Reusable Security Use Cases for Mobile Grid environments

David G. Rosado[1], Eduardo Fernández-Medina[1] y Javier López[2]
[1] *University of Castilla-La Mancha. Group Research Alarcos - Information Systems and Technologies Institute. Information Systems and Technologies Department, ESI, Paseo de la Universidad 4, 13071 Ciudad Real*
{David.GRosado, Eduardo.FdezMedina}@uclm.es
[2] *Computer Science Department, University of Málaga, jlm@lcc.uma.*

Abstract

Due to the growing complexity of software development, developing software through systematic processes is becoming more and more important. 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. In the last years, GRID technology has shown to be the most important one and it allows 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). Traditionally, systems based on GRID Computing have not been developed through adequate methodologies and have not taken into account security requirements throughout their development, only offering security technical solutions at the implementation stages. This paper shows part of a development methodology that we are elaborating for the construction of information systems based on Grid Computing highly dependent on mobile devices where security plays a very important role. Specifically, in this paper, we will present the analysis phase, managed by reusable use cases through which we can define the requirements and needs of these systems obtaining an analysis model that can be used as input to the following phase of the methodology, the design phase of mobile Grid systems.

1. Introduction

The idea of developing software through systematic development processes to improve software quality is not new. 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 [1]. That is to say, generic development processes are used to develop specific systems without taking into consideration either the subjacent technological environment or the special features and particularities of these specific systems.

Additionally, the growing need for constructing secure systems, mainly due to the new vulnerabilities derived from the use of 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 [2-7]. The main reason is that, traditionally, security aspects are only considered at the implementation stages causing that security solutions are not perfectly coupled with the design and the rest of requirements of the system [7, 8]. Model Driven Security [9] 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 based on Grid Computing are a kind of systems that have clear differentiating features where security is a very important aspect. Grids are centered on sharing resources between dynamic collections of individuals, institutions and resources in a flexible, secure and coordinated way [10]. Grid environments have special features that make them different from other systems and that we must consider throughout the whole development lifecycle. These features are, for example, user population, resources pool, and the fact that the groups of processes running on different sites are potentially large and dynamic. Also, we must consider that processes may communicate by a variety of mechanisms such as unicast or multicast, and different authentication and authorization mechanisms can be present in a single job computation, according to the local security policies of the sites involved. Finally, individual users may be associated with different local name spaces, credentials and accounts on different sites [11].

978-1-4244-3725-2/09/$25.00 © 2009 IEEE

On the other hand, today, the development of wireless technology and mobile devices enables us to access the network service from anywhere at any time [12]. Mobile Grid, in relevance to both Grid and Mobile Computing, is a full inheritor of Grid with the additional feature of supporting mobile users and resources in a seamless, transparent, secure and efficient way [13-15]. Mobile Grid allows both the mobility of the users requesting access to a fixed Grid and that of the resources that are themselves part of the Grid. Both cases have their own limitations and constraints that should be handled [16].

Security has been a central issue in grid computing from the outset, and has been regarded as the most significant challenge for grid computing [17, 18]. Security over the mobile platform is more critical due to the open nature of wireless networks. In addition, security is more difficult to implement into a mobile platform due to the limitations of resources in these devices [19]. 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 [1, 20]. The lack of adequate development methods for this kind of systems has encouraged us to build a methodology to develop them (see Figure 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 and services and to the use of mobile devices as Grid nodes.

The proposal, as a whole, is very wide. For that reason, in this paper, we will present reusable security use cases that can be used at the analysis stage to build use case diagrams integrating the requirements of specific mobile Grid applications.

The rest of paper is organized as follows: In section 2, we present the related work. Section 3 summarizes briefly the proposed methodology. In section 4, stereotypes and associations of Grid use cases will be define. In section 5, diagrams of reusable use cases will be presented. 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 [21-24]. 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 [1]. 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 [3, 25-28], even from the first stages, but however, none of them are defined for Grid Computing based systems. UMLSec [28], 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, it has been just recently given attention to integrate two emerging techniques of mobile and grid computing, for example, in [15, 29-32], although they do not elaborate on how the mobile devices may be incorporated in the current grid architecture. 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 [33-35]. In Figure 1 we can see the definition of the methodology using SPEM (Software & Systems Process Engineering Metamodel) versión 2.0 [36].

