineering these systems because it establishes the foundations for the rest of the engineering activities, supports communication on development issues and can provide a basis for systematic engineering processes. Thus, languages and tools are frequently used in some parts of the development process. Unfortunately, current modelling formalisms do not support the definition of security requirements at all or do not seamlessly and naturally integrate security. Ideally, security modelling should (i) be able to coherently deal with the different elements and concepts that are related to security (properties, requirements, security mechanisms, threats, attacks, verification, assurance, etc.) and (ii) be useful as a basis for the selection and integration of appropriate security solutions during the design phase, deployment phase and for facilitating the testing phase.

This paper describes the experience of using a novel secure modelling and engineering process developed in the EU SecFutur project [1]. The main objective of SecFutur is supporting the development of dependable and secure systems composed of embedded components. To this aim, SecFutur has developed a security modelling framework and an associated engineering process that can flexibly integrate security considerations into the system design and can be incorporated into existing engineering processes. Security solutions are provided in terms of Security Patterns (SP) and Security Building Blocks (SBBs) [2, 3] that integrate hardware and software security mechanisms in order to provide complex security properties. The architecture of the SecFutur Security Modelling Framework [4] is based on UML meta-modelling capabilities and is composed of three different layers that cover different roles in the process, the security expert, the application domain expert and the actual developer/engineer. There exists a prototypical tool-support exist for all three roles called SecFutur Process Tool (SPT).

2.0 SECURITY ENGINEERING PROCESS

2.1 Introduction

The Security Engineering Process (SEP) presented here was designed and developed in the SecFutur EU Project. The main objective of the process was to provide a way to design and model secure embedded systems but, as we worked in it, we expanded it so it can be used to model any kind of system. Besides, we have worked along with NoMagic Inc. [5] (creators of MagicDraw) for developing a tool that support the process. This tool, called SecFutur Process Tool (SPT), is very useful for working with the process and can be easily integrated by companies in their current development processes and practices.

The SEP main focus is to help developers and engineers in creating security-enhanced systems by using security aspects and elements created by security domain experts. The SEP main characteristics are:

- Adequate support for users. The SEP provides artefacts and a tool (the SPT) that can be used by all the different roles of the process. Architecture designers use it for creating a Core Security Metamodel, knowledge security experts when creating security knowledge artefacts (Domain Security Metamodels), solutions experts for their Security Patterns and Security Building Blocks and system engineers for enhancing their system.
- Enable easy and sound development of secure systems. SEP can be used by system engineers to check, import and integrate security solutions in their system models even without having security knowledge. These solutions are composed of Security Patterns and SBBs that describe the implementation functionality, the hardware/software components, etc. From the point of view of the system engineer she imports a security solution (at the modelling phase) and automatically receives the implementation, its characteristics, functionality and how to use it.
- Increase of the security and the general quality of any type of system. The SEP can be used with any type of system the user needs. We started working with domains as different as metering devices, Set-top Boxes [6] and emergency/military systems. Lately, we are working with cloud systems.

2.2 Security Engineering Process Framework

The SEP Framework is composed of three different levels, each one with its own input, output and objective. They are based in its previous one, being the basis of all of them the UML Standard Metamodel v2 [7]. The artefacts are used for i) defining the language and architecture that will store the security knowledge of the systems and ii) creating domain-specific security artefacts that contain the security properties, assumptions, elements, connections, etc. of a system. These artefacts are the Core Security Metamodel (CSM) and the Domain Security Metamodel (DSM). Figure 1 show the structure of the Framework. The CSM defines the language and grammar for the definition and creation of the security knowledge of specific domains. It contains the definition of a domain and its security elements: properties, assumptions, relations, threats, attacks, verification (dynamic and static), etc.

