Information and Software Technology 52 (2010) 517–536

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Analysis of Secure Mobile Grid Systems: A systematic approach

David G. Rosado [a], Eduardo Fernández-Medina [a,*], Javier López [b], Mario Piattini [a]

[a] University of Castilla-La Mancha, Alarcos Research Group-Information Systems and Technologies Institute, Information Systems and Technologies Department, ESI, Paseo de la Universidad 4, 13071 Ciudad Real, Spain
[b] University of Málaga, Computer Science Department, Spain

ARTICLE INFO

Article history:
Received 26 April 2009
Received in revised form 3 October 2009
Accepted 7 January 2010
Available online 20 January 2010

Keywords:
Secure mobile Grid development
Requirements Analysis
Reusable use cases
Security

ABSTRACT

Developing software through systematic processes is becoming more and more important due to the growing complexity of software development. 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. Systems which are based on Grid Computing are a kind of systems that have clear differentiating features in which security is a highly important aspect. The Mobile Grid, which is relevant to both Grid and Mobile Computing, is a full inheritor of the Grid with the additional feature that it supports mobile users and resources. A development methodology for Secure Mobile Grid Systems is proposed in which the security aspects are considered from the first stages of the life-cycle and in which the mobile Grid technological environment is always present in each activity. This paper presents the analysis activity, in which the requirements (focusing on the grid, mobile and security requirements) of the system are specified and which is driven by reusable use cases through which the requirements and needs of these systems can be defined. These use cases have been defined through a UML-extension for security use cases and Grid use cases which capture the behaviour of this kind of systems. The analysis activity has been applied to a real case.

1. Introduction

The growing need to construct secure systems, mainly due to the new vulnerabilities derived from the use of the Internet and that of the applications distributed in heterogeneous environments, has encouraged the scientific community to demand a clear integration of security into development processes [4,8,26,34,42,47]. In fact, for decades, the security community has carried out detailed research into specific areas of security, while largely ignoring the design process. A recurrent idea in the scientific community is that security aspects should not be blindly inserted into an IT-system, but that the overall system development should take security aspects into account. However, in reality most developers usually ignore security requirements and they are often retrofitted late in the design process or purposed separately from functional design [1], which typically leads to their applications having many security weaknesses [37]. It is intuitive that a better way to achieve secure software is to incorporate security into the software from the beginning of the development process [1,47]. The identification of security aspects in the first stages ensures a more robust development and permits the security requirements to be perfectly coupled with the design and the rest of the system's requirements. Requirements such as data confidentiality, encryption algorithms, communication protocols, encrypted messages, and delegation of credentials, are therefore specified in the analysis activity, and although some of them (such as communication protocols) are not completely detailed until the construction activity, they should be taken into account when designing the different models that make up the final product.

However, *generic* software development methodologies are not appropriate for the development of every kind of software system. For instance, generic development processes are sometimes used to develop Grid specific systems without taking into consideration either the subjacent technological environment or the special features and particularities of these specific systems. In fact, the majority of existing Grid applications have been built without a systematic development process and are based on ad hoc developments [11,38]. Moreover, systems which are based on Grid Computing have clear differentiating features [38], which suggests the need for adapted development methodologies. These features are the following: (i) computing grids are hardware and software infrastructures that support secure sharing and concurrent access to distributed services by a large number of competing users from different Virtual Organizations, (ii) in the grid, the computing

* Corresponding author. Tel.: +34 926295300; fax: +34 926295354.
E-mail addresses: David.GRosado@uclm.es (D.G. Rosado), Eduardo.FdezMedina@uclm.es (E. Fernández-Medina), jlm@lcc.uma.es (J. López), Mario.Piattini@uclm.es (M. Piattini).

0950-5849/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2010.01.002

resources are autonomously managed at different locations in a distributed manner, (iii) the Grid is a large scale resource sharing a distributed computing environment that couples thousands of computers, storage systems, networks, scientific instruments and other devices distributed over heterogeneous wide area networks [16,18], and (iv) security is a crucial aspect of Grid based systems. The lack of adequate development methods for this kind of systems has encouraged us to build a methodology with which to develop them, offering a detailed guide to analyze, design and implement them. Security is considered throughout these activities.

Mobile Computing is a generic term which describes the application of small, portable, and wireless computing and communication devices. Mobile Computing focuses on the necessity to provide access to information, communications and services everywhere, at anytime and by any available means. The technical solutions by which to achieve this are not always easy to implement [41]. Mobile Computing with networked information systems helps increase productivity and operational efficiency. This, however, comes at a price: Mobile Computing with networked information systems increases the risks to sensitive information supporting critical functions in the organization which are open to attack [60].

The Mobile Grid, which is relevant to both the Grid and Mobile Computing, is a full inheritor of the Grid with the additional feature that it supports mobile users and resources in a seamless, transparent, secure and efficient manner [24,32,41]. Grids and mobile Grids may be the ideal solution for many large scale applications since they are of a dynamic nature and necessitate transparency for users. The Grid will increase not only the job throughput and performance of the applications involved but also the utilization rate of resources by applying efficient resource management mechanisms to the vast amount of its resources [41].

Security has been a central issue in Grid Computing from the outset, and has been regarded as the most significant challenge for Grid Computing [27]. The characteristics of computational grids lead to security problems which are not completely addressed by existing security technologies for distributed systems [17,63]. These security challenges are for example, among others, the need to establish security relationships between hundreds of processes that collectively span many administrative domains (rather than establishing security relationships between a client and a server) when parallel computations acquire multiple computational resources; the fact that an individual user will be associated with different local name spaces, credentials, or accounts, at different sites, for the purposes of accounting and access control; or that multiple security domains must be able to interoperate and communicate with different policies, mechanisms and protocols defined and used in each local domain that governs the resources that belongs to the Grid. However, the growing size and profile of the Grid now require comprehensive security solutions since these are critical to the success of the endeavour [39]. Security remains one of the fundamental barriers to the adoption of Grid Computing in a wider commercial context. Grid security is a prime concern and necessity of all stakeholders, including Resource Providers, Virtual Organizations and the End-users (participants), since the resources in a Grid are expensive and the tasks accomplished and information exchanged is confidential and sensitive. Grid security is hard to achieve as the resources are dynamic, heterogeneous, geographically located and under the control of multiple administrative domains [5]. Furthermore, security in the mobile platform is even 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 [6]. A Grid infrastructure that supports the participation of mobile nodes will thus play a significant role in the development of Grid Computing. We therefore focus our research on the systematic development of secure systems which are based on Mobile Grid Computing.

In this research we deal with a wide context which we would like to limit. Firstly, security is defined as a sub-factor of software quality [28] which represents the capability of a software product to protect the information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them. The provision of security to information systems can therefore be tackled through the definition of technical solutions (e.g. by defining communication protocols to ensure confidentiality and integrity, defining an access control technique, etc.), but also by defining new techniques, methods, processes and tools which will integrate security and software engineering solutions, to enable software developers to analyze, design, implement, test, and deploy secure software systems [47]. In this research, we deal with the second approach, that is to say, the integration of security with software engineering, rather than defining new technical solutions, at least in the analysis activity whose main goal is the definition of requirement models.

Our idea is to define a complete development methodology (including new models, activities, tasks, services security architecture, transformation rules between models, etc., if necessary) to improve the quality and security of Mobile Grid Computing based systems. A preliminary publication of the methodology has been presented in [55] in which we describe our general approach. [54] provides an informal presentation of the first steps of our methodology which consists of analyzing the security requirements of mobile grid systems directed by misuse cases and security use cases, and which is applied in an actual case study in [52] from which we obtain the security requirements for a specific application by following the steps described in our methodology. We have the gone onto elicit some common requirements of these kind of systems, and these have been specified to be reused through a UML-extension of use cases [53,56].

In this paper, we advance in our methodology by defining the complete analysis activity (using SPEM 2.0 [48], one of the software process modelling standards), we define all tasks, integrate the new defined artifacts (focused on security and reuse), and allocate some of the most representative ideas of the security requirements engineering discipline [43,44]. In the development of this methodology, we apply the action–research method [13] in order to incrementally improve and refine our approach, and we are currently applying this activity to an actual case study (which is being developed in a European project). Some of the most representative models are presented at the end of this paper.

The remainder of the paper is organized as follows: Section 2 presents related work. In Section 3 we propose the analysis activity and briefly summarize the proposed methodology, showing all the components of this activity. In Section 4, we apply the analysis activity to a real case. Finally, in Section 5, we put forward our conclusions and some research lines for our future work.

2. Related work

Any discussion of software development necessitates the mention of the Rational Unified Process (RUP). RUP [40] describes how to effectively deploy commercially proven approaches to software development for software development teams, although it does not specifically address security. One extension of the Unified Process is defined in [59], in which the authors present a methodology for the integration of security into software systems which it is called the Secure Unified Process (SUP). SUP establishes the prerequirements to incorporate the fundamental principles of security. It also defines an optimized design process of security within

the life-cycle of software development. The problem is that it only offers a solution at a very high level without offering "practical mechanisms" (e.g. Grid-specific security artifacts or a security architecture of reference) that would permit it to implement the approach in a short space of time and with minimal effort. Another recent approach proposes the integration of security and systems engineering by using elements of UML within the Tropos methodology [9,47]. Secure Tropos [46] is an extension of the Tropos methodology [7] and has been proposed to deal with the modelling and reasoning of security requirements and their transformation to design that satisfies them. This approach does not support the reutilization of security requirements, which ensures fast development cycles and is based on tried and tested solutions, and helps to improve the quality of these requirements for subsequent projects. This is an important aspect for the development of complex systems such as mobile Grid systems.

Several approaches for the integration of the security in the development process for specific domains appear in the relevant literature. For example, in [15], the authors propose a methodology with which to build multilevel databases, taking into consideration aspects of security (with regard to confidentiality) from the earliest stages to the end of the development process. SEDAWA [62] is another approach that proposes a comprehensive methodology with which to develop secure Data Warehouses based on the MDA framework. This process defines security requirements from the business level, which are transformed throughout the entire Data Warehouse life-cycle. Approaches which integrate security in the development process for generic applications and systems also exist, such as for example, [19] which proposes a methodology based on aspect-oriented modelling (AOM) with which to incorporate security mechanisms into an application, and [14], whose authors explore current research challenges, ideas and approaches for employing model-driven development to integrate security into software systems development through an engineering-based approach, avoiding the traditional ad hoc security integration. None of these approaches are defined and designed for Grid Computing and none of them support mobile nodes.

A further approach [33,34,36] 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 permits the expression of security-relevant information. In [50], the authors show a methodological approach for the development of security-critical systems and they model security aspects with UMLsec, extending use cases with security aspects. This extension for use cases is achieved textually by specifying access policies which are incorporated into the use case description. This idea is compatible with our approach, which allows us to specify a rich set of security requirements applied to mobile grid applications (more details of our use case extension can be found in [51]). However, UMLsec has also been applied in the industrial context of a mobile communication system, analyzing the security aspects of this kind of systems [35]. Therefore, our methodological approach considers extended use case models to specify security requirements for mobile Grid systems at analysis level, and this use case view can be complimented with other UML diagrams (deployment, activity, classes, collaboration, etc.), using UMLsec to model the security aspects (generic and mobile) in these diagrams. A model driven architecture approach towards security engineering, called Model Driven Security, is introduced in reference [3]. This approach, called SecureUML [2], integrates role-based access control policies into a UML-based model-driven software development process, but is not focused on Grid systems.