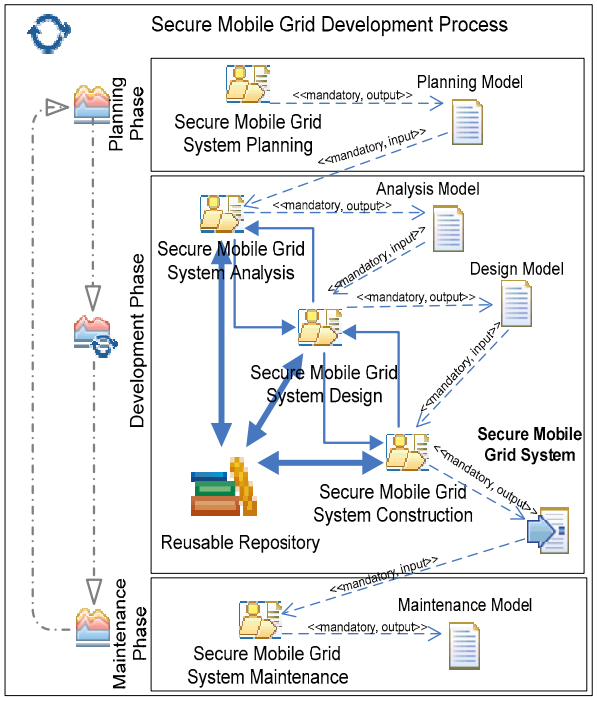

Figure 1. Development methodology for secure Mobile Grid systems with SPEM 2.0

This systematic engineering process will mainly face two great challenges: On the one hand, to establish a methodology for the secure development of Grid systems, considering the functional needs as well as the non-functional ones, especially security not only of the system to be constructed but also of the needs arising when implementing it using the Grid technology. On the other hand, the second challenge to solve is the use of mobile devices in Grid systems, with all the difficulties that constructing a Grid infrastructure that supports mobile devices entails, due to the limitations and features of these devices.

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.

4. Grid Use Cases

The analysis stage is supported by repositories where we can find several types of elements: First of all, the elements that have been developed in earlier stages; in the second place, those that have been built at the beginning of the process and finally those that come from other executions of the process where we have obtained elements that can be reused by other applications. Reuse is adequate here thanks to the common features of applications based on Grid computing (CPU intensive, data intensive, collaborative and so on) as well as to the fact that these applications use mobile devices. Therefore, we must abstract all the common features (by analyzing the main features of Grid applications and constructing, for example, generic use case diagrams where all these common features are represented) and make them available for the methodology (through the repository) in order to be able to use, at any stage, the common elements and adapt them to our needs. This also facilitates the use of automatic tools which made the work easiest.

The analysis stage is centred 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 stage is supported by the reuse of Grid use cases stored in the repository (see Figure 2) where we obtain, of way automated, 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. This repository is updated (adding, deleting or modifying elements) during all the life cycle of the process.

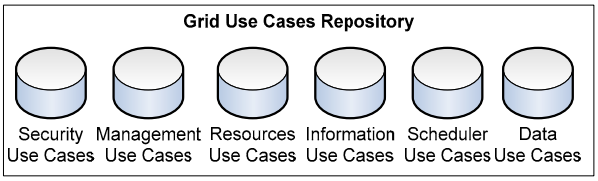

Figure 2. Repository of Use Cases for Mobile Grid environments

Among all the use cases defined in the repository, we will focus on the security use cases. Later, we will define the stereotypes and associations identified for constructing security use case diagrams for mobile Grid environments.

4.1. Use Cases stereotypes

To define reusable use case diagrams 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. Initially, we have defined a series of stereotypes to build our use case diagrams for Grid systems that are different from the rest of use cases in their behaviour, restrictions, associations and attributes that will be defined throughout the research. The way to use each one of them will be studied in this paper through examples while a more detailed description of the semantics of the stereotypes will be dealt with in further works.

In Figure 3, we can see the stereotypes that have been defined for mobile Grid systems. New use case stereotypes are defined; first of all, *<<GridUseCase>>* that defines the common behaviour of Grid systems and secondly, *<<MisuseCase>>* that defines the behaviour of threats and attacks within the system [37, 38], and *<<MobileUseCase>>* that indicates the behaviour of the mobile devices. Furthermore, we define *<<SecurityUseCase>>* that represents security use cases and *<<GridSecurityCase>>* that inherits from *<<SecurityCase>>* and *<<GridUseCase>>* and defines security use cases for Grid environments.

