

PAPER

Towards a UML Extension of Reusable Secure Use Cases for Mobile Grid systems

David G. ROSADO[†], *Nonmember*, Eduardo FERNÁNDEZ-MEDINA[†], *Member*, and Javier LÓPEZ^{††}, *Nonmember*

SUMMARY

The systematic processes exactly define the development cycle and help the development team follow the same development strategies and techniques, thus allowing a continuous improvement in the quality of the developed products. Likewise, it is important that the development process used integrates security aspects from the first stages at the same level as other functional and non-functional requirements. Grid systems allow us to build very complex information systems with different and remarkable features (interoperability between multiple security domains, cross-domain authentication and authorization, dynamic, heterogeneous and limited mobile devices, etc). With the development of wireless technology and mobile devices, the Grid becomes the perfect candidate for letting mobile users make complex works that add new computational capacity to the Grid. A methodology of development for secure mobile Grid systems is being defined. One of the activities of this methodology is the requirements analysis which is based in reusable use cases. In this paper, we will present a UML-extension for security use cases and Grid use case which capture the behaviour of this kind of systems. A detailed description of all these new use cases defined in the UML extension is necessary, describing the stereotypes, tagged values, constraints and graphical notation. We show an example of how to apply and use this extension for building the diagram of use cases and incorporating common security aspects for this kind of systems. Also, we will see how the diagrams built can be reused in the construction of others diagrams saving time and effort in this task.

key words: Security, Security Use Cases, secure development, secure Mobile Grid, Reusability.

1. Introduction

The growing need for constructing secure systems, mainly due to the new vulnerabilities in using the Internet and that of the applications distributed in heterogeneous environments, encourages the scientific community to demand a clear integration of security into the development processes [3, 5, 14, 21, 26, 27]. The main reason is that, traditionally, security aspects are only considered at the implementation stages which doesn't allow security solutions to be perfectly coupled with the design and the rest of requirements of the system [1, 27]. Model Driven Security [2] is a clear example of integration of software engineering and security engineering and, in some way, it offers ideas that we use in our workline.

Systems which are based on Grid Computing are a kind of systems that have clear differentiating features

[11, 12, 22] where security is a very important aspect. Grid environments have special features that make them different from other systems and that we should consider throughout the whole development lifecycle. Generic development processes are used to develop systems without taking into consideration either the subjacent technological environment or the special features and particularities of these specific systems.

Mobile Grid, in relevance to both Grid and Mobile Computing, is a Grid with the additional feature of supporting mobile users and resources in a seamless, transparent, secure and efficient way [13, 18, 25]. Grids and mobile Grids can be the ideal solution for many large scale applications being of dynamic nature and requiring transparency for users.

Security has been a central issue in grid computing from the outset, and has been regarded as the most significant challenge for grid computing [7, 15]. Security over the mobile platform is more critical due to the open nature of wireless networks. In addition, security is more difficult to implement in a mobile platform due to the limitations of resources in these devices [4]. Therefore, a Grid infrastructure that supports the participation of mobile nodes will play a significant role in the development of Grid computing.

The majority of existing Grid applications have been built without a systematic development process and are based on ad-hoc developments [9, 22]. The lack of adequate development methods for this kind of systems has encouraged us to build a methodology to develop them (see Fig. 1), offering a detailed guide to analyze, design and implement them. This methodology is strongly oriented to reuse and takes special care of security and the use of mobile devices in Computational Grids. Reuse is mainly concentrated on i) the analysis stage in which we start from a set of predefined use cases and we integrate them into the use cases identified for a new application and ii) the design stage in which we start from an architecture that incorporates the previously identified reusable security services and then it is specialized for each one of the new applications that are created. The set of use cases as well as the security architecture are adapted to the features of computational grids and specially oriented to support security requirements [16, 30, 44, 45] and services and to the use of mobile devices as Grid nodes.

In the previous publications related to this issue, in

Manuscript received January xx, 20xx.

Manuscript revised March xx, 20xx.

[†]The author is with NTT, Musashino-shi, 180-8585 Japan.

^{††}The author is with IEICE, Minato-ku, Tokyo, 105-0011 Japan.

[36] we explain the reusability of the use case diagrams following the UML extension, we define the repository of elements and we show complete use case diagrams that can be built in any development. This UML extension has been applied to a real case describing all possible values that can take the elements of the UML profile for a specific application [34]. In this paper we show a refinement and the final version of the UML profile, providing an in depth description of all the properties, characteristics and constraints of the elements defined in the UML extension. We also provide a detailed description of all the tagged values identified for the elements of the profile, and we define a set of possible values that can take these tagged values. Finally, we present an example of a use case diagram and how it can be reused through the repository by adding new elements of the profile to build a new diagram.

The rest of paper is organized as follows: In section 2, we will present the related work. In section 3, we will summarize briefly the proposed methodology. In section 4, stereotypes and associations of Grid use cases will be define and we will describe formally these stereotypes. In section 5, we will apply the new stereotypes to build a diagram of use cases for a real case. We will finish by putting forward our conclusions as well as some research lines for our future work in section 6.

2. Related Work

The idea of developing software through systematic development processes to improve software quality is not new [10, 17, 23, 40]. Nevertheless, there are still many information systems such as the Grid Computing ones, that are not developed through methodologies adapted to their most differentiating features [22]. In fact, we have not found other proposals for the systematic development of Grid Computing systems, in spite of this is demanded by the scientific community.

On the other hand, there are some proposals which try to integrate security into the software development process, even from the first stages, but however, none of them are defined for Grid Computing based systems. For instance, authors in [42] present a methodology for the integration of the security on software systems. This methodology is based in the Unified Process [23] and it is called Secure Unified Process (SUP). SUP establishes the pre-requirements to incorporate the fundamental principles of security. Also, it defines an optimized design process of security within the life cycle of software development. The problem is that it only offers a solution to a very high level without offering “practical mechanisms” (e.g. Grid-specific security artifacts or a security architecture of reference) that permits to implement his approach in a short space of time and with minimal effort. Other approach [19-21] concentrates on providing a formal semantics for UML to integrate security considerations into the software design process.