One highly important aspect of the process of achieving secure software systems in the software development process is known as security requirements engineering, which provides techniques, methods and norms for tackling this task in the IS development cycle. The requirements elicitation and analysis that are necessary to obtain a better set of security requirements seldom take place. SQUARE (Security Quality Requirements Engineering Methodology) [43] is a model made up of nine steps in which a means to elicit, categorize and prioritize security requirements for information technology systems and applications is provided. The Comprehensive, Lightweight Application Security Process (CLASP) is a life-cycle process that suggests a number of different activities throughout the development life-cycle in an attempt to improve security. Among these is a specific approach for security requirements [21]. In [64], CLASP is compared with other secure software development processes. The Security Requirements Engineering Process (SREP) [44,45] is an asset-based and risk-driven method for the establishment of security requirements in the development of secure Information Systems. All these approaches are extremely interesting for security requirements analysis but are not directly valid in secure mobile Grid environments in which it is necessary to take into account special security features and the technical environment of Grid Computing and Mobile Computing.

In Grid Computing, researchers and practitioners have come together to develop the Grid Security Infrastructure (GSI) [20], which defines secure grid systems and is the de facto security standard in the grid community. The Globus Toolkit [20] was developed by the Grid Community and is currently the most widely used grid infrastructure, and is an open source implementation based on the Open Grid Services Architecture (OGSA) specifications [49] which uses GSI for its security implementation. This security solution is in the implementation level and only offers a middleware technology to Grid environments, which can be used in our methodology when we have to define a technological platform in the construction activity.

All of the above approaches offer interesting contributions, and our approach is based on them, but they are not sufficiently specific or tailored for the mobile Grid development paradigm, mainly because they do not deal with security requirements in mobile Grid environments, which have special security features that should be considered. Furthermore, these approaches do not support the utilization of a reusable repository in which a wide set of artifacts are stored to be integrated into any activity or task of the methodology. Finally, our methodology defines a systematic approach through which to manage both functional requirements and non-functional requirements (including security) and it is possible to incorporate different aspects and methods of other approaches within the various activities and tasks of our methodology in order to carry out the common aspects of any secure Information systems, integrating them into the secure mobile Grid development life-cycle.

3. Analysis of Secure Mobile Grid Systems

Analysis focuses on ensuring that the system's security and functional requirements are elicited, specified and modelled. In our approach, this activity is driven by use cases and supported by the reusable repository. This obtains, builds, defines and refines the use cases of the Secure Mobile Grid Systems which represent the functional and non-functional requirements of this kind of systems. A wide set of elements which are common to these systems are stored in the repository, as are secure mobile Grid use cases, interaction diagrams, UML profiles, templates, etc., which help the analyst to define all the requirements (functional and non-functional, and security in particular) and build the necessary diagrams with which to complete the analysis activity from beginning to end.

In the following subsections we shall present an overview of our methodology, and we shall then introduce the details of the anal-

ysis activity, including the related stakeholders, the necessary artifacts and the specification of its tasks.

3.1. Overview of the development methodology

The structure of our methodology follows the classical cycle, in which we find a planning phase, a development phase including analysis, design and construction and finally a maintenance phase. However, our methodology is specially designed for this kind of systems and considers their particular features. Fig. 1 shows the definition of the methodology using SPEM (Software & Systems Process Engineering Metamodel) version 2.0 [48].

Our systematic development process is iterative and incremental, so our process is of a cyclical nature in which activities are repeated in a structured manner and it proposes an understanding of the problem through successive refinements, and an incremental growth of an effective solution through several versions. New and necessary characteristics can therefore be added in each iteration of the process, and refinement of previously modeled elements can be made so that a complete final design is obtained through several iterations.

What makes this methodology different from the rest is its analysis and design models and the details of its stages, in which we define tasks and activities that are specific to mobile Grid systems in which the reuse of elements (such as use cases, security use cases, and reference security architecture, available on the repository) is a key aspect in their development and in which the Grid technological environment and Mobile Computing are taken into account and are present in each task and activity of the methodology.

The planning phase has only one activity: "Secure Mobile Grid System Planning", in which an initial capture of requirements and necessities should be carried out in order to create a development plan. In this capture of requirements and necessities, we should identify the basic functionality of the system, the domains and organizations involved, the risks to the system, the types of resources and users (mobile devices, PDAs, etc.), the main security aspects of the grid and technology considerations.

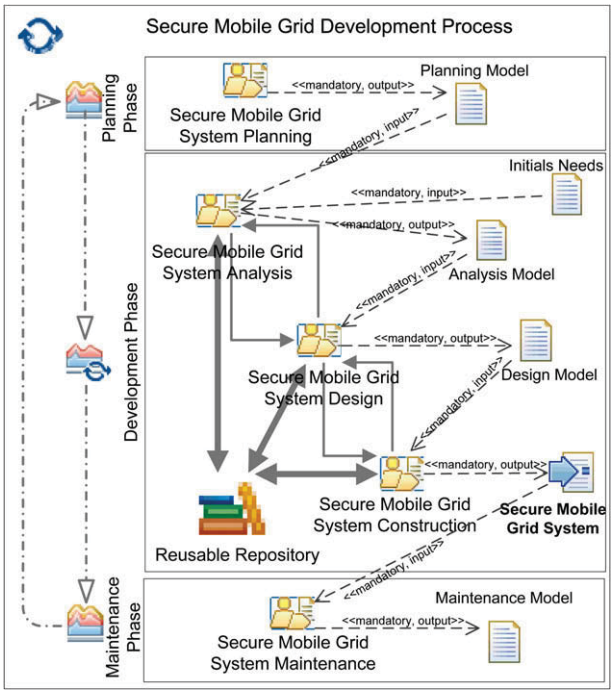

Fig. 1. Development methodology for Secure Mobile Grid Systems with SPEM 2.0.

The development phase is composed of three activities: analysis, design and construction, and their main ideas are as follows:

- The "Secure Mobile Grid System Analysis" activity is centred on identifying and analyzing the requirements and security requirements of Grid systems from a reusable use cases model in which the use case and security use cases diagrams for this kind of systems are defined. New stereotypes for use cases, actors and associations are defined to capture the behaviour of the Grid systems. These use cases and security use cases are used to identify the functional and non-functional requirements, and are refined and specified with the help of models such as class diagrams and interaction diagrams. All the diagrams that participate in the analysis belong to the analysis model, which is a detailed specification of the requirements of the system and is the input artifact for the following activity.
- In the "Secure Mobile Grid System Design" activity, we should select the structural elements of 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 provide an architectural vision of the system and contribute to the security aspects of the application that should be incorporated into the (previously constructed) reference security architecture, which offers the necessary security services to fulfil and cover the security requirements identified in the analysis model. A security architecture has been developed to be reused and redefined in particular developments. This architecture is service-oriented, and integrates a collection of security services which support the security requirements of mobile Grid environments identified in the analysis activity. This security architecture will be integrated into the software architecture, thus 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 to build the design model obtained in the last activity, and to implement the secure software architecture by defining security services together with security mechanisms and protocols for our security architecture. It is possible that the technological environment may have to be expanded to deal with mobile Grid systems.

The maintenance phase has only one activity: "Secure Mobile Grid System Maintenance" and this is a typical maintenance activity in any development process, in which a maintenance plan of the system for its later modification is defined according to the client's new necessities.

The requirements traceability, in our context, is a property of the process which allows us to link requirements with design artifacts. For example, our use cases diagrams which have been defined in the analysis activity, will be used for the definition of our reference security architecture, so each security service of our architecture will be derived (and linked) from one or more security requirements specified through our use case diagram.

3.2. Components of the analysis activity

The analysis activity is centred on capturing the requirements of the system through a use case model and with the help of a repository of secure mobile Grid use cases. As the SPEM 2.0 diagram in Fig. 2 illustrates, this activity has one mandatory input (initials needs), and two optional inputs (an analysis model built in previous iterations of the methodology and which must be refined, and a reusable UC repository from which we can obtain

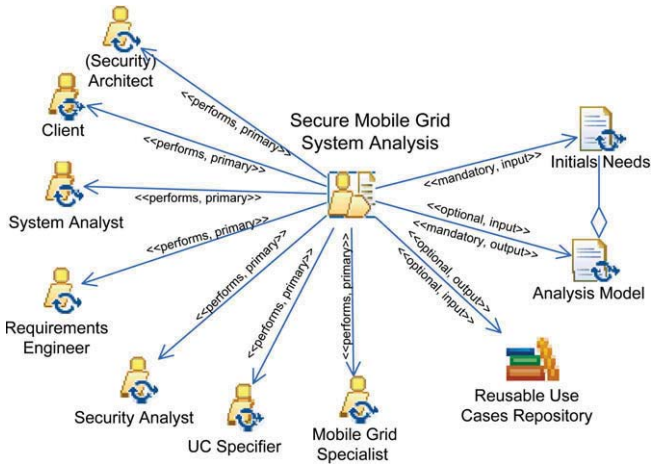

Fig. 2. SPEM 2.0 view of inputs, outputs and stakeholders of analysis activity.

information about secure mobile Grid use cases and reuse them to build a use case model). It has one mandatory output (the analysis model built or refined), and one optional output (a possible update of the repository with new or updated use cases). The stakeholders involved in this activity are the client, the system analyst, the requirements engineer, the security analyst, the UC specifier, the mobile Grid specialist and the (security) architect as primary stakeholders. The Analysis model is composed of other artifacts which can be used (as inputs or outputs) in the different tasks of this activity. Details of the set of artifacts used in this activity will be given later.

The analysis activity is based on use cases in which we define the behaviour, actions and interactions with those implied by the system (actors) to obtain a first approach to the needs and requirements (functional and non-functional) of the system to be constructed. This activity is supported by repositories in which several types of elements appear: Firstly, the elements that have been developed in earlier stages; secondly, those that have been built at the beginning of the process and finally, those that come from other executions of the process from which we have obtained elements that can be reused by other applications. Reuse is appropriate here thanks both to the common features of applications based on Grid Computing (CPU intensive, data intensive, collaborative and so on) and 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 in which all these common features are represented) and make them available for the methodology (through the repository) in order to be able to use the common elements in any activity and adapt them to our needs.

3.2.1. Stakeholders

We have identified certain stakeholders who take part in the analysis activity of this methodology. These are as follows:

- *Client*. This is an organization that requests a system, software product or software service from a provider. The client should collaborate and agree with the remaining stakeholder in this activity in order to define the initial needs and features of the system.
- *System analyst.* The system analyst leads and coordinates requirements elicitation and use-case modelling by outlining the system's functionality and delimiting the system.
- *Security analyst.* The security analyst leads and coordinates security requirements elicitation and misuse and use-case modelling, integrating them with the system requirements.
- *Requirements engineer.* The requirements engineer is responsible for the requirements specification itself. S/he is also responsible for coordinating, supervising and carrying out the analysis activity.
- *UC Specifier.* The use-case specifier details the specification of a part of the system's functionality by describing the requirements aspect of one or several use cases. The use-case specifier may also be responsible for a use-case package and for maintaining the integrity of that package. The use-case specifier responsible for a use-case package is also responsible for its use cases and actors. This stakeholder collaborates with the security analyst and mobile grid specialist to define the security use cases and Grid use cases that are defined in the analysis model.
- *Mobile grid specialist.* The specialist is responsible for the organization's security policies and for supporting the definition of security requirements. This stakeholder should contribute with all security aspects for mobile Grid environments.
- *(Security) architect.* The role of the (security) architect is to design the technical architecture that will later be implemented in the process.

3.2.2. Artifacts

An artifact is a piece of information that is produced, modified, or used by a process. Fig. 3 shows the artifacts produced and used in the analysis activity. The output artifact of this activity is the analysis model which is composed of a set of artifacts produced and used in the different tasks of this activity. We have the *Initials Needs* which are input artifacts for this activity and define initial needs and requirements that the stakeholders wish the system to cover and support functional needs, security needs and environmental needs; the *Requirements* artifact is formed of a requirements specification template, based on the IEEE std. 1233, 12207.1, 830 standards [25,61], which can be obtained from the repository and it should be instanced with specific requirements found and identified during this activity; the Analysis conflicts artifact defines the problems and errors found during this activity and which should be taken into account in future iterations of this activity for refining aspects, elements, behaviour, etc., thus improving the analysis model and, therefore, the system analysis; the *Static View* artifact represents the generic class diagrams built from the use cases using traditional software engineering methods; the *Interaction View* artifact is composed of sequence diagrams and collaboration diagrams for mobile grid environments related to mobile grid use cases. The sequence diagrams are instances of the generic sequence diagrams which are available in the repository and they should be instanced with specific elements identified in the scenarios and use cases of the application; and finally, the *Use Cases View* artifact is the most significant in this activity, and is the most innovative, with new techniques and reusable elements which can be used to produce the use cases view.