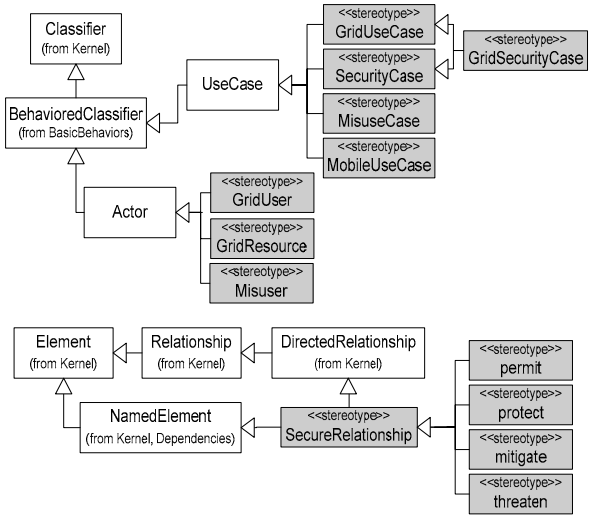

Figure 3. Stereotypes for Grid Use Cases

We define the following stereotypes of actors within the Grid domain: *<<GridUser>>* that identifies users (mobile users, administrators, virtual organizations, etc), *<<GridResource>>* that identifies all resources forming these Grid systems (mobile devices, servers, repositories, services, domains etc), and *<<Misuser>>* that initiates misuse cases, either intentionally or inadvertently.

At last, we define the stereotypes of relationships, inherited from the *DirectedRelationship* and *NamedElement* metaclasses, which define the relationships existing between all the use cases that can take part in the global diagram for mobile Grid systems. We can define four types of relationships, (*<<permit>>*, *<<protect>>*, *<<mitigate>>* and *<<threaten>>*); all of them inheriting from *<<SecureRelationship>>* and defining the security relationships between the different use cases defined (*<<GridUseCase>>*, *<<MisuseCase>>*, *<<MobileUseCase>>*, <<Grid*SecurityCase*>> and *<<SecurityCase>>*) that will be dealt with in the following section.

4.2. Associations between use cases

In Figure 4, we can see the associations between these new stereotypes defined for constructing use case diagrams for mobile Grid systems. The stereotype *<<GridSecurityCase>>* inherits the relationships of *<<SecurityCase>>* that are the relationships with *<<permit>>*, *<<protect>>* and *<<mitigate>>*. Our purpose is that the stereotype *<<GridUseCase>>* and *<<MobileUseCase>>* are the only ones inheriting the relationships from *UseCase* (shown in Figure 4). To do so, we must define restrictions indicating the elimination of these relationships (*<<permit>>*, *<<protect>>* and *<<threaten>>*) from the inherited types (*<<SecurityCase>>* and *<<MisuseCase>>*).

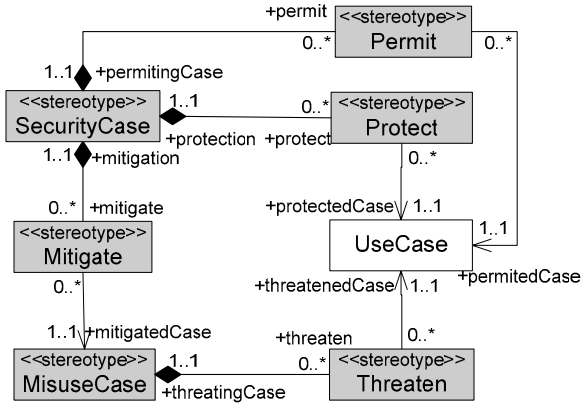

Figure 4. Relation between UseCase and DirectedRelationship.

The stereotype *<<protect>>* specifies that the origin security use case protects the destination use case. Therefore, it has two associations, an association *protection* indicating the security use case (*<<SecurityCase>>*) that represents the protection and owns the protect relationship, and other association *protectedCase* indicating the use case that is being protected (*<<UseCase>>*). *SecurityCase* is associated with *<<protect>>* indicating the protect relationships owned by this security use case.