The approach presents UMLsec which is an extension of UML and allows expressing security-relevant information. In [33], authors show a methodical approach for the development of security-critical systems and the modelling of security aspects in the application core with UMLsec. This approach defines a specification of use cases with textual description analyzing threats and the vulnerability of input and output data, applying UMLsec to the rest of diagrams UML of the application. Our approach extends UML (textual and graphical description) to be applied in the specification of diagrams of use cases allowing to express security aspects from the beginning of the development, where the capture of requirements (functional and non-functional) is essential. UMLSec and our proposal are compatible, while models from UMLSec can be used for specifying general security aspects of systems, and our approach could be used for specifying security features for Grid environments.

On the other hand, the current grid architecture and algorithms do not take into account the mobile computing environment since mobile devices have not been seriously considered as valid computing resources or interfaces in grid communities. It has been just recently given attention to integrate these two emerging techniques of mobile and grid computing, for example, in [6, 8, 18, 24, 32], although they do not elaborate on how the mobile devices may be incorporated in the current grid architecture. All these proposals attempt to incorporate mobile devices to an existing Grid infrastructure through of tools and platforms that allow it. Our methodology considers on the one hand, the incorporation of mobile devices as a resource more and not as an external element of the system, and on other hand, this incorporation is performed from the initial activities of the methodology considering security aspects and limitations of these devices from the beginning of the development.

3. Methodology overview

The structure of the methodology follows the classical cycle, where we can find a planning phase, a development phase including analysis, design and construction and finally a maintenance phase. However, it is specially designed for this kind of systems and considers their particular features. Further detail on activities and tasks of our methodology can be found in [35, 37]. In Fig. 1 we can see the structure of the methodology using SPEM (Software & Systems Process Engineering Metamodel) version 2.0 [29].

What makes this methodology different from the rest can be found in the development of its stages in which we define tasks and activities specific for mobile Grid systems where the reuse of elements (such as use cases, security use cases, reference security architecture,

etc., available on the repository) is a key aspect in the development and where the Grid technological environment and mobile computing are taken into account and present in each task and activity of the methodology.

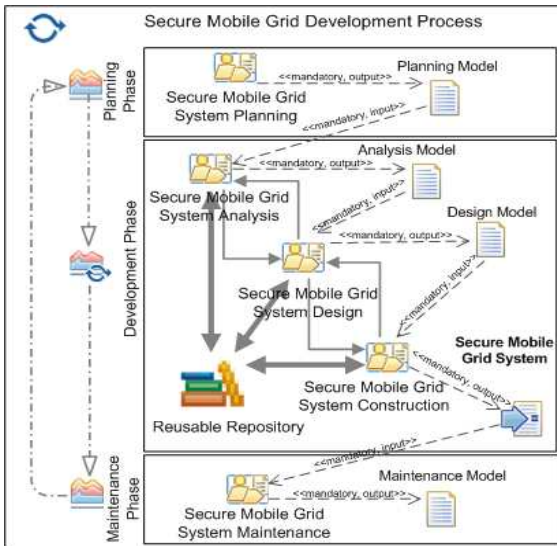


Fig. 1. Structure of our methodology

The planning phase has only one activity: “Secure Mobile Grid System Planning”, where it should do an initial capture of requirements and necessities permitting to elaborate a development plan. In this capture of requirements and necessities, we should identify the basic functionality of the system, involved domains and organizations, risks of the system, types of resources and users (mobile devices, PDAs, etc.), the main security aspects of the grid and technology considerations.

The development phase is composed of three activities: analysis, design and construction.

The “Secure Mobile Grid System Analysis” activity is centred on identifying and analyzing the requirements and security requirements of mobile Grid systems. The analysis activity is based on use cases in which we define the behaviour, actions and interactions with those people implied in the system (actors) to obtain a first approach to the needs and requirements (functional and non-functional) of the system to be constructed. We have used the UML profile which was specifically defined for this purpose as a basis, and we use a reusable use case model in which the use case diagrams and security use cases that have a common behaviour for this kind of systems, and which have been built in previous developments, are defined. These use cases and security use cases are used to identify, refine and specify the functional and non-functional requirements with the help of a UML profile, and are finally integrated with the other typical analysis models. The aim of this activity is to reduce

the time and effort spent on the construction of use cases diagrams for this kind of systems by reusing diagrams which have already been built and which show a similar behaviour to that which we wish to define. These reusable use cases are stored in a repository which is available for the development process that we are elaborating and in which we store the use case diagrams that are candidates for reuse. This repository will be managed by a tool which facilitates the design and construction of the use case diagrams by following the UML profile and using the repository of reusable elements in an easy and intuitive manner.

In the “Secure Mobile Grid System Design” activity, we should select the structural elements from which the system is composed and the behaviour and interfaces between them. A full design of classes, interfaces and state diagrams is necessary together with collaboration, components and deployment diagrams. All these models give an architectural vision of the system contributing with security aspects of the application that should be incorporated to the reference security architecture, previously build, that offers the necessary security services that fulfil and cover the security requirements identified in the analysis model. This architecture will be a service-oriented architecture where we define a collection of security services supporting the security requirements of mobile Grid environments. This security architecture will be integrated in the software architecture obtaining a secure software architecture specified for Mobile Grid systems.

In the “Secure Mobile Grid System Construction” activity, the implementation model (components and deployment diagrams) are refined and a Grid technological platform should be selected for building the design model obtained in the last activity, and to implement and test the secure software architecture defining security services together with security mechanisms and protocols for our security architecture. It is possible that we have to expand the technological environment for treating with mobile Grid systems.

The maintenance phase has only one activity: “Secure Mobile Grid System Maintenance” and it is a typical activity of maintenance of any development process, where a plan of maintenance of the system for its later modification is defined according to the new necessities of the client.

The general repository of the process contains a set of reusable elements, which originate from executions of the process for other Mobile Grid applications where common aspects are extracted and stored, or that were initially specifically built for this kind of systems and that are available to be used by the different activities and tasks of the process. With the execution of the process to develop new mobile Grid applications we obtain new use case, security use case and misuse case

diagrams which define some specific behaviour or function of the system that we consider to be common to many other applications and we therefore store these diagrams in the repository for their subsequent use.