The *Use Cases View* artifact (see Fig. 3) is composed of user-defined UC diagrams which represent the use cases and actors defined by the user and capture the functional requirements of the system, the secure mobile grid UC diagrams which contain the use cases for a security context in mobile Grid environments are defined by the user using convenient reusable use cases or by defining new ones according to the requirements; and the reusable Grid UC diagrams extracted from the repository, which are diagrams tested and built in other developments and which define the common behaviour and possible scenarios in any Grid environment. For the use case view we have had to define a UML profile (GridUCSec-profile) that supports the security aspects and common behaviour found in any mobile grid system. This artifact is explained in detail in Section 3.2.2.1.

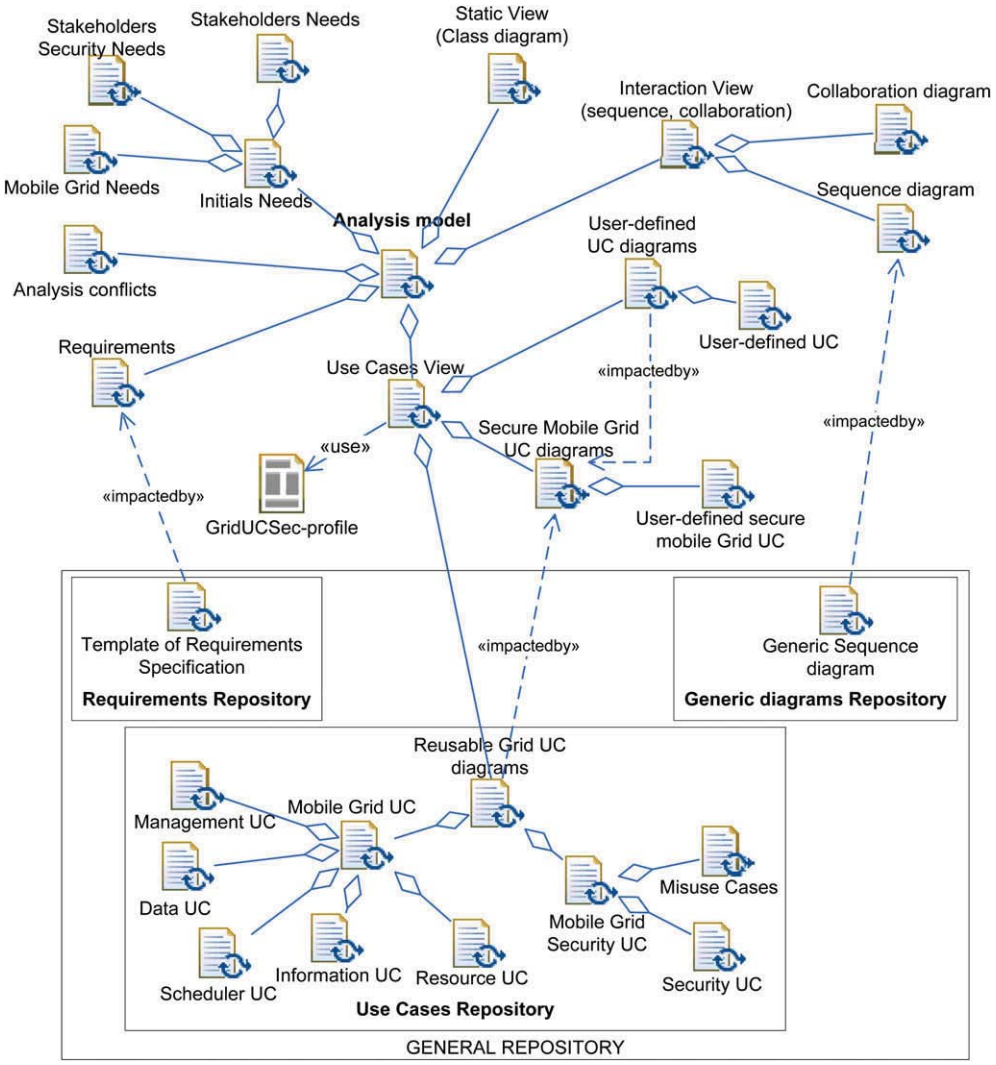

Fig. 3. Artifacts of analysis activity.

Fig. 3 shows a general repository, composed of several specific repositories. This is one extremely important element of our methodology, and this repository has been used to define a broad set of artifacts that are used in the different activities of the methodology. For the analysis activity we must use three specific repositories from the general repository (see Fig. 3): (1) the Requirements repository in which the *requirements specification template* artifact that is used in the requirements specification to define the *Requirements* artifact of the analysis model is defined; (2) the Generic diagrams repository in which generic sequence diagrams for common scenarios and Grid use cases which are instanced and transformed in specific sequence diagrams for the application to define the *Interaction View* artifact of the analysis model are defined; and (3) the Use Cases repository in which all the common use cases and security use cases for mobile Grid environments involved in the *Use Case View* artifact of the analysis model are described.

3.2.2.1. Artifact: use case view. The use case view represents the system's use case diagrams (use cases, actors and relationships), describing the behaviour and capturing the requirements for the mobile Grid system to be developed. This artifact is in turn composed of others artifacts, some of which are defined in the repository and are solution tested to help to improve and reduce time and effort in the analysis activity. The aim of this artifact is provide a vision of the system through use case diagrams by capturing the main features of the mobile Grid systems, including security aspects. To do this, it has been necessary to define a new UML profile for use cases in which all the significant features that should been take into account in any mobile Grid environment are identified. The artifacts that make up this view are as follows:

- Reusable Grid UC diagrams. This artifact defines the reusable use case diagrams for mobile Grid systems which have been built to define common scenarios and behaviour within a Grid context. This reusable artifact defines generic use case diagrams built or defined in other developments and are useful for this application because they contain common aspects that do not vary from one system to another. These diagrams are stored in the repository and can be integrated into other more complex diagrams. This artifact represents the reusable use cases, actors and the relationships between them for Secure Mobile Grid Systems and is formed of two artifacts:
  - Mobile Grid UC. This represents the use cases defined within the mobile Grid environment. These use cases may be new use cases defined by the "UC specifier" or reusable use cases defined in the repository, and this represents the common functionality and requirements of this kind of systems. The

reusable artifacts are grouped by functionality (data, resource, information, scheduler and management use cases). All these use cases are defined according to the new UML profile with new features, tagged values and constraints for mobile Grid environments. These use cases are used in the activity in which the application's final use cases diagram is built.

– Mobile Grid Security UC. This artifact differs from the previous one in that it captures the security aspects of the mobile Grid systems. This artifact is in turn composed of two reusable artifacts, security use cases and misuse cases, which show the security behaviour in these environments, identifying possible threats and attacks to the assets that we should protect by defining appropriate security requirements. These artifacts are additionally used to build the final diagram with the security aspects and are also defined in the repository according to the UML profile.

- User-defined UC diagrams. This artifact defines the application's use case diagrams in which the relationships between actors and user-defined use cases are built, including only functional aspects of the application outside the mobile grid context. This artifact is formed of user-defined use cases identified by the user by capturing the functional requirements of the application without considering aspects of the Grid environment. They are use cases which are defined through interviews and meetings with the client to identify the needs of the application which interact mainly with the end user. These artifacts are defined according to the new profile.
- Secure Mobile Grid UC diagrams. This artifact represents use case diagrams defined for the application in a mobile Grid environment, both by defining new secure mobile Grid use cases or making use of previously built diagrams (reusable artifacts) stored in the repository. These diagrams are availability in the repository to be used in this activity and present common functionalities and behaviours which are tried and tested solutions that help to improve the quality and validity of the diagrams to be built for the application to be developed. This artifact presents the overall diagram (or diagrams) of the secure mobile Grid application in which the "user-defined UC diagrams" and "reusable secure mobile Grid UC diagrams" artifacts are integrated, and it is also composed of the user-defined secure mobile grid UC artifact which defines new use cases that are not identified in the repository.

As we mentioned earlier, it is necessary to use a new UML profile to describe the use cases for mobile Grid environments. The following subsection shows a brief overview of this UML profile for secure mobile Grid use cases.

3.2.2.2. GridUCSec-profile. This UML-extension (defined in [56]) has been built as a UML profile which is an extensibility mechanism that allows us to adapt the metaclasses of a model so that the incorporation of new elements in a domain is possible. For the representation of the Grid use cases and security use cases, a set of stereotypes have been defined, which have been grouped by packages, *GridUCSec* and *TypesGridUCSec*, which are part of GridUCSec-profile (see Fig. 4).

The *GridUCSec* package is composed of Grid use cases, mobile use cases, security use cases, Grid security use cases, misuse cases, associations of permit, protect, threaten and mitigate, together with the involved actors. This package has 12 stereotypes: five specialize the UseCase (GridUC, SecurityUC, GridSecurityUC, MisuseCase and MobileUC), two specialize the Actor (GridActor and MisActor), and five specialize the DirectedRelationship and NamedElement (Permit, Protect, Threaten, and Mitigate). Fig. 5 shows

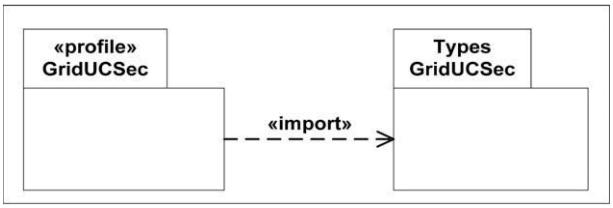

Fig. 4. Overview of the GridUCSec-profile.

the metamodel of the stereotypes defined in the *GridUCSec* package. The stereotypes of which this package is composed have been summarized in Table 1.

The *TypesGridUCSec* package defines the types of data for the tagged values of the stereotypes of the GridUCSec-profile as being the level of protection and the risk, types of permission, the requirement, the asset, the attack, etc., based on security standards and recommendations [29,30,49]. This package is composed of nine stereotypes which specialize the Enumeration class (AssetType, AttackType, AttackerType, CredentialType, FrequenceType, GridActorType, LevelType, PermissionType and RequirementType). Fig. 6 shows the stereotypes of which the TypesGridUCSec package is composed. These are the types of values necessary for the GridUCSec package.

Table 1 summarizes the main features of the stereotypes which are part of the *GridUCSec* profile. In this table we show, for each stereotype defined, the name of the stereotype, a brief textual description, the tagged values that enrich the semantics of the stereotype, and the graphical notation defined. More details of this UML profile (i.e. constraints) can be found in [56].

3.2.3. Tasks of the analysis activity

The analysis activity is composed of tasks which build uses case diagrams and specifications to obtain the analysis model in which the requirements are defined. This activity produces internal artefacts which are the output of some tasks and the input of others. All these internal artefacts are included in the analysis model to be used in the following activities if this is necessary. Fig. 7 shows a graphical representation of the analysis activity tasks using SPEM 2.0 icons.

Initially, in the *Defining UC of the application* task, we define the functional use cases of the application identified from the stakeholder needs and study the interactions with the user without considering the specific aspects of the mobile Grid environments.