The stereotype *<<permit>>* establishes permission for the performance of the destination use case of the relationship. It has two associations, an association

permitingCase indicating the security use case that represents the permission and owns the permit relationship, and other association *permittedCase* indicating the use case that is being permitted. *SecurityCase* is associated with <<*permit*>> indicating the permit relationships owned by this security use case.

The stereotype <<*mitigate*>> represents the origin use case as prevention against the destination use case (misuse case). It has two associations, an association *mitigation* indicating the security use case that represents the mitigation and owns the mitigate relationship, and other association *mitigatedCase* indicating the misuse case that is being mitigated. *SecurityCase* is associated with <<*mitigate*>> indicating the mitigate relationships owned by this security use case.

Finally, the stereotype <<*threaten*>> specifies that the destination use case is threatened by the origin use case of the relationship (misuse case). It has two associations, an association *threatingCase* indicating the misuse case that represents the threat and owns the threaten relationship, and other association *theratenedCase* indicating the use case that is being threatened. *MisuseCase* is associated with <<*threaten*>> indicating the threaten relationships owned by this misuse case.

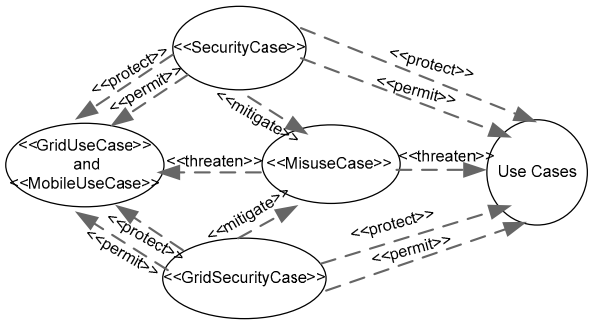

Figure 5. Associations between Use Cases

In Figure 5, the associations between the different use cases that take part in the use case diagrams of the application to be built are shown through the graphical notation for the different use case stereotypes that we have previously defined.

5. Diagrams of reusable Use cases

Let us show an example, where all previously defined use cases take part and in which we can see how to build use case diagrams and security use cases that protect use cases and prevent misuse cases started by attackers. Let us suppose that a Grid user wants to obtain information (picture) stored in a Grid resource (mobile device) and an attacker that has no access rights tries to obtain this information from the resource. In this example, we suppose that the user has been previously authenticated and we are sure that his/her credentials and identity are authentic. To prevent this attack, we must check the attributes and privileges provided by the user and make the decision on allowing or denying access. This function is presented by the security use case *Authorize access*. This Grid security use case is reusable since it allows us to establish the security relationships (<<*permit*>>, <<*protect*>> and <<*mitigate*>>) with the rest of diagram use cases, making it possible that security is incorporated into the use case final diagram of the application.

This reusable use case is extracted from the repository together with other security use cases related to it such as the security use cases representing the need to negotiate with the policies of the participating domains and interoperate with them to exchange information. In addition, we have to take into account not only the management of identity and credentials but also the mobility management to control the location of the mobile device where the required information is stored (picture), etc. In Figure 6, we can see the use cases taking part in this simple scenario. The so-called reusable use cases are use cases that define a reusability tagged value, allowing us to establish relationships with the different use cases of the specific application. In this way, only the reusable use cases (those with the reusability property) are extracted from the repository to be used in the construction of the use case diagram of the application, implicitly obtaining all relationships that such use case establishes within the repository without the participation of the analyst.

Other more complex cases of security use cases diagrams can include authentication, confidentiality, credentials and identity management, etc. All these security use cases can be extracted from the repository that has been previously defined and facilitate the construction of use case diagrams of the application independently of the level of complexity and the required security needs. Security use cases are integrated within the Grid security use case diagram, indicating that they provide us with common security aspects necessary to carry out the security functions specific for the Grid and all the set forms the security use case diagram for Grid systems.