We can thus have diagrams in the repository in which typical use cases, Grid use cases, security use cases and misuse cases describing some scenario of the mobile Grid environments are defined, for example, to ensure the confidentiality of the requests that are sent from mobile devices protecting the system from the alteration of the message by unauthorized users, who are shown in Fig. 4. This type of scenario is common in Grid systems (with some possible variation) and, therefore, having a diagram that can be reused is an advantage when we construct use case diagrams for the application which we are developing.

Finally, we have developed a prototype tool to give automatic support to the process and to help analysts build use cases diagrams in a simple, automatic and intuitive manner by following the UML extension. This tool is focused on the construction and definition of secure Grid use cases diagrams, and on the management of the repository that stores reusable artefacts which can be reused in the construction of diagrams. This tool allows us to define use cases, Grid use cases, security use cases, Grid security use cases, mobile use cases and misuse cases in a graphical manner, together with all the information related to them.

4. Extension for specification of Grid use cases for secure mobile Grid systems

The analysis activity of the methodology is based on use cases where we define the behaviour, actions and interactions with those implied by the system (actors) obtaining a first approach to the needs and requirements (functional and non-functional) of the system to construct. This activity is supported by the reuse of Grid use cases and security use cases stored in the repository where we obtain correct use cases that define a common behaviour of the Grid system that are very frequently used in the majority of use case diagrams that are built for different Grid systems. We have defined new stereotypes for constructing use case diagrams for secure mobile Grid environments.

The developers or analysts who take part in the analysis of the system begin by planning the system capturing and defining the initial requirements and needs that can be informally defined in plain text format or through templates with information to be filled, considering the information defined in the UML extension such as the tagged values. Next, in the analysis activity, all the information which was initially captured, together with other information obtained from the system, the environment, the functionality, the features, and so on, have to be translated to an analysis model. This analysis model is based on use case models following the

notation and modelling of the new UML profile defined for this purpose in which we can formally model all the information of the system in order to develop use case diagrams. In this activity, which has been defined in depth in [38], we build the use cases diagrams by following the steps and tasks described in the process. It is in the analysis activity tasks of “*Building secure mobile Grid UC diagram*” and “*Identifying secure mobile Grid UC*” that the UML profile is used to build use cases diagrams for mobile Grid systems.

To define reusable use case diagrams, which are specific for mobile Grid systems, we need to extend the UML 2.0 metamodel and define stereotypes. A stereotype is an extension of the UML vocabulary that allows us to create new building blocks derived from the existing ones but specific for a concrete domain, in our case, the Grid computing domain. In this section we present the extension GridUCSec-Profile through which it is possible to represent specific mobile Grid features and security aspects for diagrams of use cases obtaining as result diagrams of use cases for secure mobile Grid environments. This extension has been built as UML profile which is a mechanism of extensibility that allows to adapt the metaclasses of a model so the incorporation of new elements in a domain is possible.

This section is organized as follows: In subsection 4.1, we will introduce our extension for secure Grid use cases. Subsection 4.2 will show the stereotypes considered, and, finally in the subsection 4.3, we will define the tagged values identified and the types for these tagged values.

4.1 GridUCSec-Profile Extension

For the representation of the Grid use cases and security use cases, a set of stereotypes have been defined, which have been grouped in packages, *GridUCSec* and *TypesGridUCSec* that are part of GridUCSec-Profile.

The *GridUCSec* package (see Fig. 2) is composed of Grid use cases, security use cases, misuse cases, associations of permission, protection, threaten and mitigation, together with the involved actors. This package has 11 stereotypes: 4 specialize to UseCase (from UseCases), 2 specialize to Actor (from UseCases), and 5 specialize to DirectedRelationship (from Kernel) and NamedElement (from Kernel and Dependencies). The stereotypes that compose this package will be defined in the next subsections.

The *TypesGridUCSec* package (see Fig. 2) defines the types of data for the tagged values of the stereotypes of GridUCSec-Profile, as are level of protection and of risk, types of permission, of requirement, of asset, of attack, etc. This package is composed of nine stereotypes which specialize the Enumeration class (from Kernel).

domain) to make their work available to a trusted network of peers the same instant it is produced, either from desktop or mobile devices. We want to build a system that will cater for the reporter who is on the move with lightweight equipment and wishes to capture and transmit news content. This user needs to safely and quickly upload the media to a secure server to make it easier for others to access, and to avoid situations where his device’s battery dies or another malfunction destroys or makes his media unavailable.

Using the GridUCSec-Profile extension, we will build a diagram of use cases for this application helping us of the reusable use cases defined in the repository of proposed methodology. For all possible use cases defined for this application, we are only going to consider one of them (due to space constraints), which is *Query* of information.

We consider that the mobile user has access to the grid system (authorization) and then the user can realize queries to the Grid for obtaining information. These queries are messages which are sent to the system following some communication protocol; therefore, we must protect these queries of attacks offering confidentiality of messages. In the repository we can find reusable security use cases of confidentiality (ensure confidentiality) which will be used in the diagram of the application.

We should define the relationships with the reusable security use cases, with reusable grid use cases, and with the misuse cases for building the diagram of use cases of the application. In the Fig. 4 we can see this diagram with reusable use cases for this case study with one use case, query. The security use cases that are directly related will be included automatically, because they are related with reusable use cases and they are necessary for the realization of the reusable use cases. The analyst does not need define these use cases.

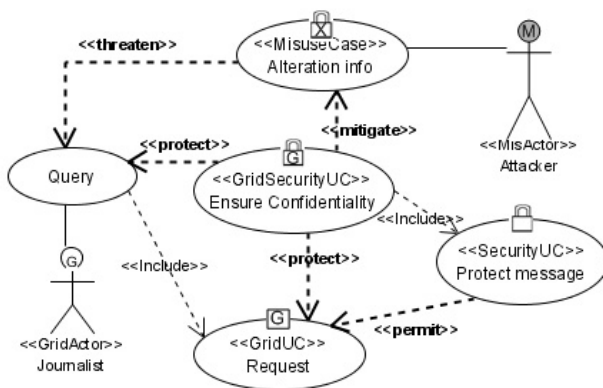


Fig. 4. Using reusable use cases of the repository for building the overall diagram of the application.

The diagram in Fig. 4 shows how the “*GridUC* Request” use case is protected, through the «protect» relationship, by the “*GridSecurityUC* Ensure

Confidentiality” security use case which mitigates the “*MisuseCase* Alteration info” misuse case that threatens the “*Query*” use case. It also establishes a «permit» relationship from the “*SecurityUC* Protect message” security use case, meaning that once the message is protected, the request can be carried out.