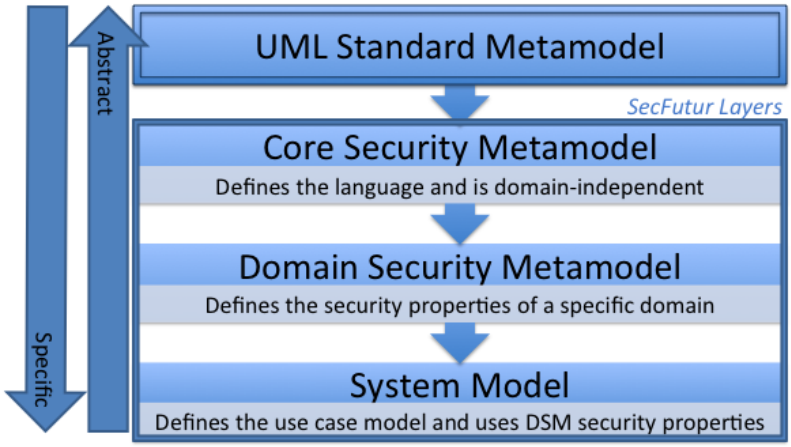

Figure 1: Security Engineering Process Architecture.

The Domain Security Metamodel (DSM) uses the CSM as basis for the creation of domain-specific security artefacts. The security properties, assumptions, threats, real world elements, actors, etc. are specific for one domain and are created by experts in the security of such domain. These experts have the necessary knowledge and expertise for creating not only the definitions and attributes of the security properties relevant in the domain, but the information about the common threats and attacks that may happen to systems in that domain, assurance mechanisms, available solutions, etc. Additionally, they are responsible for associating the appropriate security solutions to the security properties defined in the DSM. These tasks are supported by the SecFutur Process Tool, thus facilitating the adoption of SEP.

The System Model (SM) represents the system under development. SEP allows system engineers to specify requirements for the elements that constitute their models. Requirements are instances of the `SF_RM_Requirement` metaclass, and they associate security properties to these elements. This is a key part of the SEP process because it enables the tool to use the specified requirements for completing the model with the knowledge contained in the DSMs of the development repository. Next section describes how this is achieved.

2.3 Security-enhanced System Model Creation

The creation of security-enhanced system models entails a series of tasks. As we will discuss in section 2.4, the SecFutur Process Tool provides support for performing these tasks. In order to use the SPT engineers first identify the DSM (or DSMs) that fit their system in a repository. The search for the appropriate DSMs can be based on keywords, the domain name, a domain description (e.g. mesh network), the name of a security property, the element where they want to apply a security property (e.g. metering device), etc. The search functionality provides the results with a very detailed description. Once the DSM that fits better with the system under development is identified, they are imported to the SPT.

The first task is the security requirement analysis of the target system. Along with the normal requirement analysis that system engineers perform in their system (in order to identify the classes, attributes, relations, functionality, etc.) they identify the security requirements of the system, the elements that are affected by these requirements, the security goals of the system, the security assumptions, the optimal solutions, etc. SEP does not oblige engineers to take into account these requirements up-front or a posteriori. Instead, security requirements can be incorporated at any point of the development in which they appear. In order to facilitate the specification of requirements, the DSM provides a catalogue of security properties to the engineers, along with detailed information about the property. In summary, system engineers specify requirements by associating security properties to elements of the system model. This process is supported by the SPT and the

only thing system engineers have to do is to right-click in the UML element to which they want to apply the security property, and choose the property from a list. More details about this functionality are provided in Section 4. The SPT may also automatically introduce some requirements and security properties to the system, based on the domain knowledge captured in the DSM (e.g. legal or regulatory requirements for the domain).

When a requirement (or a set of them) is identified system engineers must find a way to fulfil it, but in order to provide a sound and secure solution, engineers need more information. The SPT supports this task by using the information contained in the DSM as the basis for enriching the system model with additional security information such as the relevant threats and attacks that can affect the security property, possible solutions, assurance and testing mechanisms, etc. All this specialized information helps developers select the best security solutions for their system.