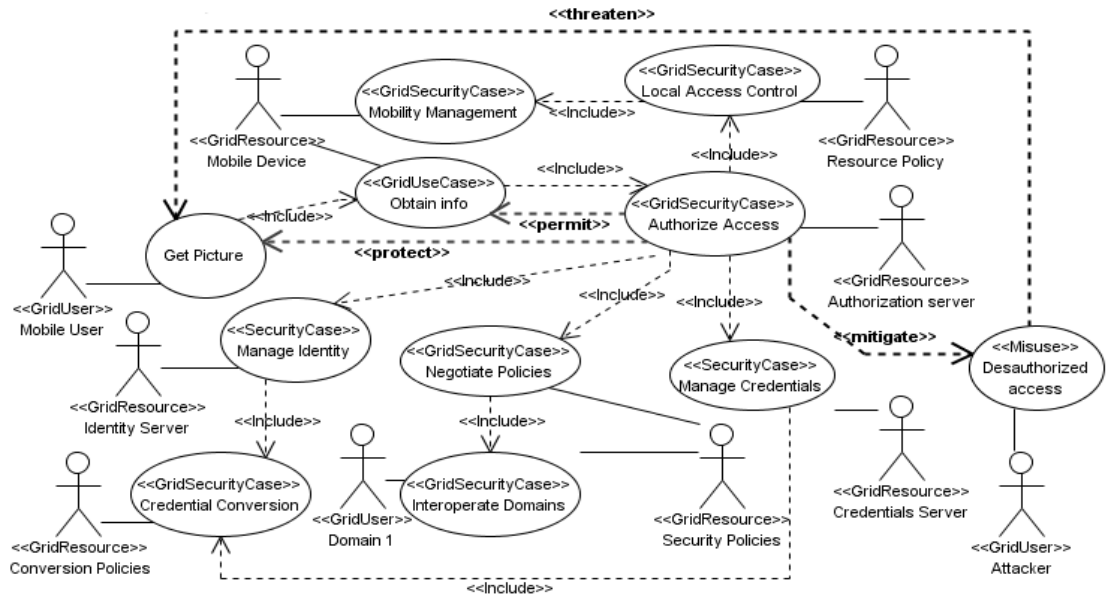

Figure 6. Use Case "Get Picture" protected by "Authorize access". Associations with others security use cases and misuse case.

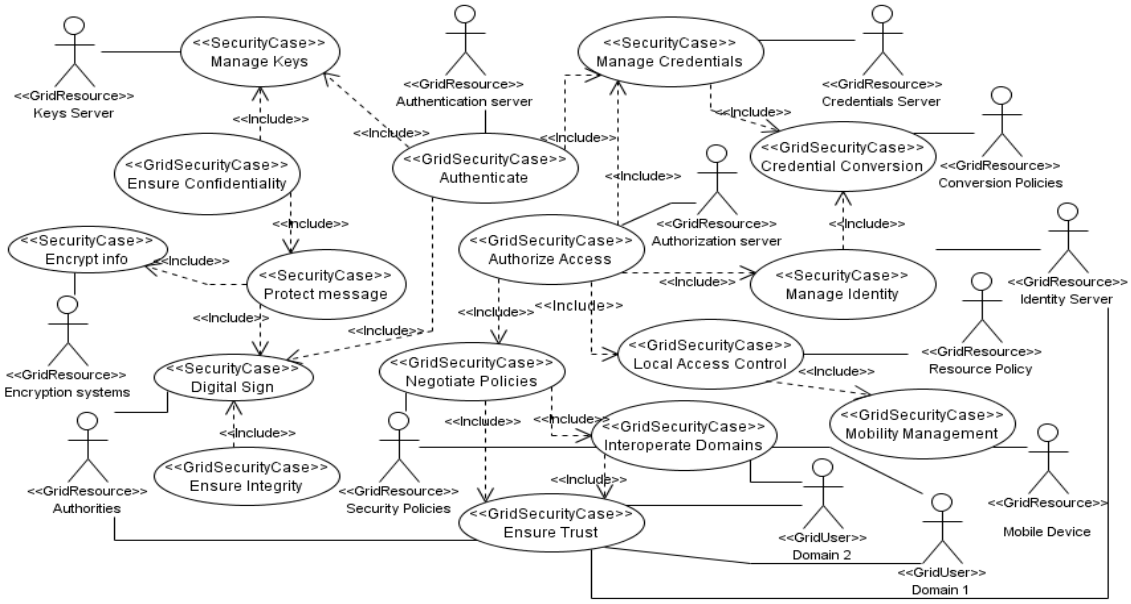

Figure 7. Repository of Security use cases for Mobile Grid environments

In the diagram shown in Figure 7, that we have reduced due to space constraints, we can see the repository of security use cases that we can reuse at the analysis stage to build the use case diagrams of the application to be constructed. This diagram is completed with other security use cases such as Establish Trust, Delegate privileges, Ensure Non-repudiation, Ensure Privacy, Protection Device, Mobility, etc and the relationships between them and actors.