So, for example, for the “*GridSecurityUC* Ensure Integrity” use case, we assign the value of “Integrity” to the “*SecurityRequirement*” tagged value, indicating the incorporation of this security requirement into the application; The values of “*Message, Data*” are assigned to the “*InvolvedAsset*” tagged value signifying that they are the important asset to be protected; A value of “*High*” is assigned to the “*SecurityDegree*” tagged value, indicating a high degree of security of the message and data in the system; Finally, we assign a value of “*VLow*” (very low) to the “*SecurityDependence*” tagged value which indicates that this use case has a very low risk level and does not, therefore, need to be protected by other security use cases. Moreover, it is also necessary to assign (in the same way) the different values for the tagged values defined in the relationships that this use case has with the other use cases in the diagram, such as two “*protect*” relationships, two “*mitigate*” relationships and one “*threaten*” relationship.

Once the diagram that has been built has been defined and checked (see Fig. 4), it will be observed that this diagram shows a common behaviour with that of many Grid systems, in order to represent the confidentiality of messages which flow in the system. This diagram will be stored in the repository with its relationships, actors and detailed information to be used in a new development or iteration of the process.

In the case of a new iteration to which we wish to add new requirements, we can extract the use case diagram previously built from the repository and modify, add, and build a new use case diagram from the reused diagram, adding new relationships and information. So, for example, if in this new iteration we wish to incorporate the integrity requirement in the messages and additionally protect the system from a new threat such as disclosure of information, we can refine the previously built diagram and store this in the repository, and incorporate these new use cases, the new relationships and the all information according to these new elements by following the GridUCSec-profile extension. The resulting diagram is shown in Fig. 5 in which we can see the new elements that have been added (in bold type for the relationships and with a grey background for the use cases).

The new information for this diagram (that we have omitted owing to the strict paper size restrictions) is added in the same way aforementioned, but with different values, as the information that was added to the diagram in Fig. 4. The information to be added concerns “*Ensure Integrity*” and “*Disclosure info*” use cases and the “*protect*”, “*mitigate*” and “*threaten*” relationships

which are either in or out of the new use cases that have been added to the diagram. Finally, we can see that this new diagram also represents a common behaviour (confidentiality and integrity) which may appear in many Grid systems and we can therefore store it in the repository for its future reuse.

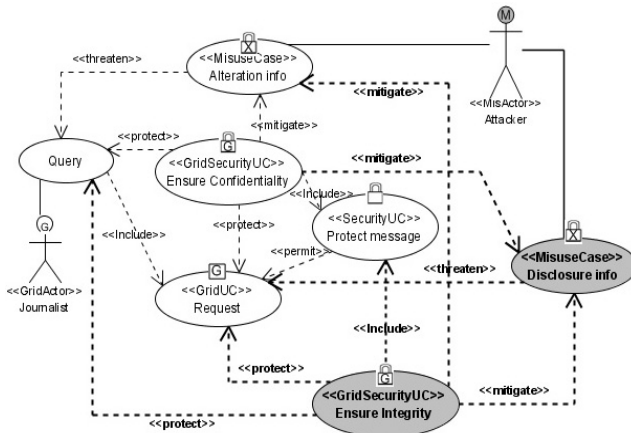


Fig. 5. Extended Diagram with two new use cases

6. Conclusions and future work

The complexity of current applications forces us to think and follow an action plan to control the whole software lifecycle as well as to ensure that decisions are made in a controlled way. A systematic process is essential to build quality software, offering methods, techniques and tools that facilitate the work of all the team involved in software development. There are numerous referring studies to incorporate security into the whole life cycle of software in order to obtain an end product that fulfils the required security requirements. In the case of the life cycle of a mobile Grid system, the same situation occurs; it is necessary to incorporate security from the first stages of development, by defining a methodology that, in addition to developing a mobile Grid system by considering the peculiarities and necessities of this type of systems, incorporates all aspects of Grid security and mobile devices into the life cycle and consequently obtains a secure end product. This process must always be flexible, scalable and dynamic, so that it adapts to the necessities, always changing, of the Grid systems.

An important stage of the methodology is the requirements analysis stage that has been managed by reusable use cases and that facilitates the specification of both system and security requirements of our application. For the definition of use cases for this kind of systems, it is necessary to define UML-extension that captures the behaviour and remarkable features of the mobile Grid environments. The development of mobile Grid system is a complex and tedious task. For that reason, firstly, with a methodology, secondly with reuse, and thirdly with a UML-extension for use cases, we can facilitate the capture of requirements and reduce time and effort in the

development of this kind of systems.

As future work, we aim to complete the details of this methodology (activities, tasks, etc.) through the research-action method. Security requirements engineering techniques (UMLSec, etc.) will be integrated into our process. We will define the traceability of artefacts from use cases in the analysis activity, identifying design elements in the design activity in order to arrive at any implementation platform (i.e. Globus) in the construction activity.

Acknowledgments

This research is part of the following projects: QUASIMODO (PAC08-0157-0668), SISTEMAS (PII2I09-0150-3135) and SEGMENT (HITO-09-138) financed by the "Viceconsejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha" (Spain) and FEDER, and MEDUSAS (IDI-20090557), BUSINESS (PET2008-0136) and PEGASO/MAGO (TIN2009-13718-C02-01) financed by the "Ministerio de Ciencia e Innovación (CDTI)" (Spain). Special acknowledgment to GREDIA (FP6-IST-034363) funded by European Commission.