When a solution is selected to fulfil a requirement, the SPT guides the integration of the solution in the target system, verifying different restrictions that are associated to the solution and expressed as OCL constraints. For instance, the tool will notify the engineer if there are elements that must be present in the system for the solution to work. An easy example of this could be if the chosen solution is an encryption algorithm, which relies on a keystore to contain the necessary keys: in this case the SPT would prompt the engineers informing them that the solution requires a keystore element in order to work and asking them whether they want to identify an existing keystore in their model or to add one. This integration checks are very useful to avoid errors due to incorrect use of a security solution.

3.0 SECURITY SOLUTIONS

3.1 Security Patterns

A Security Pattern (SP) represents a product or service designed to meet a particular security need. SPs are a way to combine SBBs or generic solutions in order to provide implementations to a security property. An example of a SP is XML Signature, which uses the RSA encryption algorithm (a SBB) in a specific way that provides Integrity and Authenticity for XML data. Thus, SPs provide specific solutions for security properties using general security artefacts such as SBBs. The SP has, like the traditional security patterns, information of how to use those SBBs by means of diagrams (use case diagrams, business model, sequence diagrams, etc.). Each SP is composed of a list of SBBs, the information of how to use it, some examples, advantages and disadvantages, related patterns, the contextual requirements, etc. One of the main components of a SP is the solution template. This UML Model defines a diagram acting as a template for all the SBBs attached to this Security Pattern. All the implementations should fit with the template and fulfil the validation rules modelled in OCL. Thanks to all these characteristics, the SPs improve the reusability, trust, integration, etc. of the SBBs. Different SPs can use the same SBBs for providing alternative solutions. It depends on how the solution is implemented and how each SBB is used. The SPs can be processed and managed by the SPT. They were designed to work with the SEP but it can be used as a security pattern with any other engineer process. They do not need a central authority to approve them and can be uploaded to a private or public repository in local or remote location.

3.2 Security Building Blocks

In contrast to a security pattern, one single security building block (SBB) does not describe a complex integrated security solution. SBBs should be seen as encapsulated components that are domain-independent and can interact with external components in order to provide a clearly defined security service. In the SEP these SBBs will be used by security solution experts to provide exact information on particular solutions in order to make this solution available for the security design process. SBBs can use other SBBs and can also interact with other components in a clearly defined way. In principle, a security building block can be just a concrete implementation of a security solution. However, in order to integrate a SBB into the engineering process, a description of the SBB is required. In the SecFutur project this description is done in terms of a

UML model. Thus, a so-called SBB-Model represents one (or several) instantiations (i.e. implementations) of the SBB. The SBB Metamodel defines the different artifacts for SBB models and their relations.

In addition to the security properties (or security service) provided by the SBB, the description of the SBB (and thus the SBB model) also needs to provide information on preconditions and constraints, as well as on postconditions on the system that need to be fulfilled by the system after the SBB was applied.

4.0 USE CASE

4.1 Use Case Description

Secure ad-hoc wireless mesh communication is the key component in the domain of spontaneous broadband communication among crisis management vehicles. This tactical communication domain plays an important role in the field of crisis management for both civilian and military supported operations, where the fixed communication infrastructure is destroyed, overloaded or not available. Examples of such cases are when infrastructure is destroyed after an earthquake or flooding, or when there is no infrastructure deployed at all, like in rural environments or in mountains.

We focus here on the use case of a mobile command post (MCP), normally achieved through wireless connections between several vehicles. These vehicles normally possess multiple communication means, but these vehicles are often not capable of interconnecting with each other in order to lead action forces, especially in the early, often chaotic initial phase of a mission. The mobile command post network is based on an ad-hoc mesh network technology. This decreases the deployment burden by reducing the entire networking configuration effort. The network is configured and operated autonomously in a self-configuring manner. Rescue forces and military acting in crisis management missions have also an increasing need to communicate while on the move, both between vehicles and between on foot action forces. Therefore, the ad-hoc nature is essential for such mobile command post networks.