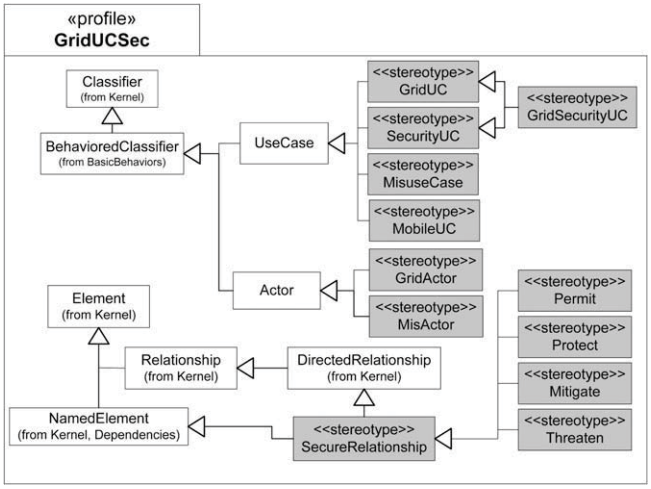

Fig. 5. Metamodel of GridUCSec-profile.

Table 1
Overview of stereotypes defined in GridUCSec-profile.

Stereotypes	Definition	Tagged values	Notation
GridUC	Identifies requirements of the Grid system and represents the common behaviour and relationships for this kind of systems. This specializes the UseCase within the Grid context defining the behaviour and functions for the Grid system.	GridRequirement, ProtectionLevel, SecurityDependence, InvolvedAsset	<<GridUC>>
SecurityUC	Identifies the system's security requirements, describing security tasks that the users will be able to perform with the system.	SecurityRequirement, InvolvedAsset, SecurityDegree, SecurityDomain	<<SecurityUC>>
GridSecurityUC	This represents specific security features of Grid systems. It adds specific special security features which are covered by this stereotype, and specializes to common security use cases of other applications, providing unique features for mobile Grid environments.	InvolvedAsset, SecurityRequirement, SecurityDegree, SecurityDependence, SecurityDomain	<<GridSecurityUC>>
MisuseCase	Identifies a sequence of actions, including variants, that a system or another entity can perform, interacting with misusers of the entity and causing harm to a stakeholder if the sequence is allowed to be completed [57,58].	InvolvedAsset, ImpactLevel, RiskLevel, ThreatLikelihood, KindAttack	<<Misuse>>
MobileUC	This represents mobile features of the mobile devices within Grid systems. It defines the mobile behaviour of the system and specializes the UseCase within the Grid context and Mobile Computing defining the behaviour and functions for the mobile Grid system.	MobileRequirement, ProtectionLevel, SecurityDependence, InvolvedAsset, NetworkProtocol, NameDomain	<<MobileUC>>
Protect	This relationship specifies that the behaviour of a use case may be protected by the behaviour of a security UC.	InvolvedAsset, ProtectionLevel, KindAttack	<<protect>>
Permit	This relationship specifies that the behaviour of a UC may be permitted by the behaviour of a security UC.	PermissionCondition, KindPermission	<<permit>>
Mitigate	This relationship specifies that the behaviour of a misuse case may be mitigated by the behaviour of a security UC.	SuccessPercentage, KindCountermeasure	<<mitigate>>
Threaten	This relationship specifies that the behaviour of a UC may be threatened by the behaviour of a misuse case.	SuccessPercentage, KindVulnerability, KindAttack	<<threaten>>
GridActor	This actor specifies a role played by a Grid user or any other Grid system that interacts with the system.	KindGridCredential, KindGridActor, KindRole, OrganizationName, Site-Credential	«GridActor»
MisActor	This actor specifies a role played by an attacker or misuser or any other attack that interacts with the system.	KindMisActor, HarmDegree	«MisActor»

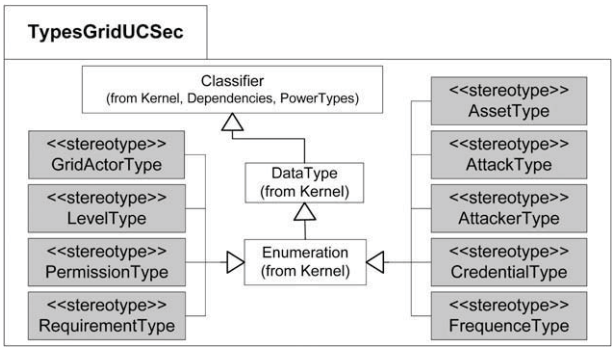

Fig. 6. Metamodel of TypesGridUCSec.

Next, in the *Identifying secure mobile Grid UC* task, we study the security aspects of the application within the mobile Grid context and identify the possible security use cases and misuse cases that can be reused of those defined in the repository, for the system in development. Once the use cases have been identified and defined, we build the overall use case diagram (or diagrams) in which we define the relationships between all the use cases and actors previously identified, and we describe the information from all the diagram's elements by following a new UML profile for mobile Grid use cases. We can also reuse and integrate some diagrams with common features of the repository which have been previously built for mobile Grid environments. This is carried out in the *Building secure mobile Grid UC diagram* task. In the *Supporting with UML models* task, we complete the analysis model with different UML models such as the sequence and collaboration diagrams according to use cases and scenarios, or class diagrams for an initial structural description of the system from the use cases diagrams built in previous tasks. In the "*Verifying Analysis model*" task, we must therefore verify that the artifacts have been correctly generated and the possible conflicts or errors in the analysis model have to be identified and analyzed for their subsequent refinements and corrections in the following iterations of this activity. Finally, the *Specifying Requirements* task consists of the formal definition of the requirements identified in previous tasks (functional requirements and non-functional requirements including security) from a template defined in the repository.

We shall now provide a detailed description of the activity that we have considered in our process using the SPEM 2.0 textual notation. We define the tasks, roles, steps, work products and guidance, which will be characterized according to the discipline that they belong to. According to SPEM, the Secure Mobile Grid System Analysis activity is described by using the structure shown in Fig. 8.

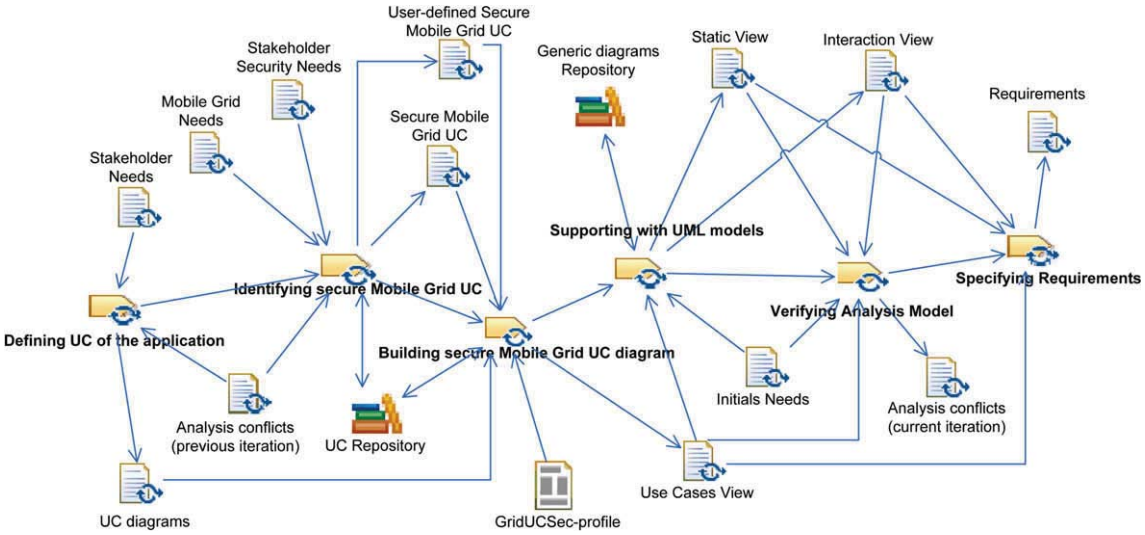

Fig. 7. Tasks and artifacts of Secure Mobile Grid System Analysis activity.

Each activity specifies *WorkProductUse* as both input and output respectively, the roles that perform or participate in this *RoleUse* activity, and the collection of *Steps* defined for a *Task Definition* which represents all the work that should be carried out to achieve the overall development goal of the Task Definition.

We shall now briefly define each task in this activity, indicating the steps that must be followed to successfully execute these tasks, by either seeking the assistance of a well-known UML-based development process for the common development tasks, or defining new techniques and steps that make our methodology specific to mobile Grid environments.

- *Defining UC of the application.* This task studies and defines the actors and use cases involved in the system but considers only the use cases that interact with the client apart from the mobile Grid environment. By beginning with the stakeholders needs, and following a particular development methodology such as the Unified Process [31], we can obtain a set of functional use cases and actors defined for the application. Fig. 9 shows the steps for this task using SPEM 2.0 textual notation.
- *Identifying secure mobile Grid UC.* In this task, a study of the system security must be carried out before identifying the security use cases and misuse cases of the repository. First, the assets that we wish to protect should be identified; second, the possible threats and attacks to these assets should be defined and, finally, the risk associated with these threats should be studied. The security use cases and misuse cases should then be defined, thus obtaining certain elements of the reusable repository such as the misuse cases for the system and the security use cases that mitigate them. Finally, a security assessment should be carried out. Some of the security use cases and misuse cases identified for the application are therefore stored in the repository and can be reused for this specific application since they are part of the secure mobile Grid UC output artefact. During this task, it is possible to discover new use cases which are suitable for incorporation into the repository, or we may wish to modify or update existing use cases in the repository. The repository is an input and output artefact from which we can obtain different elements and add or create new ones.

```
Activity {kind = Iteration}: Name of the iterative Activity
    TaskUse: Name of the Task
        ProcessPerformer {kind: primary}
            RoleUse: Role Name {kind: in}
        WorkDefinitionParameter {kind: in}
            WorkProductUse: Name
        WorkDefinitionParameter {kind: out}
            WorkProductUse: Name {state: state}
        Steps
            Step: Name of the step
        Guidance
            Guidance {kind: type}: Name
```

Fig. 8. Structure of the secure mobile Grid development using SPEM 2.0.

```
Activity {kind = Iteration}: First Secure Mobile Grid System Analysis

    TaskUse: Defining UC of the application
        ProcessPerformer {kind: primary}
            RoleUse: Client {kind: in}
            RoleUse: UC specifier {kind: in}
            RoleUse: Requirements engineer {kind: in}
            RoleUse: System Analyst {kind: in}
        WorkDefinitionParameter {kind: in}
            WorkProductUse: Stakeholder Needs
        WorkDefinitionParameter {kind: out}
            WorkProductUse: User-defined UC diagrams {state: initial draft}
        Steps
            Step: This task defines the use cases of the application using traditional
                  methods and mechanisms of requirements engineering as Unified
                  Process, etc.
        Guidance
            Guidance {kind: Practice}: Meetings
            Guidance {kind: Practice}: Interviews
```

Fig. 9. Detailed description of the task of defining UC of the application using SPEM 2.0.

Fig. 10 shows the steps for this task using SPEM 2.0 icons, and in Fig. 11 the SPEM 2.0 textual notation is used.

- *Building a secure mobile Grid UC diagram.* Once all the use cases (from the application and the repository) and the actors that take part in the system have been identified, the overall use case diagram is built. In the repository, not only the use cases and actors, but also the relationships between Grid use cases (UC, security UC, misuse case, mobile UC and actors) which can be reused for the system diagram are defined. We have defined a UML-extension (GridUCSec-profile) for secure mobile Grid use cases that helps to define the behaviour, attributes and relationships of this kind of mobile Grid systems. This UML profile will therefore be used and the new relationships will be defined for the overall diagram. This diagram, which is validated and analyzed, should have use cases, security UC, Grid security UC, mobile UC and misuse cases together with Grid actors and mis-

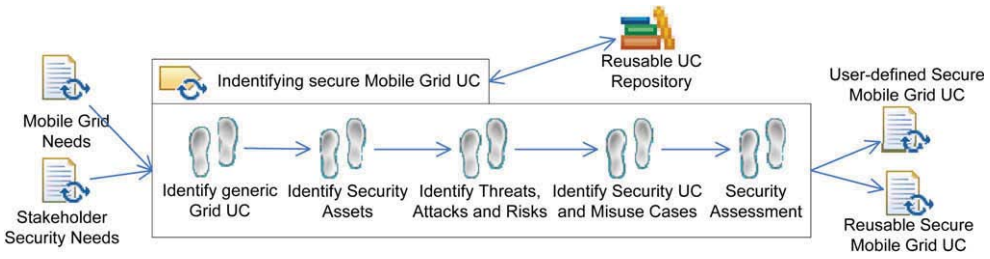

Fig. 10. Task of Identifying secure mobile Grid UC.