References




- [1] Artelsmaier, C. and R. Wagner. Towards a Security Engineering Process. in The 7th World Multiconference on Systemics, Cybernetics and Informatics. 2003. Orlando, Florida, USA.
- [2] Basin, D., J. Doser, and T. Lodderstedt. Model driven security for process-oriented systems. in ACM Symposium on Access Control Models and Technologies. 2003. Como, Italy: ACM Press.
- [3] Bass, L., F. Bachmann, R.J. Ellison, A.P. Moore, and M. Klein. Security and survivability reasoning frameworks and architectural design tactics. SEI, 2004.
- [4] Bradford, P.G., B.M. Grizzell, G.T. Jay, and J.T. Jenkins, Cap. 4. Pragmatic Security for Constrained Wireless Networks, in Security in Distributed, Grid, Mobile, and Pervasive Computing, A. Publications, Editor. 2007: The University of Alabama, Tuscaloosa, USA. p. 440.
- [5] Breu, R., K. Burger, M. Hafner, J. Jürjens, G. Popp, V. Lotz, and G.Wimmel. Key issues of a formally based process model for security engineering. in International Conference on Software and Systems Engineering and their Applications. 2003.
- [6] Clarke, B.a.M.H. Beyond the 'Device as Portal': Meeting the Requirements of Wireless and Mobile Devices in the Legion Grid Computing System, ". in Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing at the International Parallel and Distributed Processing Symposium. 2002: IEEE Press.
- [7] Chakrabarti, A., A. Damodaran, and S. Sengupta, Grid computing security: A taxonomy. IEEE Security & Privacy, 2008. 6(1): p. 44-51.
- [8] Chu, D. and M. Humphrey. Mobile osgi.net: Grid computing on mobile devices. in 5th IEEE/ACM International Workshop on Grid Computing -Grid2004 (at Supercomputing 2004). 2004.
- [9] Dail, H., O. Sievert, F. Berman, H. Casanova, A. YarKhan, S. Vadhiyar, J. Dongarra, C. Liu, L. Yang, D. Angulo, and I. Foster, Scheduling In The Grid Application Development Software Project, in Grid resource management:state of the art and future

- trends. 2004. p. 73-98.
- [10] Flechais, I., M.A. Sasse, and S.M.V. Hailes. Bringing Security Home: A process for developing secure and usable systems. in *Nwe Security Paradigms Workshop (NSPW'03)*. 2003. Ascona, Switzerland.
- [11] Foster, I. and C. Kesselman, *Globus: A Toolkit-Based Grid Architecture*, in *The Grid: Blueprint for a New Computing Infrastructure*. 1999, Morgan Kaufmann. p. 259-278.
- [12] Foster, I., C. Kesselman, J.M. Nick, and S. Tuecke, Grid services for distributed system integration. *Computer*, 2002. 35(6): p. 37-46.
- [13] Guan, T., E. Zaluska, and D.D. Roue. A Grid Service Infrastructure for Mobile Devices. in *First International Conference on Semantics, Knowledge, and Grid (SKG 2005)*. 2005. Beijing, China.
- [14] Haley, C.B., J.D. Moffet, R. Laney, and B. Nuseibeh. A framework for security requirements engineering. in *Software Engineering for Secure Systems Workshop*. 2006. Shanghai, China.
- [15] Humphrey, M., M.R. Thompson, and K.R. Jackson, *Security for Grids*. Lawrence Berkeley National Laboratory. Paper LBNL-54853, 2005.
- [16] ITU, *ITU_T Recommendation X.1121. Framework of security technologies for mobile end-to-end data communications*. 2004.
- [17] Jacobson, I., G. Booch, and J. Rumbaugh, *The Unified Software Development Process*. 1999: Addison-Wesley Professional. 512.
- [18] Jameel, H., U. Kalim, A. Sajjad, S. Lee, and T. Jeon. Mobile-To-Grid Middleware: Bridging the gap between mobile and Grid environments. in *European Grid Conference EGC 2005*. 2005. Amsterdam, The Netherlands: Springer.
- [19] Jurjens, J. Towards Development of Secure Systems Using UMLsec. in *Fundamental Approaches to Software Engineering (FASE/ETAPS)*. 2001.
- [20] Jurjens, J. UMLsec: Extending UML for Secure Systems Development. in *5th International Conference on the Unified Modeling Language (UML)*. 2002. Dresden, Germany.
- [21] Jürjens, J., *Secure Systems Development with UML*. 2004: Springer-Verlag.
- [22] Kolonay, R. and M. Sobolewski. Grid Interactive Service-oriented Programming Environment. in *Concurrent Engineering: The Worldwide Engineering Grid*. 2004. Tsinghua, China: Press and Springer Verlag.
- [23] Kruchten, P., *The Rational Unified Process: An Introduction*. 2nd ed. 2000: Addison-Wesley. 320.
- [24] Kwok-Yan, L., Z. Xi-Bin, C. Siu-Leung, M. Gu, and S. Ji-Guang, Enhancing Grid Security Infrastructure to Support Mobile Computing Nodes. *Lecture Notes in Computer Science*, 2004. 2908/2003: p. 42-54.
- [25] Litke, A., D. Skoutas, and T. Varvarigou. Mobile Grid Computing: Changes and Challenges of Resource Management in a Mobile Grid Environment. in *5th International Conference on Practical Aspects of Knowledge Management (PAKM 2004)*. 2004.
- [26] Lodderstedt, T., D. Basin, and J.r. Doser. *SecureUML: A UML-Based Modeling Language for Model-Driven Security*. 2002. Dresden, Germany: Springer.
- [27] Mouratidis, H. and P. Giorgini, *Integrating Security and Software Engineering: Advances and Future Vision*. 2006: IGI Global.
- [28] OMG, *OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2*. 2007.
- [29] OMG, *Software & Systems Process Engineering Meta-Model Specification (SPEM) 2.0*. 2008.
- [30] Open Grid Forum, *The Open Grid Services Architecture, Version 1.5*. 2006.
- [31] P. Resnick, P.Z., R. Friedman, K. Kuwabara, *Reputation Systems*. *Communications of the ACM*, 2000. 43(12): p. 45-48.
- [32] Phan, T., L. Huang, and C. Dulan. Challenge: Integrating Mobile Wireless Devices Into the Computational Grid. in *8th annual international conference on Mobile computing and networking (MobiCom'02)*. 2002. Atlanta, Georgia, USA: ACM Press.
- [33] Popp, G., J. Jürjens, G. Wimmel, and R. Breu. Security-Critical System Development with Extended Use Cases. in *Tenth Asia-Pacific Software Engineering Conference (APSEC'03)*. 2003: IEEE.
- [34] Rosado, D.G., E. Fernández-Medina, and J. López. Applying a UML Extension to build Use Cases diagrams in a secure mobile Grid application. in *5th International Workshop on Foundations and Practices of UML, in conjunction with the 28th International Conference on Conceptual Modelling, ER 2009*. 2009. Gramado, Brasil: LNCS 5833.
- [35] Rosado, D.G., E. Fernández-Medina, and J. López, Obtaining security requirements for a mobile grid system. *International Journal of Grid and High Performance Computing*, 2009. 1(3): p. 1-17.
- [36] Rosado, D.G., E. Fernández-Medina, and J. López. Reusable Security Use Cases for Mobile Grid environments. in *Workshop on Software Engineering for Secure Systems, in conjunction with the 31st International Conference on Software Engineering*. 2009. Vancouver, Canada.
- [37] Rosado, D.G., E. Fernández-Medina, J. López, and M. Piattini. Engineering Process Based On Grid Use Cases For Mobile Grid Systems. in *The Third International Conference on Software and Data Technologies- ICSOFT 2008*. 2008. Porto, Portugal.
- [38] Rosado, D.G., E. Fernández-Medina, J. López, and M. Piattini, Analysis of secure mobile grid systems: A systematic approach. *Information and Software Technology*, 2010. 52: p. 517-536.
- [39] Røstad, L. An extended misuse case notation: Including vulnerabilities and the insider threat. in *XII Working Conference on Requirements Engineering: Foundation for Software Quality*. 2006. Luxembourg.
- [40] Schmidt, D.C., *Model-Driven engineering*. *IEEE Computer*, 2006. 39(2).
- [41] Sindre, G. and A.L. Opdahl. Capturing Security Requirements by Misuse Cases. in *14th Norwegian Informatics Conference (NIK'2001)*. 2001. Tromsø, Norway.
- [42] Steel, C., R. Nagappan, and R. Lai, Chapter 8. The Alchemy of Security Design Methodology, Patterns, and Reality Checks, in *Core Security Patterns: Best Practices and Strategies for J2EE™, Web Services, and Identity Management*. 2005, Prentice Hall PTR/Sun Micros. p. 1088.
- [43] *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*. W3C Recommendation. 2002 16 April; Available from: <http://www.w3.org/TR/P3P/>.
- [44] *Trusted Computing Group Administration, Securing Mobile Devices on Converged Networks*. 2006.
- [45] Vivas, J.L., J. López, and J.A. Montenegro, Chapter 12. Grid Security Architecture: Requirements, fundamentals, standards, and models, in *security in distributed, grid, mobile, and pervasive computing*, A. Publications, Editor. 2007: Tuscaloosa, USA. p. 440.