The system must be able to accommodate such rapid topology changes and to find routes through the network to enable connectivity between any two vehicles. A military application is a mobile command post, which today only has wired connections between vehicles for high-bandwidth applications and VHF radios for interconnecting with vehicles on the move. Here the ad-hoc wireless mesh technology increases high-bandwidth connectivity time, as vehicles are already interconnected with high-bandwidth means as they approach and until they leave the vicinity of the command post. Also the cable-laying time is saved. Finally, having the choice between wired and wireless high-bandwidth communications enables flexibility in signals tactics. This scenario is shown in Figure 2 along with some of its characteristics.

4.2 Security-enhanced System Model Creation

Following we describe how we create the security-enhanced system model of the mobile command post scenario presented above. Due to page limit we could not describe neither the complete scenario with all its characteristics, functionality, roles, requirements, etc. nor the process for the creation of its system model. Thus, we will describe how we apply the process to the scenario and the results we obtain. The first phase is the security requirements analysis. Some of the security objectives and requirements of the system are:

- All the communications between the nodes and the command post must be secured.
- The information stored in the nodes must be protected against unauthorized intrusions.
- Since nodes spontaneously build the network and they can join and leave the network at any time, nodes have to be able to recognize trustworthiness of neighbouring nodes, to detect malicious or compromised nodes and to exclude such nodes from the network. Therefore, authentication and authorization of network nodes are key security functions to ensure only authorized access to the network and to information transmitted on the network.

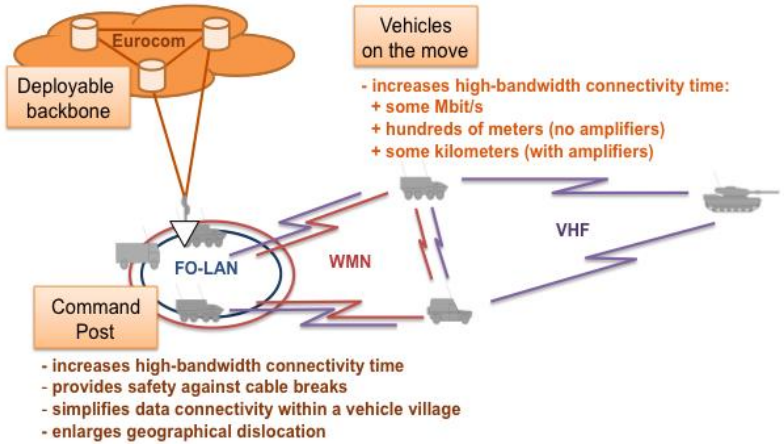

Figure 2: Excerpt of the Military Mobile Command Post Scenario

Although the list would be extensive we will focus only in one security requirement for showing the general functionality of the process. Once the security requirements are identified the system engineer uses the SPT to search in the repository of security solutions a DSM that can fit with her domain of work. She uses the keyword "Ad-hoc Mesh Network" and the SPT returns some DSMs that have this keyword in their name, description, etc. She checks the results and selects the DSM that better fits with the scenario. Next, she applies the mandatory security properties of the DSM to her system model (if any has been defined by the domain expert) and selects or creates the additional elements the system needs (for example an external node that stores a log). Following she applies security properties in order to fulfil the security requirements identified previously. We use in this example the security requirement of "Data protection": the data stored in a node must be confidential and private. She only has to right-click in the element of the model where she wants to apply the security property (the Node) and selects it from the list of security properties. Once she applies it the SPT integrates it automatically in the system model all the required elements of the property such as the assumptions, threats, attributes, etc. Figure 3 shows an excerpt of applying the Secure Data property to the Node element. She applies this process to the system till she fulfils all the security requirements. The next step is to select the solutions for each security property. This is done by checking the list of SPs of each security property and selecting the one the user finds that fits better. The SPT offers information to the system engineer by providing functionality, diagrams, list of SBBs, elements it needs, examples, etc. That way the system engineer is able to provide the security functionality and characteristics to her system without the need of a high knowledge or expertise on security.