Activity {kind = **Iteration**}: First **Secure Mobile Grid System Analysis**

TaskUse: **Identifying secure mobile Grid UC**
- ProcessPerformer {kind: primary}
  - RoleUse: **Client** {kind: in}
  - RoleUse: **UC specifier** {kind: in}
  - RoleUse: **Security requirements engineer** {kind: in}
  - RoleUse: **System Security Analyst** {kind: in}
  - RoleUse: **Mobile Grid specialist** {kind: in}
- WorkDefinitionParameter {kind: in}
  - WorkProductUse: **Stakeholder security needs**
  - WorkProductUse: **mobile Grid needs**
  - WorkProductUse: **Repository of secure mobile Grid Use Cases**
- WorkDefinitionParameter {kind: out}
  - WorkProductUse: **Reusable Secure mobile Grid UC diagrams** {state: **initial draft**}
  - WorkProductUse: **User-defined Secure mobile Grid UC** {state: **initial draft**}
  - WorkProductUse: **Repository of secure mobile Grid Use Cases** {state: **reviewed**}
- Steps
  - Step: **Identify generic Grid UC** for the application
  - Step: **Identify security Assets** of the application in a mobile Grid environment
  - Step: **Identify Threats, Attacks and Risk** of the application in a mobile Grid environment
  - Step: **Identify the Security UC and Misuse cases** from the repository
  - Step: **Security Assessment**
- Guidance
  - Guidance {kind: **Checklist**}: **Catalogue of security assets to protect.**
  - Guidance {kind: **Checklist**}: **Catalogue of possible threats in the system.**
  - Guidance {kind: **Practice**}: **Well-defined misuse cases and security use cases for mobile Grid environments.**
  - Guidance {kind: **Practice**}: **Cost/effort-benefit vs risk analysis**
  - Guidance {kind: **Practice**}: **Security use cases**
  - Guidance {kind: **Practice**}: **Misuse cases**
  - Guidance {kind: **Practice**}: **Meetings**
  - Guidance {kind: **Practice**}: **Interviews**

Fig. 11. Detailed description of the task of identifying secure mobile Grid UC using SPEM 2.0.

actors. All these elements are defined by following the GridUC-Sec-profile, thus obtaining a complete diagram of secure mobile Grid use cases. The resulting view is the output of this task which is the input for the following task.

Fig. 12 shows the steps for this task using SPEM 2.0 icons, and the Fig. 13 the SPEM 2.0 textual notation is used.

- *Supporting with UML models.* A detailed description of use cases needs other models to complete the dynamic aspects that can-

Activity {kind = **Iteration**}: First **Secure Mobile Grid System Analysis**

TaskUse: **Building secure mobile Grid UC diagram**
- ProcessPerformer {kind: primary}
  - RoleUse: **UC specifier** {kind: in}
  - RoleUse: **Requirements engineer** {kind: in}
  - RoleUse: **Security requirements engineer** {kind: in}
  - RoleUse: **System analyst** {kind: in}
  - RoleUse: **Security analyst** {kind: in}
  - RoleUse: **Mobile Grid specialist** {kind: in}
- WorkDefinitionParameter {kind: in}
  - WorkProductUse: **User-defined secure mobile Grid UC**
  - WorkProductUse: **User-defined UC**
  - WorkProductUse: **Reusable secure mobile Grid UC diagrams**
  - WorkProductUse: **Repository of secure mobile Grid Use Cases**
- WorkDefinitionParameter {kind: out}
  - WorkProductUse: **Use Cases View** {state: **initial draft**}
  - WorkProductUse: **Repository of secure mobile Grid Use Cases** {state: **reviewed**}
- Steps
  - Step: **Integrate reusable use cases** identified of the repository **in the overall diagram** of use cases of the application
  - Step: **Define according to GridUCSec-profile** the elements of the diagram
  - Step: **Validate the overall diagram** of the application
- Guidance
  - Guidance {kind: **Template**}: **GridUCSec-profile**
  - Guidance {kind: **Practice**}: **Security use cases**
  - Guidance {kind: **Practice**}: **Misuse cases**
  - Guidance {kind: **Practice**}: **Meetings**

Fig. 13. Detailed description of the task of building secure mobile Grid UC diagram using SPEM 2.0.

not solely be described with use cases. These models are the sequence and collaboration diagrams which are related to use cases or use case scenarios and help to capture some aspects of the behaviour which are not captured with the definition of use cases. Many of these models are generically defined in the repository and are available to be instanced with specific elements associated with the use cases or scenarios. In the analysis stages it is also common to describe an initial structural view with class, subsystem, package, component etc. diagrams from the use cases identified in previous tasks, and by following traditional methods and techniques such as the Unified Process (UP). In this task, the stakeholders are free to select suitable techniques and methods with which to support the full definition of the use cases identified and defined with UML models,

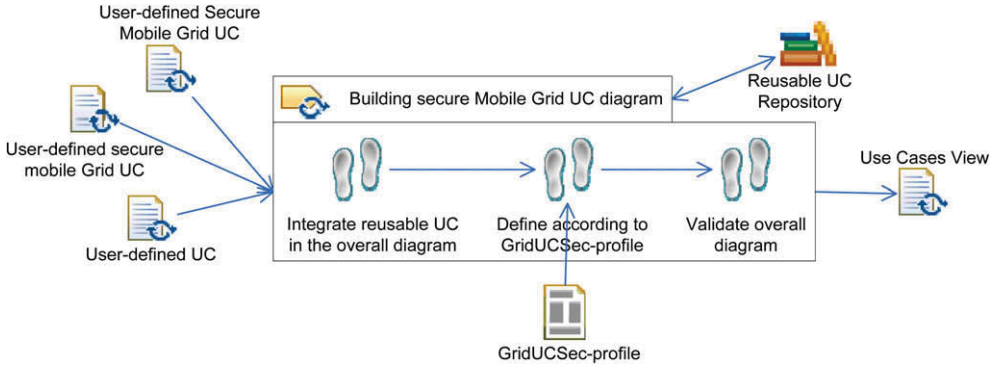

Fig. 12. Task of Building secure mobile Grid UC diagram of the application.

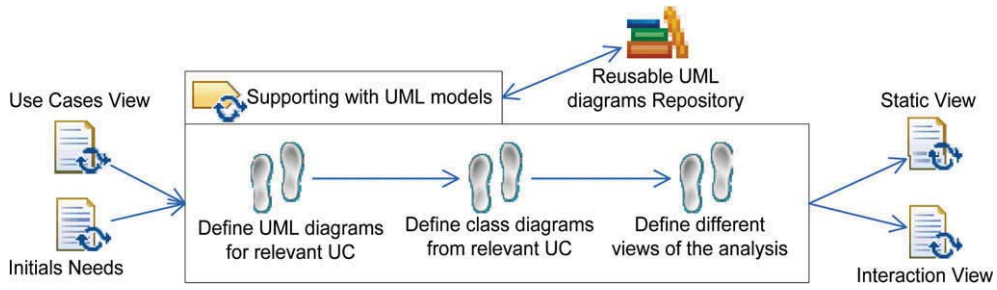

Fig. 14. Task of supporting with UML models.

and they are also free to the initial structural view (with the help of the UP, of use-case realizations and analysis classes, and so on). Several UML profiles with which to model the mobile aspects [12,22] and applications based on Grid services [10] exist which can be used in this analysis activity to model the static and dynamic behavior, completing the use cases with other UML diagrams oriented towards mobile systems and Grid environments such as activity diagrams, deployment, classes, interaction, etc., which capture these specific aspects of the mobile Grid systems. This structural view is a first vision of the architecture that will be built in the following design activity. The UML models defined in this task are output artefacts that form part of the analysis model.

Fig. 14 shows the steps for this task using SPEM 2.0 icons, and in Fig. 15 the SPEM 2.0 textual notation is used.

- *Verifying analysis model*. This task verifies whether the analysis artifacts have been correctly generated, and whether the different UML diagrams which have been defined to complement the use case view according to the UML profile are related and coordinated to correctly define and describe the behavior of the different scenarios identified from use cases diagrams with the purpose of identifying possible conflicts and problems that can be analyzed and corrected in order to improve all the artifacts in the following iterations of this activity before continuing with the design activity.

Activity {kind = **Iteration**}: First **Secure Mobile Grid System Analysis**

TaskUse: **Supporting with UML models**
ProcessPerformer {kind: primary}
RoleUse: **UC specifier** {kind: in}
RoleUse: **Requirements engineer** {kind: in}
RoleUse: **Security requirements engineer** {kind: in}
RoleUse: **Security analyst** {kind: in}
RoleUse: **Mobile Grid specialist** {kind: in}
RoleUse: **(Security) Architect** {kind: in}
WorkDefinitionParameter {kind: in}
WorkProductUse: **Use Cases view**
WorkProductUse: **Initials needs**
WorkProductUse: **Repository of reusable UML diagrams**
WorkDefinitionParameter {kind: out}
WorkProductUse: **Static View** {state: **initial draft**}
WorkProductUse: **Interaction View** {state: **initial draft**}
WorkProductUse: **Repository of reusable UML diagrams** {state: **reviewed**}
Steps
Step: **Define UML diagrams for the relevant secure mobile Grid use cases**
Step: **Define class diagrams from the relevant secure mobile Grid use cases** (for example, with help of analysis package, analysis class and use-case realizations in the Unified Process)
Step: **Define the structural view with UML diagrams (classes, components and packages) from secure mobile Grid uses cases**
Guidance
Guidance {kind: **Template**}: **Reusable Interaction diagrams**
Guidance {kind: **Template**}: **UMLSec**
Guidance {kind: **Guideline**}: **UML**
Guidance {kind: **Guideline**}: **Unified Process**

Fig. 15. Detailed description of the task of supporting with UML models using SPEM 2.0.

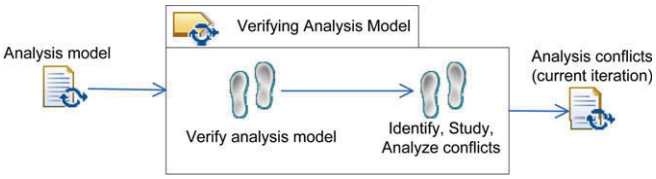

Fig. 16. Verifying Analysis model task.

Activity {kind = **Iteration**}: First **Secure Mobile Grid System Analysis**

TaskUse: **Verifying Analysis model**
ProcessPerformer {kind: primary}
RoleUse: **Requirements engineer** {kind: in}
RoleUse: **Security Requirements engineer** {kind: in}
RoleUse: **Security analyst** {kind: in}
RoleUse: **System analyst** {kind: in}
RoleUse: **Mobile Grid Specialist** {kind: in}
WorkDefinitionParameter {kind: in}
WorkProductUse: **Analysis model**
WorkDefinitionParameter {kind: out}
WorkProductUse: **Analysis conflicts** {state: **initial draft**}
Steps
Step: **Verify Analysis model**
Step: **Identify, Study and Analyze conflicts**
Guidance
Guidance {kind: **Practice**}: **Review Meetings**

Fig. 17. Detailed description of the task of Verifying Analysis model using SPEM 2.0.