Appendix

Table 1. Detailed description of Stereotypes for the GridUCSec package

Stereotype	GridUC	SecurityUC
Description	Specify requirements of the Grid system and represent the common behaviour and relationships for this kind of systems. It specializes to UseCase within Grid context defining the behaviour and functions for the Grid system.	Specify security requirements of the system, describing security tasks that the users shall be able to perform by means of the system.
Generalization	Classifier::BehavioredClassifier::UseCase	
Associations	-isPermitting:Permit[0..*]. It references to Permit relationship that is permitting to this use case. -isProtecting:Protect[0..*]. It references to Protect relationship that is protecting to this use case. -isThreatening:Threaten[0..*]. It references to Threaten relationship that is threatening to this use case.	-mitigate:Mitigate[0..*]. It references the Mitigate relationships owned by this security use case. -permit:Permit[0..*]. It references the Permit relationships owned by this security use case. -protect:Protect[0..*]. It references the Protect relationships owned by this security use case.
Notation		
Tagged values	GridRequirement, ProtectionLevel, SecurityDependence, InvolvedAsset	SecurityRequirement, InvolvedAsset, SecurityDegree
Constraints	- It defines some type of value for some tagged values context GridUC inv: self.GridRequirement->size()=1 inv: self.InvolvedAsset->size()=1 inv: self.ProtectionLevel->size()=1 inv: self.SecurityDependence->size()=1 - It only associates with GridActor context GridUC inv: self.MisActor->size()=0 inv: self.GridActor->size()>=0	- It does not inherit the relationship with Threaten. context SecurityUC inv: self.Threaten->size()=0 - It defines some type of value for all the tagged values context SecurityUC inv: self.SecurityRequirement->size()=1 inv: self.InvolvedAsset->size()=1 inv: self.SecurityDegree->size()=1 - It must have almost one association context SecurityUC inv: (self.mitigation->size() + self.permit->size() + self.protect->size())>=1 - It only associates with GridActor context SecurityUC inv: self.MisActor->size()=0 inv: self.GridActor->size()>=0
Stereotype	GridSecurityUC	MisuseCase
Description	They represent specific security features of Grid systems. Add specific special security features which are covered by this stereotype, and specialize to common security use cases of other applications, providing unique features for Grid environments.	A sequence of actions, including variants, that a system or other entity can perform, interacting with misusers of the entity and causing harm to some stakeholder if the sequence is allowed to complete [39, 41].
Generalization	-Classifier::BehavioredClassifier::UseCase::SecurityUC -Classifier::BehavioredClassifier UseCase::GridUC	- Classifier::BehavioredClassifier::UseCase
Associations	It inherits associations of SecurityUC. It only inherits of GridUC the association isPermitting.	- threaten:Threaten [1..*]. It references the Threaten relationships owned by this misuse. - isMitigating:Mitigate [0..*]. It references to Mitigate relationship that is mitigating this misuse case.
Notation		
Tagged values	InvolvedAsset, SecurityRequirement, SecurityDegree, SecurityDependence	InvolvedAsset, ImpactLevel, RiskLevel, ThreatLikelihood, KindAttack
Constraints	- It inherits the restrictions of GridUC and SecurityUC - It defines some type of value for all the tagged values context GridSecurityUC inv: self.SecurityRequirement->size()=1 inv: self.InvolvedAsset->size()=1 inv: self.ProtectionLevel->size()=1 inv: self.SecurityDependence->size()=1. - The threaten and protect relationships are not inherited of GridUC. context GridSecurityUC inv: self.Threaten->size()=0 inv: self.Protect->size()=0	- Some relationships are not inherited of UseCase. context MisuseCase inv: self.Permitt->size()=0 inv: self.Protect->size()=0 - It only associates with MisActor context MisuseCase inv: self.MisActor->size()>=0 inv: self.GridActor->size()=0 - It defines some type of value for all the tagged values context MisuseCase inv: self.KindAttack->size()=1