5.0 STATE OF THE ART & RELATION WITH THE NAFV3

The design of system architectures has to face a large number of challenges due to its complexity and the heterogeneity of the issues to solve. Some of them are the concrete definition of the operational context, the deployment of the solutions, and the use of a supported modelling methodology and the fulfilment of all the requirements attached to the system functionality. One of the main concerns of any engineering process is the evaluation and integration of security in the overall process. There exist many approaches addressing the design of secure system architectures but, most of the cases, they have a low level of assistance for integrating security mechanisms in the initial stages of the software development methodology. A comprehensive engineering process to design secure architectures should (1) be able to deal with complex and evolving security requirements; (2) provide helpful support to the system engineer and (3) be adaptable to the current approaches adopted by industry.

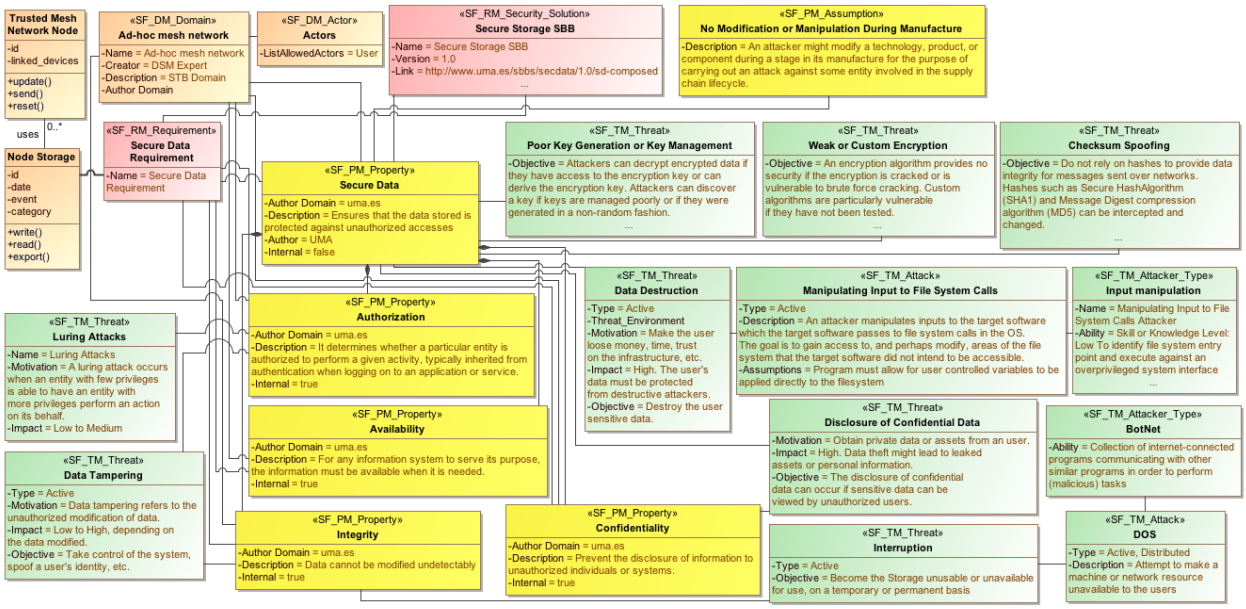

Figure 3: Secure Data property applied to the Node element of the system

Nowadays, the selection of an engineering process is not a simple task. On the one hand it should provide a way to capture the system requirements and on the other a language for representing and using this information. The Unified Modelling Language (UML) [7] has become the standard notation for the architectural design in the development process. An extension of it is the Systems Modelling Language (SysML) [8]. It is a more interdisciplinary language that focuses in the systems engineering, extending UML with many improvements such as the requirements modelling or defining a better protocol to avoid misunderstandings between systems communications. Unfortunately, the SysML simplification and specialization prevents the use of this language for latter stages of the engineering process There exist many examples of modelling frameworks such as the Eclipse Model Development Tool (MDT) [9] specially focused for developers and code generation although it does not support all the types of standard specifications. Visual Paradigm for UML [10] is a more standard user-oriented tool with many other capabilities such as software planning or requirements capturing.