Fig. 16 shows the steps for this task using SPEM 2.0 icons, and in Fig. 17 the SPEM 2.0 textual notation is used.

- *Specifying requirements*. In this task, we have sufficient information regarding 'what' the system does, and it is therefore possible to specify the requirements identified, defined and described which fulfil the initial needs. A specification of requirements together with the remaining artefacts generated in a task previous to this activity make up the analysis model that is the result of this activity and is the input artefact for the design activity. The requirements can be defined with generic templates based on the IEEE std. 1233, 12207.1, 830 standards [25,61] which are available in the repository.

Fig. 18 shows the steps for this task using SPEM 2.0 icons, and in Fig. 19 the SPEM 2.0 textual notation is used.

4. Case study

The GREDIA project [23] aims to develop a Grid application platform, providing high level support to the implementation of Grid business applications through a flexible graphical user interface. This platform will be generic in order to combine both existing and arising Grid middleware, and facilitate the provision of

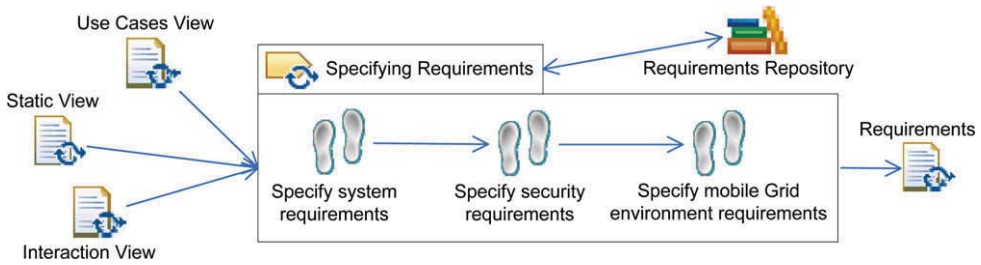

Fig. 18. Requirements specifying task.

Activity {kind = **Iteration**}: First **Secure Mobile Grid System Analysis**

TaskUse: **Specifying requirements**
ProcessPerformer {kind: primary}
RoleUse: **Requirements engineer** {kind: in}
RoleUse: **Security requirements engineer** {kind: in}
WorkDefinitionParameter {kind: in}
WorkProductUse: **Use Cases View**
WorkProductUse: **Static View**
WorkProductUse: **Interaction View**
WorkProductUse: **Requirements Repository**
WorkDefinitionParameter {kind: out}
WorkProductUse: **Requirements** {state: **initial draft**}
WorkProductUse: **Requirements Repository** {state: **reviewed**}
Steps
Step: **Specify system requirements**
Step: **Specify security requirements**
Step: **Specify mobile Grid environment requirements**
Guidance
Guidance {kind: **Template**}: **Template of specification of requirements**

Fig. 19. Detailed description of the requirements specifying task using SPEM 2.0.

business services, which mainly demand access to and the sharing of large quantities of distributed annotated numerical and multimedia content. Furthermore, GREDIA will make it easier for mobile devices to exploit Grid technologies in a seamless way by enabling mobile access to distributed annotated numerical and multimedia content. The potential effects of the platform will be validated through two pilot applications, servicing the vital sectors of media (news) and banking.

GREDIA is a system which aims to enable commercial users (such as those represented by the banking and media application pilots) to manipulate data and use services in a Grid Computing environment, thus leveraging the potential of computing Grids for business purposes as well as providing nontrivial business functionality solutions for end users in a controlled, secure environment. GREDIA will work on the specifications of a Security Framework to provide protection for data and transactions at all levels through a dedicated security framework that will be specifically developed for Grid based applications. The framework will address security issues in grid P2P architecture, such as the authentication of entities, confidentiality and integrity to enable the secure accessing of rich multimedia content.

Our development methodology is being applied to one of the pilot applications, the media (news) sector (see Fig. 20). The methodology is helping us to build a Mobile Grid application, which will allow journalists and photographers (media domain actors) to make their work available to a trusted network of peers at the same moment as it is produced, either from desktop or mobile devices.

With the explosion of ultra portable photo/video capture media (i.e. based on mobile phones, PDAs or solid state camcorders)

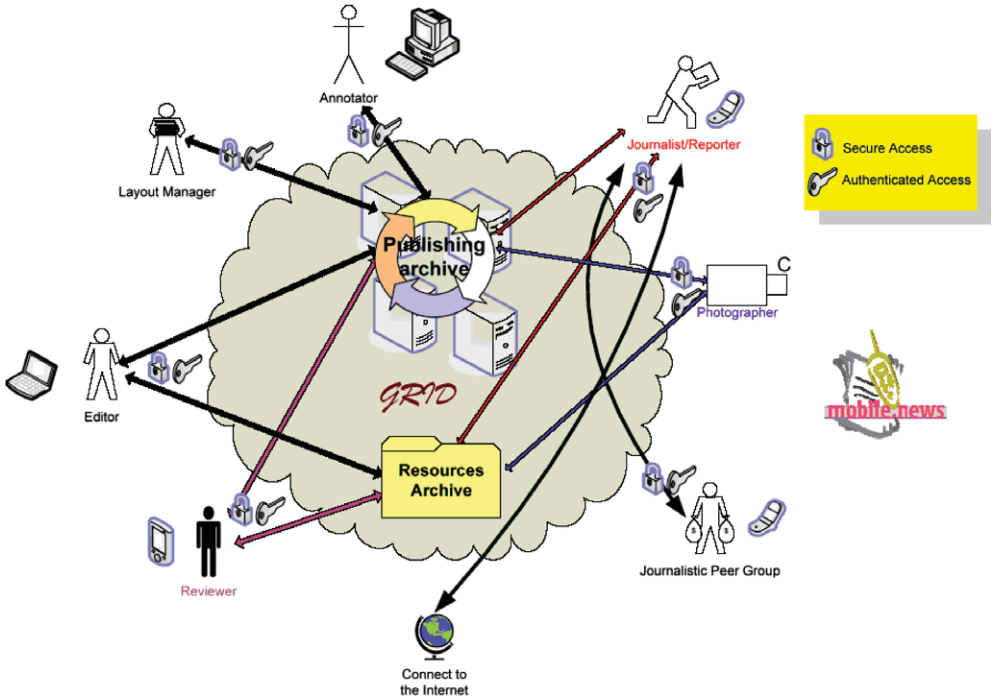

Fig. 20. Mobile Grid Computing system for media application.

Table 2
Use cases for media application.

Use case name	Login to the system
Goals/description	Provide authentication mechanisms
Scenario example	All users log into the grid system
Description	– A user launches the Grid application. – The user provides username and password. – The system checks the user data and permits or denies entry to the system.
	Search for news
Goals/description	A journalist can search for news material through the system interface in: 1. Public sources, 2. His organization's historical archive, 3. Trusted commercial portals according to the subscriptions paid-for.
Scenario example	The journalist familiarizes himself with the topic
Description	A user formulates a search query – The user selects sources to search from a list – The user submits the query

everyone can capture reasonably good quality audiovisual material while on the move. We wish 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 in which the device's battery dies or another malfunction destroys or makes his/her media unavailable.

We apply the analysis activity in this real case but consider only a reduced set of use cases owing to space constraints. The tasks in the analysis activity are shown below.

4.1. Defining UC of the application

We can define the use cases and actors for the application by applying a UML-based development process, such as Unified Process and OPEN, which guides us towards a definition and identification (through the typical techniques) to find the use cases and actors in an application. Once the use cases of the application have been defined and the use case diagram has been built, we can continue with the following task.

Of all the possible use cases defined for this application, we have considered two: login to the system and search for news (see Table 2).

4.2. Identifying secure mobile Grid UC

In this task, all the use cases and actors of the repository (mobile Grid UC, security UC, Grid security UC, Misuse cases, Grid (mis)actors) are identified by following the steps shown in Fig. 10.

By knowing the use cases of the application, we can identify and define the generic Grid use cases which are related to these use cases but are not within the initial stakeholder needs. A complete catalogue of generic Grid use cases can be found in the repository, and we can therefore select according to the use cases identified in the previous task. For example, we can associate the "Search News" use case with a generic Grid use case such as "<<GridUC>> Request" (defined in the repository) which is the use case responsible for managing the request inside the Grid.

First, we should identify the security assets involved in these use cases which are: Personal information about the journalist or editors: name, age, address, subscriptions, salaries; Media information used: photos, articles, recordings, videos, intellectual property rights; and exchange information: messages, queries, transactions.

Second, we should identify the threats which may attack these assets. In a first iteration, we identify several possible types of threats to information:

- Unauthorized access to grid system. In this scenario, the user wishes to login to the system, so we must ensure authorized access.
- Unauthorized disclosure and alteration of information. The user can send information to or receive information from the system. We must therefore protect the information which is both transmitted or stored. We must also protect the personal information that is transported through credentials.
- Unauthorized unavailability to resources. The user must have available resources at anytime and anywhere.

Once we have identified assets and threats, we can identify the necessary security UC and misuse cases which respectively protect and threaten the security assets. The misuse cases that define the bad behaviour of the threats are: Alteration of information which attacks the content of a message (integrity); Disclosure of information which attacks the confidentiality of a message from grid system to user; and Unauthorized access which attacks a user's authenticity and access privileges. The security UC associated with these misuse cases and security assets are: Ensure Integrity, Ensure Confidentiality, Authenticate and Authorize Access.

All these security UC and misuse cases are common to Grid environments and are therefore defined in the repository, and can be used in this activity to define these UC. An example of the definition of security UC and misuse cases defined in the repository but instanced for this real case is shown in Table 3.

Finally, it is necessary to assess whether the threats are relevant according to the security level specified by the security objectives. We must therefore estimate the security risks based on the relevant threats, their likelihood and their potential negative impacts, in other words, we have to estimate the impact (what may happen) and risk (what will probably happen) to which the assets in the system are exposed. We must therefore interpret the meaning of impact and risk. In Table 4 we define the impact and risk for two of these threats.

4.3. Building secure mobile Grid UC diagram

Having identified initial use cases of the application and of the repository, we can now build the use case diagram by following the steps described in Fig. 12, and we can establish relationships between them by following the UML-extension, GridUCSec-profile, thus defining the new relationships, tagged values and constraints defined in this profile. Previously built diagrams that can be reused and adapted to the diagram that we are building also exist in the repository. These reusable diagrams define the common behaviour

Table 3
Example of one misuse case and one Grid security use case.

Misuse case	Alteration of information	
Attack	Attack on the content of a message (integrity).	
Summary	The external attacker type gains access to the message exchanged between the journalist and the Grid system, and modifies the part of the message that contains the media information with the intention of changing its meaning by modifying some aspect of the information such as authors, dates, or secrecy information.	
Preconditions		
1	The external attacker has physical access to the message.	
2	The external attacker has a clear knowledge of where the secrecy information is located within the message.	
Interactions		
1	User interactions	The journalist sends a query message to obtain media information
2	Misuser interactions	The external attacker intercepts it and identifies the part of the message, modifies the media information and forwards it onto media Grid
3	System interactions	Media Grid receives the corrupted message and processes it incorrectly due to its altered semantic content. That is, it establishes that the journalist wishes to receive new media information which is in fact the media information that has been modified by the attacker.
Postconditions		
1	Media Grid will remain in a state of error with regard to the journalist's original intentions.	
2	In the system register in which the media grid was executed, the request received with an altered semantic content will be reflected.	
Grid security use case	Ensure integrity	
Use case path	System message integrity	
Security threat	A misuser corrupts a message from the Grid system to a user.	
Preconditions		
1	The misuser has the means to intercept a message from the Grid system to a user.	
2	The misuser has the means to modify an intercepted message.	
3	The misuser has the means to forward the modified message to the user.	
Interactions		
1	System interactions	The Grid system sends a message to a user.
	System actions	The Grid system ensures that modifications to the message will be obvious to the user
2	Misuser interactions	The misuser intercepts and modifies the Grid system's message and forwards it onto the user.
3	User interactions	The user receives the corrupted message.
	System actions	The Grid system will recognize that the message was corrupted.
4	System interactions	The Grid system will notify the user that the message was corrupted
Postconditions	None	

and interactions found in any mobile Grid environments, so it is possible reuse the same behaviour for this application.