		inv: self.InvolvedAsset->size()=1 inv: self.ImpactLevel->size()=1 inv: self.RiskLevel->size()=1 inv: self.ThreatLikelihood->size()=1
Stereotype	MobileUC	
Description	It represents mobile features of the mobile devices within Grid systems. It defines the mobile behaviour of the system and specializes to UseCase within the Grid context and mobile computing defining the behaviour and functions for the Mobile Grid system.	
Generalization	- Classifier::BehavioredClassifier::UseCase	
Associations	- isPermitting: Permit [0..*]. It refers to the Permit relationship that is permitting this use case. - isProtecting: Protect [0..*]. It refers to the Protect relationship that is protecting this use case. - isThreatening: Threaten [0..*]. It refers to the Threaten relationship that is threatening this use case.	
Notation		
Tagged Values	MobileRequirement, ProtectionLevel, SecurityDependence, InvolvedAsset, NetworkProtocol, DomainName	
Constraints	- It defines some types of value for some tagged values context MobileUC inv: self.MobileRequirement->size()=1 inv: self.ProtectionLevel->size()=1 inv: self.InvolvedAsset->size()=1 inv: self.SecurityDependence->size()=1 inv: self.NetworkProtocol->size()=1 inv: NameDomain->size()=1	- It only associates with GridActor context MobileUC inv: self.MisActor->size()=0 inv: self.GridActor->size()>=0
Stereotype	Protect	Permit
Description	This relationship specifies that the behaviour of a use case may be protected by the behaviour of a security UC.	This relationship specifies that the behaviour of a use case may be permitted by the behaviour of a security UC.
Generalization	Element::Relationship::DirectedRelationship:: SecureRelationship Element::NamedElement:: SecureRelationship	
Associations	- protection:SecurityUC [1..1]. It references the use case that represents the protection and owns the protect relationship. - protectedCase:UseCase [1..1]. It references the use case that is being protected.	- permittingCase:SecurityUC [1..1]. It references the use case that represents the permission and owns the permit relationship (SecurityUC or GridSecurityUC). - permittedCase: UseCase [1..1]. It references the use case that is being permitted (UseCase).
Notation		
Tagged values	InvolvedAsset, ProtectionLevel, KindAttack	PermissionCondition, KindPermission
Constraints	- protectedCase can be of the kind UseCase or GridUC. context Protect inv: (self.protectedCase->isTypeOf(UseCase) or self.protectedCase->isTypeOf(GridUC)) - protection can be of the kind SecurityUC or GridSecurityUC. context Protect inv: (self.protection->isTypeOf(SecurityUC) or self.protection->isTypeOf(GridSecurityUC)) - It defines one value for all the tagged values context Protect inv: self.InvolvedAsset->size()=1 inv: self.ProtectionLevel->size()=1 inv: self.KindAttack->size()=1 - InvolvedAsset must be the same that the InvolvedAsset of the protected case. context Protect inv: self.InvolvedAsset= self.protectedCase.InvolvedAsset - KindAttack must be the same that the KindAttack of the attacked use case. context Protect inv: self.IKindAttack = self.protectedCase.KindAttack	- permittedCase can be of the kind GridUC or GridSecurityUC. context Permit inv: (self.permittedCase->isTypeOf(GridUC) or self.permittedCase->isTypeOf(GridSecurityUC)) - permittingCase can be of the kind SecurityUC or GridSecurityUC. context Permit inv: (self.permittingCase->isTypeOf(SecurityUC) or self.permittingCase->isTypeOf(GridSecurityUC)) - It defines one type of value for the KindPermission context Permit inv: self.KindPermission->size()=1
Stereotype	Mitigate	Threaten
Description	This relationship specifies that the behaviour of a misuse case may be mitigated by the behaviour of a security UC.	This relationship specifies that the behaviour of a use case may be threatened by the behaviour of a misuse case.
Generalization	Element::Relationship::DirectedRelationship:: SecureRelationship - Element::NamedElement:: SecureRelationship	
Associations	- mitigation:SecurityUC [1..1]. It references the use case that represents	- threatenedCase: UseCase [1..1]. References the use case that is being



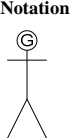
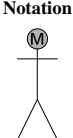
	the mitigation and owns the mitigate relationship (SecurityUC or GridSecurityUC) - mitigatedCase: MisuseCase [1..1]. It references the use case that is being mitigated (MisuseCase).		threatened (UseCase or GridUC). - threateningCase: MisuseCase [1..1]. References the use case that represents the threat and owns the threaten relationship	
Notation				
Tagged values	SuccessPercentage, KindCountermeasure		SuccessPercentage, KindVulnerability, KindAttack	
Constraints	- mitigation can be of the kind SecurityUC or GridSecurityUC. context Mitigate inv: (self.mitigation->isTypeOf(SecurityUC) or self.mitigation->isTypeOf(GridSecurityUC)) - It defines some type of value for all the tagged values context Mitigate inv: self.SuccessPercentage->size()=1 inv: self.KindCountermeasure->size()=1		- threatenedCase only can be UseCase or GridUC. context Threaten inv: (self.threatenedCase->isTypeOf(UseCase) or self.threatenedCase->isTypeOf(GridUC)) - It defines some type of value for all the tagged values context Threaten inv: self.SuccessPercentage->size()=1 inv: self.KindVulnerability->size()=1 inv: self.KindAttack->size()=1	
Stereotype	GridActor		MisActor	
Description	This actor specifies a role played by a Grid user or any other Grid system that interacts with the subject.	Notation 	This actor specifies a role played by an attacker or misuser or any other attack that interacts with the subject	Notation 
Generalization	Classifier::Actor		Classifier::Actor	
Associations	It has associations with UseCase, GridUC, SecurityUC and GridSecurityUC	«GridActor»	It only has associations with MisuseCase	«MisActor»
Tagged values	KindGridCredential, KindGridActor, KindRole, OrganizationName		KindMisActor, HarmDegree	
Constraints	- It defines some type of value for some tagged values context GridActor inv: self.KindGridCredential->size()>=0 inv: self.KindGridActor->size()=1 inv: self.KindRole->size()=1 inv: self.OrganizationName->size()>=1 - It does not have association with MisuseCase context GridActor inv: self.MisuseCase->size()=0		- It can only associate with MisuseCase context Misuser inv: self.MisuseCase->size()>=1 inv: self.GridUC->size()=0 inv: self.SecurityUC->size()=0 - It should define one type for all the tagged values context Misuser inv: self.KindMisuser->size()=1 inv: self.HarmDegree->size()=1	