There exist many security processes based on some form of abstract modelling. The exhaustive survey [11] focuses in several model-based methodologies that take the system design into account and compares different approaches like the Model Driven Security (MDS) [12], which uses transformations on security-enriched models to generate implementations. Architecture-driven methods such as [13], which provides mechanisms for incorporating security at architectural level, often using formal methods to guide the deployment process. Pattern-driven methodologies such as [14] use patterns as the core elements to introduce security properties into a system. Finally, Agent-driven methodologies based on the Agent-Oriented Software Engineering (AOSE) [15] use agents for the definition of the requirements and properties. The goals of the agents direct the functionalities and the architecture design decisions.

Domain Specific Modelling Languages (DSMLs) [16] is another common approach in security engineering that focuses in the implementation and the context requirements. The development of DSMLs is in fact a challenging task itself due to the involvement of different levels of abstraction in several domains. Its final result has usually a bounded applicability, usually restricted to a subset of components of system agents. An example of this complexity method is presented in SecEML [17]. It was created to check the validity of the security problem formulation and security design decisions but mostly oriented to business process experts and used only in general security engineering.

Another methodology for model-driven software engineering is the UML profiles. One of them is SecureMDD [18], which provides formal specification for verification and executable code for security critical applications. Real examples of this profile (e.g. [19]) show its use with smart cards and cryptographic protocols. Another well known profile is the UMLsec [20], proposed for modelling security properties of computer systems. Some applications of this profile are [21] and [22]. These profiles demonstrate their skills for representing and modelling security requirements and adjust the design process to the user needs. Unfortunately, their use is subordinated to all the elements described in the profile. New demands require an extension of the profile, causing low tolerance to changes and evolving systems. These profiles have a bounded applicability but multiple practices show their great usability under circumstances where they fit.

While working with our framework we thought it could be very interesting to check its compatibility with the NAF (NATO Architecture Framework). As the NAF is a framework for building architectures of systems we think the SecFutur SEP could fit here because of its ability to represent secure-sensitive systems. This suits perfectly in the primary use of the NAF, which is defense and security. For that reason we are working in a way to integrate our approach with the NAF methodology. The main work is to expand and create equivalences between our meta-model (the CSM) and the NAF v3 meta-model, interacting between the NAF v3 ontology and ours. As the basis of both of them is the UML Metamodel we can create equivalences and communications between the artefacts. The support will be done by means of the SPT, which, as allows UML/XMI work, can be extended with that functionality. This will allow us to define a new CSM that can be used in a NAFv3 architecture model that can benefit of the experience and work done in the SecFutur project. This experience is not only about integration of security concerns in the system model from the beginning of the process but also the experience working with heterogeneous, real-time and distributed systems. As a short example of the equivalence that can be done between the metamodels, we describe the architecture elements by means of the artefacts of our approach:

- Actor (NNEC) - Node (NMM): the SF_Actor represent actors or roles that interact with the elements of the system.
- Operational objective (NNEC) - EnterpriseGoal (NMM): we fulfil the security requirements of the system by means of security properties, which are the way to reach the goal of the security objectives.
- Capability (NNEC-NMM): the security properties are composed of attributes and requirements, which define the necessary elements and constraints of the system

Currently we are analizing the NMM and checking how to integrate our approach with it. Although we are in our initial steps seems promising and can be very beneficial for our process.