Fig. 21 shows a first resulting diagram (for a first iteration) for the use cases identified, including the security use case and repository misuse case diagram. We must define the news relationships (protect, mitigate, permit and threaten) in order to relating them with the use cases of the application. Finally, we should validate this diagram by checking whether the relationships between the use cases are well defined and that there are no redundancies or faults. This can be done in a first iteration or can be refined in successive iterations of this task by adding news use cases and relationships.

This diagram is completed with a detailed description of relationships, constraints and tagged values which are defined in the new UML-extension, a GridUCSec-profile that defines different attributes and security properties for use cases that should be applied to this real case. Using the secure mobile Grid use case diagram from the previous task as a starting point, we must describe all the use cases, actors, tagged values and relationships identified in the diagram, thus obtaining a detailed description which, together with the diagram, make up the secure mobile Grid use case diagram. Table 5 shows the use cases, actors and relationships from Fig. 21 and also adds important information about all

Table 4
Assessment of threats, attacks and risks.

Threat	Unauthorised alteration of information	
Impact	LOW if there is no personal information modified	HIGH if the opposite is the case
Attack	Modification of information	
Probability	Frequent	Frequent
Risk	LOW	HIGH
	Unauthorised disclosure of information	
Impact	LOW when the disclosed information is not sensitive or important	HIGH if the opposite is the case
Attack	Interception of information	
Probability	Frequent	Very frequent
Risk	LOW	HIGH

these elements of the diagram such as the tagged values defined in GridUCSec-profile, assigning specific values to each one of them.

The table is built with two purposes: (a) to have the use case diagram built for the application in text format and to incorporate useful information about the use cases, actors, relationships, constraints and features of the environment together with security aspects. This text format used is that of the CASE tool that we want to build in order to save and open the tool's graphical diagram; (b) the table will be used in the following activity (the design activity) in which the relevant information, which may be tagged values, is extracted in order to make decisions and to select security mechanisms, methods, policies and security services when building the application's security architecture.

This table is used to define all possible security information from the use case diagrams in which the security experts are involved, and can be treated with automatic or semiautomatic tools to make transformations or represent the information in graphical notation. This is also passed to the design activity to extract the necessary information. This table, together with the diagrams built and the use cases identified in the previous tasks in this activity, define and describe the use case view.

4.4. Supporting with UML models

In this task UML models are used as interaction diagrams to complete the capture of requirements and their dynamic behaviour. These models can be used to define the actions between actors and use cases, and the flow of events produced between the elements of the diagram. The purpose of these UML models is to complete the definition of use cases in order to obtain better knowledge with regard to the system's behaviour and all the elements involved in refining the use case model with the new aspects identified thanks to these UML models.

UML statechart diagrams can be used to describe the states of the use cases and the transitions between those states. Activity diagrams can be used to describe the transitions between states in more detail as a sequence of actions. Interaction diagrams can be used to describe how an instance of a use case interacts with an instance of an actor. The interaction diagram then shows the use case and the participating actor (or actors). In the repository we can define generic interaction diagrams which are related to reusable use cases and these diagrams can then be instanced for the application's use cases. For example, in Fig. 22 we can see a generic sequence diagram associated with a Grid security use case (GridSecurityUC) called "*Ensure Integrity*".

By obtaining the "⟨⟨*GridSecurityUC*⟩⟩ *Ensure Integrity*" of the reusable repository, we can also obtain its associated generic sequence diagram and this can be instanced with the application's use cases and actors. Fig. 23 shows the sequence diagram associated with Ensure Integrity and Search News with the actors and messages specific to the application.

In this task, we can follow, for example, the Unified Process to define the class diagrams from the realization of the use case diagrams, and initially describe the package and development diagrams which will be used and refined in the design activity.

4.5. Verifying analysis model

Once the artifacts have been defined, with the exception of the "*Requirements*" artifact, we must verify that they have been correctly generated, that is, that the UML diagrams, such as for example the sequence diagram for message integrity, define the correct elements involved in the use case or in a scenario of use cases, in this case, the "⟨⟨*GridSecurityUC*⟩⟩ *Ensure Integrity*" use case. This verification should also be made with the remaining diagrams and guides by the use cases defined in this activity. We can check

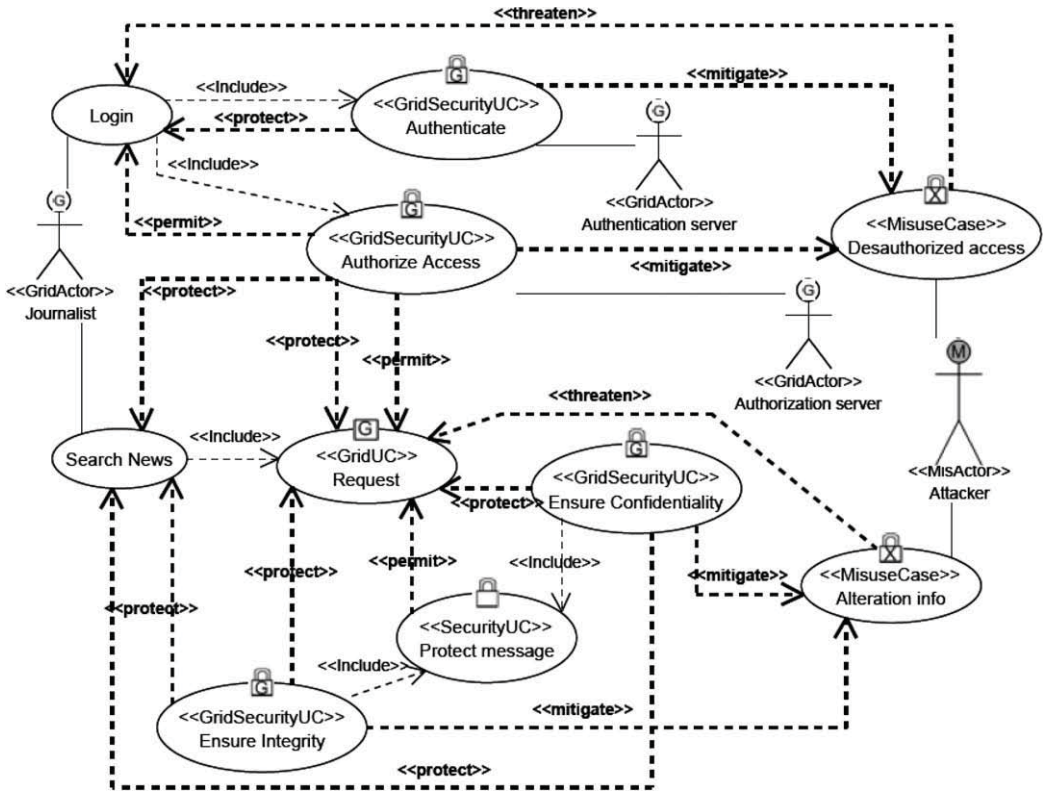

Fig. 21. Diagram of secure mobile Grid use cases for media application.

Table 5
Application of GridUCSec-profile to a case study.

Element	Stereotype	Associations (Assoc) and tagged values (TagV)			
Authenticate	⟨⟨GridSecurityUC⟩⟩	Assoc	mitigate: Mitigate	Assoc	mitigation: Authenticate mitigatedCase: Desauthorized access
				TagV	SuccessPercentage: {VHigh} KindCountermeasure: Trust Authorities
			protect: Protect	Assoc	protection: Authenticate protectedCase: Login
				TagV	InvolvedAsset: {Identity} ProtectionLevel: {High} KindAttack: {MaliciousAtt}
		TagV	InvolvedAsset: {Identity} Securityrequirement: {Authentication} SecurityDegree: {High} SecurityDependence: [65]		
Authorize Access	⟨⟨GridSecurityUC⟩⟩	Assoc	mitigate: Mitigate	Assoc	mitigation: Authorize Access mitigatedCase: Desauthorized access
				TagV	SuccessPercentage: {VHigh} KindCountermeasure: Access control
			protect: Protect	Assoc	protection: Authorize Access protectedCase: Search News
				TagV	InvolvedAsset: {Rights} ProtectionLevel: {High} KindAttack: {MaliciousAtt}
			protect: Protect	Assoc	protection: Authorize Access protectedCase: Request
				TagV	InvolvedAsset: {Rights} ProtectionLevel: {Medium} KindAttack: {AccessControlAtt, MaliciousAtt}
			permit: Permit	Assoc	permittingCase: Authorize Access permittedCase: Login
				TagV	PermissionCondition: user has access rights KindPermission: Execute
			permit: Permit	Assoc	permittingCase: Authorize Access permittedCase: Request
				TagV	PermissionCondition: valid user rights KindPermission: Execute
		TagV	InvolvedAsset: {Message} Securityrequirement: {Authorization&AC} SecurityDegree: {High} SecurityDependence: {Medium}		
Ensure Confidentiality	⟨⟨GridSecurityUC⟩⟩	Assoc	mitigate: Mitigate	Assoc	mitigation: Ensure Confidentiality mitigatedCase: Alteration info
				TagV	SuccessPercentage: [65] KindCountermeasure: encrypt message
			protect: Protect	Assoc	protection: Ensure Confidentiality protectedCase: Search News
				TagV	InvolvedAsset: {Message} ProtectionLevel: {High} KindAttack: {MasqueradingAtt, EavesdroppingAtt}
			protect: Protect	Assoc	protection: Ensure Confidentiality protectedCase: Request
				TagV	InvolvedAsset: {Message} ProtectionLevel: {High} KindAttack: {MasqueradingAtt}
		TagV	InvolvedAsset: {Message} Securityrequirement: {Confidentiality} SecurityDegree: {High} SecurityDependence: {VLow}		
Ensure Integrity	⟨⟨GridSecurityUC⟩⟩	Assoc	mitigate: Mitigate	Assoc	mitigation: Ensure Integrity mitigatedCase: Alteration info
				TagV	SuccessPercentage: {Medium} KindCountermeasure: signed messages
			protect: Protect	Assoc	protection: Ensure Integrity protectedCase: Search News
				TagV	InvolvedAsset: {Message} ProtectionLevel: {High} KindAttack: {SniffingAtt}
			protect: Protect	Assoc	protection: Ensure Confidentiality protectedCase: Request
				TagV	InvolvedAsset: {Message} ProtectionLevel: {VHigh} KindAttack: {EavesdroppingAtt}

Table 5 (*continued*)