6. Conclusions and future work

The complexity of current applications forces us to planify 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. To build a

secure Grid system, we have defined a methodology that, apart from developing a Grid system, allows us to incorporate all Grid security aspects into the lifecycle thus obtaining a secure end product.

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. The development of mobile Grid system is a complex and tedious task. For that reason, firstly, with a methodology and secondly with reuse, we can reduce time and effort in the development of this kind of systems.

As future work, we are aimed at finishing the detail of this methodology (activities, tasks, etc) through the research-action method, integrating common security requirements engineering techniques, linking with others approaches for security as UMLSec, and defining the traceability of artefacts and starting from use cases, identifying services within the architecture to arrive at any implementation platform (Globus, etc). Our methodology is being validated through a real case application defined within the GREDIA European project.

7. Acknowledgment

This research is part of the following projects: QUASIMODO (PAC08-0157-0668) financed by the "Viceconsejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha" (Spain), and ESFINGE (TIN2006-15175-C05-05) granted by the "Dirección General de Investigación del Ministerio de Educación y Ciencia" (Spain).

8. References

[1] R. Kolonay and M. Sobolewski, "Grid Interactive Service-oriented Programming Environment," presented at Concurrent Engineering: The Worldwide Engineering Grid, Tsinghua, China, pp. 97–102, 2004.

[2] L. Bass, F. Bachmann, R. J. Ellison, A. P. Moore, and M. Klein, "Security and survivability reasoning frameworks and architectural design tactics," *SEI*, 2004.

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

[4] T. Lodderstedt, D. Basin, and J. r. Doser, "SecureUML: A UML-Based Modeling Language for Model-Driven Security," Dresden, Germany, pp. 426-441, 2002.

[5] R. Breu, K. Burger, M. Hafner, J. Jürjens, G. Popp, V. Lotz, and G.Wimmel, "Key issues of a formally based process model for security engineering," presented at International Conference on Software and Systems Engineering and their Applications, 2003.

[6] C. B. Haley, J. D. Moffet, R. Laney, and B. Nuseibeh, "A framework for security requirements engineering," presented at Software Engineering for Secure Systems Workshop, Shangai, China, pp. 35-42, 2006.

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

[8] C. Artelsmair and R. Wagner, "Towards a Security Engineering Process," presented at The 7th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, Florida, USA, 2003.

[9] D. Basin, J. Doser, and T. Lodderstedt, "Model driven security for process-oriented systems," presented at ACM Symposium on Access Control Models and Technologies, Como, Italy, pp. 100-109, 2003.

[10] I. Foster and C. Kesselman, *The Grid2: Blueprint for a Future Computing Infrastructure*. San Francisco, CA: Morgan Kaufmann Publishers; 2 edition, 2004.

[11] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A Security Architecture for Computational Grids," presented at 5th Conference on Computer and Communications Security, San Francisco, USA, pp. 83-92, 1998.

[12] D. Bruneo, M. Scarpa, A. Zaia, and A. Puliafito, "Communication paradigms for mobile grid users," presented at 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'03). pp. 669 - 676, 2003.

[13] A. Litke, D. Skoutas, and T. Varvarigou, "Mobile Grid Computing: Changes and Challenges of Resourse Management in a Mobile Grid Environment," presented at 5th International Conference on Practical Aspects of Knowledge Management (PAKM 2004), 2004.

[14] T. Guan, E. Zaluska, and D. D. Roure, "A Grid Service Infrastructure for Mobile Devices," presented at First International Conference on Semantics, Knowledge, an Grid (SKG 2005), Beijing, China, 2005.

[15] H. Jameel, U. Kalim, A. Sajjad, S. Lee, and T. Jeon, "Mobile-To-Grid Middleware: Bridging the gap between mobile and Grid environments," presented at European Grid Conference EGC 2005, Amsterdam, The Netherlands, pp. 932-941, 2005.

[16] S.-M. Park, Y.-B. Ko, and J.-H. Kim, "Disconnected Operation Service in Mobile Grid Computing," presented at International Conference on Service Oriented Computing (ICSOC'2003), Trento, Italy, 2003.