6.0 CONCLUSIONS

We presented in this paper a novel Security Engineering Process that focuses in the creation of security-enhanced systems since the beginning of the modelling phase, integrating security naturally in the system. Its main goal is a clear separation between the expertise domains. Security engineering is not only the task of the highly experienced security expert. Using this approach, also system engineers are in a position to make sound security engineering decisions. In general terms, the integration of security engineering into the regular UML-based system engineering has at least two benefits. First and foremost, as security engineering is naturally integrated from the beginning into the modelling process, it helps to avoid design decisions that are contrary to the security requirements (a frequent problem when security is only considered in the final stages of development). Second, the SEP artefacts provide a common language for the roles involved in the engineering of the system, facilitating their communication. The SEP can be applied independently from the software development process that is used, but is especially designed to fit modern iterative and agile development process, as well as model-driven approaches

To illustrate our experience we use a real-world Mobile Command Post scenario. More concretely, we focus on the establishment of a secure ad-hoc wireless mesh communication, which is a key component in the domain of spontaneous broadband communication. The experience and feedback received from the users (companies and developers) after working on this scenario has been very positive and useful for improving the process and artefacts and can be very valuable for developers in many other scenarios.

7.0 REFERENCES

[1] SecFutur Consortium. Design of secure and energy-efficient embedded systems for future internet applications (SecFutur), IST-25668. FP7.

[2] Grawrock, D. (2009). Dynamics of a Trusted Platform: A Building Block Approach. Intel Press

[3] Pearson, S. (2002). Trusted computing platforms, the next security solution. Technical report, Trusted Systems Lab, HP Laboratories.

[4] Jose Fran. Ruiz, R. H. and A. Maña, (2011). A security- focused engineering process for systems of embedded components. SD4RCES 2011.

[5] NoMagic (1995). http://www.nomagic.com/

[6] Jose Fran. Ruiz, Marcos Arjona, Antonio Ma, Antoine Monsifrot, Michel Morvan, and Andre Rein. Security Engineering and Modelling of Set-top Boxes. In RISE Workshop on Redefining and Integrating Security Engineering. IEEE, 2012.

[7] Unified Modeling Language. http://www.uml.org/

[8] Systems Modeling Language. (http://www.sysml.org/)

[9] Eclipse Model Development Tool. (http://eclipse.org/modeling/mdt/)

[10] Visual Paradigm for UML. (http://www.visual-paradigm.com/product/vpuml/)

[11] A.V. Uzunov, E.B. Fernandez & K. Falkner (2012), Engineering security into distributed systems: A survey of methodologies, *Journal of Universal Computer Science (J.UCS)* 18(20): 2920-3006

[12] Model Driven Architecture. (http://www.omg.org/mda/)

[13] Taylor, R.N., Medvidovic, N., Dashofy, E.M.: "Software Architecture: Foundations, Theory, and Practice"; Wiley, (2010).

[14] E.B.Fernandez, Designing secure architectures using security patterns. To appear in the W. S. on Software Design Patterns, 2013.

[15] Jennings 2001] Jennings, N.R.: "An agent-based approach for building complex software systems"; Commun. ACM 44(4), (2001), 35–41

[16] Gray, J., Tolvanen, J.-P., Kelly, S., Gokhale, A., Neema, S., Sprinkle, J. 2007. Domain-specific modeling. CRC Handbook on Dynamic System Modeling, Paul Fishwick, Ed. CRC Press, Boca Raton, FL.

[17] Eichler, J.; Fuchs, A.; Lincke, N., Supporting Security Engineering at Design Time with Adequate Tooling, IEEE 15th International Conference on Computational Science and Engineering (CSE), 2012.

[18] N. Moebius, K. Stenzel, and W. Reif. 2009. Generating formal specifications for security-critical applications - A model-driven approach. In Proceedings of the 2009 ICSE Workshop on Software Engineering for Secure Systems (IWSESS '09).

[19] Moebius, N., Reif, W., Stenzel, K.: Modeling Security-Critical Applications with UML in the SecureMDD Approach. International Journal On Advances in Software 1, 59–79 (2009)

[20] Jürjens, J. Towards development of secure systems using UMLSec, LNCS, 2001

[21] Ruhroth, T.; Jurjens, J., Supporting Security Assurance in the Context of Evolution: Modular Modeling and Analysis with UMLsec, (HASE), 2012

[22] Ling Shen, "Design and verification of secure channel based on UMLsec,", 2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN), 27-29 May 2011