Table 2. Tagged Values of the stereotypes defined in the GridUCSec package

Tagged Value	Description	Type	Used in
GridRequirement	It contains the types of requirement involved in the UC	RequirementType	«UseCase», «GridUC»
HarmDegree	It defines the degree of harm that an attacker can cause in the system	LevelType	«MisActor»
ImpactLevel	It indicates the level of impact in the system that can cause if some of the threats carry out an attack with success	LevelType	«MisuseCase»
InvolvedAsset	It identifies the assets that must be protected and that they take part in the realization of the use case.	AssetType	«GridUC», «MisuseCase», «SecurityUC», «GridSecurityUC»
KindAttack	It describes the type of attack that is carried out over the system for opening a gap of security.	AttackType	«Protect», «Threaten», «MisuseCase»
KindCountermeasure	It describes the decisions of how to protect the security and the privacy of the potential attacker and of the vulnerabilities	String	«Mitigate»
KindGridActor	It defines the type of Grid actor that interacts with the system	GridActorType	«GridActor»
KindGridCredential	When a Grid actor wants to interact with the system, this must present or support a type of credential of security.	CredentialType	«GridActor»
KindMisActor	It describes the type of attacker that involves in the system with the purpose of damage it.	AttackerType	«MisActor»
KindPermission	It indicates the type of permission granted for the realization of a use case	PermissionType	«Permit»
KindRole	It indicates the role of an actor in the system, so the privileges associated to the actor are known.	String	«GridActor»
KindVulnerability	It identifies the types of vulnerabilities found in the system and that are possible candidates of attack.	String	«Threaten»
OrganizationName	All users must belong to some organization	String	«GridActor»
PermissionCondition	It defines the necessary conditions under which a use case permits the realization of other use case	String	«Permit»
ProtectionLevel	It indicates the level of protection that the use case should have	LevelType	«UseCase», «GridUC», «Protect»
RiskLevel	The level of risk is considered in function of threats and vulnerabilities of the assets	LevelType	«MisuseCase»
SecurityDegree	It describes the degree of security that this use case contributes to the system.	LevelType	«SecurityUC», «GridSecurityUC»
SecurityDependence	It indicates if a use case need be ensured because its behaviour can generate a risk for the system.	LevelType	«GridUC», «GridSecurityUC»

SecurityRequirement	It contains the types of security requirements involved in the security use cases	RequirementType	«SecurityUC», «GridSecurityUC»
SuccessPercentage	It indicates the percentage that a certain action (of security, of attack, etc.) has success.	LevelType	«Threaten», «Mitigate»
ThreatLikelihood	It is the likelihood that a threat is carried out.	FrequencyType	«MisuseCase»

Table 3. Stereotypes of Types defined in TypesGridUCSec package

Type	Values
AssetType	{Accounting}, {Credential}, {Data}, {General}, {Identity}, {Message}, {Resource}, {User}
AttackType	{AccesControlAtt}, {ColludingAtt}, {DefeatingAtt}, {DoSAtt}, {EavesdroppingAtt}, {IntruderAtt}, {MaliciousAtt}, {MasqueradingAtt}, {ObjectReuseAtt}, {SniffingAtt}
AttackerType	{hacker}, {cracker}, {script kiddies}, {newbies}, {lamers}, {virus}, {trojan}
CredentialType	{UserPass}, {X509}, {Kerberos}, {SAML}, {PIN}, {Biometric}
FrequencyType	{Vfrequent}, {Frequent}, {Normal}, {Rare}
GridActorType	{31}, {MobileUser}, {Service}, {MobileResource}, {Resource}, {VO}, {Host}
LevelType	{VHigh}, {High}, {Medium}, {Low}, {VLow}
PermissionType	{Execute}, {CheckExecute}, {Interact}, {Include}, {Extend}, {Protect}, {Mitigate}
RequirementType	{Accounting}, {Anonymity}, {Antivirus}, {Authentication}, {AuthenticationMutual}, {Authorization&AC}, {Availability}, {Confidentiality}, {Credential}, {Delegation}, {Firewall&IntrusionPrevention}, {Integration}, {Integrity}, {Interoperability}, {MappingIdentity}, {MultipleImplementation}, {NonRepudiation}, {Privacy}, {Revocation}, {Scalability}, {SSO}, {Trust}, {Usability}



David G. Rosado holds a Ph.D. in Computer Science from University of Castilla-La Mancha and has an MSc in Computer Science from the University of Málaga (Spain). He is Assistant Professor at the Escuela Superior de Informática of the University of Castilla-La Mancha at Ciudad Real (Spain). His research activities are focused on security architectures for Information Systems and Mobile Grid Computing. He has published several papers in national and international conferences on these subjects. He is a member of the GSyA research group of the Information Systems and Technologies Department at the University of Castilla-La Mancha.



Javier Lopez received his M.S. and Ph.D. degrees in computer science in 1992 and 2000, respectively, from the University of Malaga, where he currently is a Full professor. His research activities are mainly focused on network security and critical information infrastructures, and he leads national and international research projects in those areas. He is also Co-Editor in Chief of Springer’s International Journal of Information Security (IJIS), a member of the editorial boards of international journals, and the Spanish representative on the IFIP Technical Committee 11 on security and protection in information systems.



Eduardo Fernández-Medina holds a Ph.D. and a MSc. in Computer Science from the University of Sevilla. He is Associate Professor at the Escuela Superior de Informática of the University of Castilla-La Mancha at Ciudad Real (Spain), his research activity being in the field of security in databases, datawarehouses, web services and information systems, and also in security metrics. He is co-editor of several books and chapter books on these subjects, and has several dozens of papers in national and international. Author of several manuscripts in national and international journals (Information Software Technology, Computers And Security, Information Systems Security, etc.), he is a member of the GSyA research group of the Information Systems and Technologies Department at the University of Castilla-La Mancha. He belongs to various professional and research associations (ATI, AEC, ISO, IFIP WG11.3 etc.)