[17] M. Humphrey, M. R. Thompson, and K. R. Jackson, "Security for Grids," *Lawrence Berkeley National Laboratory. Paper LBNL-54853*, 2005.

[18] A. Chakrabarti, A. Damodaran, and S. Sengupta, "Grid Computing Security: A Taxonomy," *IEEE Security & Privacy*, vol. 6 (1), pp. 44-51, 2008.

[19] P. G. Bradford, 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, Ed. The University of Alabama, Tuscaloosa, USA, pp. 440, 2007.

[20] H. Dail, 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*, pp. 73-98, 2004.

[21] D. C. Schmidt, "Model-Driven Engineering," *IEEE Computer*, vol. 39 (2), 2006.

[22] P. Kruchten, *The Rational Unified Process: An Introduction*, 2nd ed: Addison-Wesley, 2000.

[23] I. Jacobson, G. Booch, and J. Rumbaugh, *The Unified Software Development Process*: Addison-Wesley Professional, 1999.

[24] I. Flechais, M. A. Sasse, and S. M. V. Hailes, "Bringing Security Home: A process for developing secure and usable systems," presented at Nwe Security Paradigms Workshop (NSPW'03), Ascona, Switzerland, 2003.

[25] J. Jurjens, "Towards Development of Secure Systems Using UMLsec," presented at Fundamental Approaches to Software Engineering (FASE/ETAPS), 2001.

[26] J. Jurjens, "UMLsec: Extending UML for Secure Systems Development," presented at 5th International Conference on the Unified Modeling Language (UML), Dresden, Germany, pp. 1-9, 2002.

[27] C. Steel, 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*: Prentice Hall PTR/Sun Micros, pp. 1088, 2005.

[28] G. Popp, J. Jürjens, G. Wimmel, and R. Breu, "Security-Critical System Development with Extended Use Cases," presented at Tenth Asia-Pacific Software Engineering Conference (APSEC'03), 2003.

[29] D. Chu and M. Humphrey, "Mobile osgi.net: Grid computing on mobile devices," presented at 5th IEEE/ACM International Workshop on Grid Computing -Grid2004 (at Supercomputing 2004), 2004.

[30] L. Kwok-Yan, Z. Xi-Bin, C. Siu-Leung, M. Gu, and S. Jia-Guang, "Enhancing Grid Security Infrastructure to Support Mobile Computing Nodes," *Lecture Notes in Computer Science*, vol. 2908/2003 (pp. 42-54, 2004.

[31] T. Phan, L. Huang, and C. Dulan, "Challenge: Integrating Mobile Wireless Devices Into the Computational Grid," presented at 8th annual international conference on Mobile computing and networking (MobiCom'02), Atlanta, Georgia, USA, pp. 271 - 278, 2002.

[32] B. a. M. H. Clarke, "Beyond the 'Device as Portal': Meeting the Requirements of Wireless and Mobile Devices in the Legion Grid Computing System,"" presented at Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing at the International Parallel and Distributed Processing Symposium., 2002.

[33] D. G. Rosado, E. Fernández-Medina, J. López, and M. Piattini, "PSecGCM: Process for the development of Secure Grid Computing based Systems with Mobile devices," presented at International Conference on Availability, Reliability and Security (ARES'08), Barcelona, Spain, pp. 136-142, 2008.

[34] D. G. Rosado, E. Fernández-Medina, J. López, and M. Piattini, "Engineering Process Based On Grid Use Cases For Mobile Grid Systems," presented at The Third International Conference on Software and Data Technologies- ICSOFT 2008, Porto, Portugal, pp. 146-151, 2008.

[35] D. G. Rosado, E. Fernández-Medina, and J. López, "Obtaining Security Requirements for a Mobile Grid System," *International Journal of Grid and High Performance Computing*, pp. (to be published in April 1, 2009), 2008.

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

[37] G. Sindre and A. L. Opdahl, "Capturing Security Requirements by Misuse Cases," presented at 14th Norwegian Informatics Conference (NIK'2001), Tromsø, Norway, 2001.

[38] L. Røstad, "An extended misuse case notation: Including vulnerabilities and the insider threat," presented at XII Working Conference on Requirements Engineering: Foundation for Software Quality, Luxembourg, 2006.