Element	Stereotype	Associations (Assoc) and tagged values (TagV)			
		TagV	InvolvedAsset: {Message} Securityrequirement: {Integrity} SecurityDegree: {VHigh} SecurityDependence: {VLow}		
Protect Message	⟨⟨SecurityUC⟩⟩	Assoc	permit: Permit	Assoc	permittingCase: Protect Message permittedCase: Request
				TagV	PermissionCondition: all messages encrypted KindPermission: Execute
		TagV	SecurityRequirement: {Confidentiality, Integrity} InvolvedAsset: {Message} SecurityDegree: {High}		
Request	⟨⟨GridUC⟩⟩	Assoc	isPermitting: Permit (defined in Protect message) isPermitting: Permit (defined in Authorize Access) isProtecting: Protect (defined in Authorize Access) isProtecting: Protect (defined in Ensure Confidentiality) isProtecting: Protect (defined in Ensure Integrity) isThreating: Threaten (defined in Alteration info)		
		TagV	GridRequirement: {Interoperatibility, Availability} ProtectionLevel: {Medium} SecurityDependence: {High} InvolvedAsset: {Message}		
Desauthorized access	⟨⟨MisuseCase⟩⟩	Assoc	threaten: Threaten	Assoc	threateningCase: Desauthorized access threatenedCase: Login
				TagV	SuccessPercentage: {VHigh} KindVulnerability: User and password KindAttack: {AccessControlAtt, MaliciousAtt}
			isMitigating: Mitigate (defined in Authenticate) isMitigating: Mitigate (defined in Authorize Access)		
		TagV	InvolvedAsset: {Identity, User} ImpactLevel: {VHigh} RiskLevel: {VHigh} ThreatLikelihood: {Frequent}		
Alteration info	⟨⟨MisuseCase⟩⟩	Assoc	threaten: Threaten	Assoc	threateningCase: Alteration info threatenedCase: Request
				TagV	SuccessPercentage: {High} KindVulnerability: messages by distributed network KindAttack: {EavesdroppingAtt, MasqueradingAtt}
			isMitigating: Mitigate (defined in Ensure Confidentiality) isMitigating: Mitigate (defined in Ensure Integrity)		
		TagV	InvolvedAsset: {Message, Identity, Data} ImpactLevel: {VHigh} RiskLevel: {VHigh} ThreatLikelihood: {Frequent}		
Login	⟨⟨UseCase⟩⟩	Assoc	isTheatening: Threat (defined in Desauthorized access) isProtecting: Protect (defined in Authenticate) isPermitting: Permit (defined in Authorize Access)		
Search news	⟨⟨UseCase⟩⟩	Assoc	isprotecting: Protect (defined in Authorize Access) isProtecting: Protect (defined in Ensure Integrity) isProtecting: Protect (defined in Ensure Confidentiality)		
Journalist	⟨⟨GridActor⟩⟩	Assoc	UseCase: Login Use Case: Search News		
		TagV	KindGridActor: {Mobile User}		

(*continued on next page*)

Table 5 (*continued*)

Element	Stereotype	Associations (Assoc) and tagged values (TagV)	
			KindRole: employee OrganizationName: News KindGridCredential: {UserPass}
Authentication Server	⟨⟨GridActor⟩⟩	Assoc	GridSecurityUC: Authenticate
		TagV	KindGridActor: {Service} KindRole: Grid server OrganizationName: VO KindGridCredential: {UserPass, X509, Kerberos}
Authorization server	⟨⟨GridActor⟩⟩	Assoc	GridSecurityUC: Authorize Access
		TagV	KindGridActor: {Service} KindRole: Grid server OrganizationName: VO KindGridCredential: {UserPass, X509, Kerberos}
Attacker	⟨⟨MisActor⟩⟩	Assoc	MisuseCase: Desauthorized access MisuseCase: Alteration info
		TagV	KindMisActor: cracker HarmDegree: {VHigh}

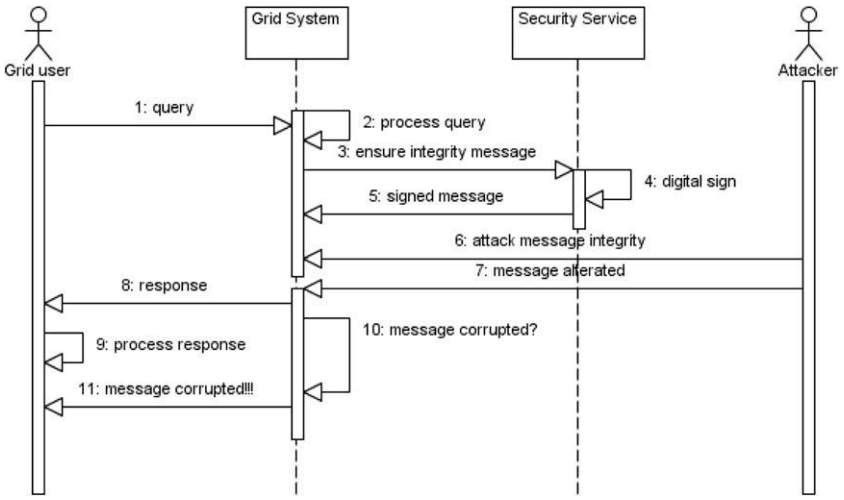

Fig. 22. Template of sequence diagram for the message integrity.

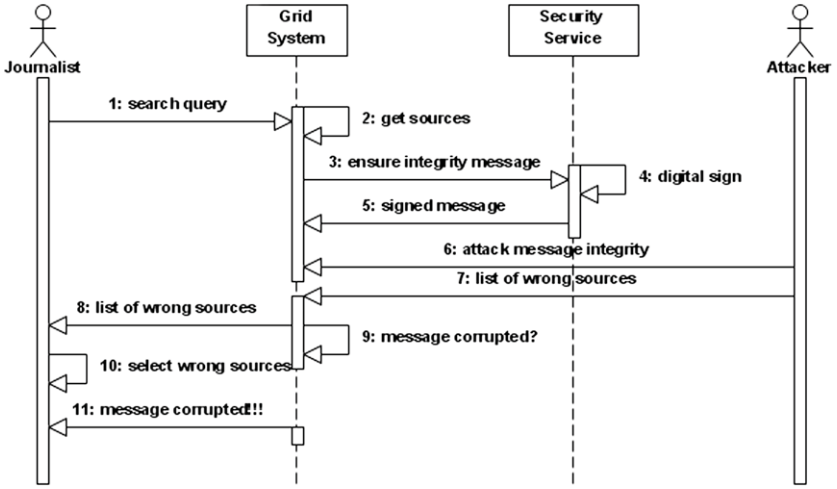

Fig. 23. Sequence diagram for message integrity associated with "Search News" use case.

that any error has occurred, so the "*Analysis conflicts*" artifact is not defined in this iteration.

4.6. Specifying requirements

This is the final analysis activity task and it specifies the set of requirements identified during all the previous tasks, thus obtaining a formal description of the system's functional and non-functional requirements. This description should indicate all the elements involved in the definition of the requirements together with the interactions between use cases and actors and the attributes of these elements. All this information has been generated in previous tasks through a use case model using the GridUCSec-profile and with UML models. Templates for the definition of requirements exist in the repository in which the main characteristics of the models generated in this activity are summarized and formally described in a document which is part of the analysis model.

5. Conclusions

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 those of Grid Computing which are not developed through methodologies adapted to their most differentiating features. 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.

The complexity of current applications forces us to plan and follow an action plan to control the whole software life-cycle as well as to ensure that decisions are made in a controlled manner. A systematic process is essential to build quality software, offering methods, techniques and tools that facilitate the work of the entire team involved in software development. In order to build a secure Grid system, we have defined a methodology which, apart from developing a Grid system, allows us to incorporate all Grid security aspects into the life-cycle thus obtaining a secure end product.

In this paper, we have presented a systematic development for secure mobile Grid environments and we have defined the analysis activity, which has been managed by reusable use cases and which facilitates the specification of both the system and the security requirements of our application. In this methodology we can follow, in some cases, the typical development processes, such as the Unified Process, and in others cases, we must use the new associated tasks, techniques, guidelines and practices to build Secure Mobile Grid Systems. This methodology should be compatible with any typical development process and techniques (i.e. UMLSec) but helping us of artifacts, templates, definitions, diagrams, etc., defined in a reusable repository which is essential to develop and build a system under a secure mobile Grid environment, and we must also integrate security aspects from the first phases of the methodology. Therefore, both security and the mobile Grid environment are present in all the activities of the methodology, and all the differentiating features of these systems are taken into account from the beginning.

The methodology proposed in this paper helps to develop Grid Computing based systems by using reusable, tried and tested elements that make it easier for stakeholders to analyze, design and construct a Secure Mobile Grid System and improve the quality of these systems for subsequent projects. The GREDIA case study has allowed us to improve many aspects of the methodology, such as for example, the definition of the UML profile by identifying relationships, tagged values and stereotypes to cover certain aspects that we have gone finding to apply the methodology to the case study, and which had not been considered in earlier versions of the profile.

As future work, we intend to define the design and construction activities of this methodology through the research–action method, by integrating security requirements engineering techniques (UMLSec, etc.) and defining the traceability of artefacts and starting from use cases, identifying services within the architecture in order to arrive at any implementation platform (i.e. Globus) through automatic transformations or MDA. We shall also define the catalogue of artefacts and elements of the repository which can be reused in our methodology.

Acknowledgments

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). Special acknowledgment to GREDIA (FP6 34363 – Grid enabled access to rich media content) funded by European Commission.

References

[1] C. Artelsmair, R. Wagner, Towards a security engineering process, in: The 7th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, Florida, USA, 2003.
[2] D. Basin, J. Doser, SecureUML: a UML-based modeling language for model-driven security, in: 5th International Conference on the Unified Modeling Language, Lecture Notes in Computer Science 2460, 2002.
[3] D. Basin, J. Doser, T. Lodderstedt, Model driven security for process-oriented systems, in: ACM Symposium on Access Control Models and Technologies, ACM Press, Como, Italy, 2003.
[4] L. Bass, F. Bachmann, R.J. Ellison, A.P. Moore, M. Klein, Security and survivability reasoning frameworks and architectural design tactics, SEI (2004).
[5] S. Bhanwar, S. Bawa, Securing a Grid, in World Academy of Science, Engineering and Technology, 2008.
[6] P.G. Bradford, B.M. Grizzell, G.T. Jay, J.T. Jenkins, Cap. 4. Pragmatic Security for Constrained Wireless Networks, in Security in Distributed, Grid, Mobile, and Pervasive Computing, A. Publications, Editor, The University of Alabama, Tuscaloosa, USA, 2007, p. 440.
[7] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, A. Perin, TROPOS: an agent-oriented software development methodology, Journal of Autonomous Agents and Multi-Agent Systems 8 (3) (2004) 203–236.
[8] R. Breu, K. Burger, M. Hafner, J. Jürjens, G. Popp, V. Lotz, 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.
[9] J. Castro, M. Kolp, J. Mylopoulos, A requirements-driven development methodology, in: 13th Int. Conf. on Advanced Information Systems Engineering, CAiSE'01, 2001.
[10] A. D'Ambrogio, L. Conticelli, A UML profile for modeling software applications based on grid services, in: IASTED International Conference on Software Engineering, ACTA Press, 2008.
[11] H. Dail, O. Sievert, F. Berman, H. Casanova, A. YarKhan, S. Vadhiyar, J. Dongarra, C. Liu, L. Yang, D. Angulo, 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.
[12] V. Dehlen, J.Ø. Aagedal, A UML profile for modeling mobile information systems, in: Distributed Applications and Interoperable Systems – DAIS'07 (LNCS 4531), 2007.
[13] C. Estay, J. Pastor, Towards the project-based action–research for information systems, in: 10th Annual Business and Information Technology Conference (BIT'2000), Manchester, United Kingdom, 2000.
[14] E. Fernández-Medina, J. Jurjens, J. Trujillo, S. Jajodia, Model-driven development for secure information systems, Information and Software Technology 51 (5) (2009) 809–814.
[15] E. Fernández-Medina, M. Piattini, Designing secure databases, Information and Software Technology 47 (7) (2005) 463–477.
[16] I. Foster, C. Kesselman, J.M. Nick, S. Tuecke, The physiology of the grid: an open grid services architecture for distributed systems integration, Open Grid Service Infrastructure WG, Global Grid Forum, 2002.
[17] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, A security architecture for computational grids, in: 5th Conference on Computer and Communications Security, ACM Press, San Francisco, USA, 1998.

[18] I. Foster, C. Kesselman, S. Tuecke, The anatomy of the grid: enabling scalable virtual organizations, in: 7th International Euro-Par Conference Manchester on Parallel Processing, Springer-Verlag, 2001.
[19] G. Georg, I. Ray, K. Anastasakis, B. Bordbar, M. Toahchoodee, S.H. Houmb, An aspect-oriented methodology for designing secure applications, Information and Software Technology 51 (5) (2009) 846–864.
[20] Globus Project. <http://globus.org/>.
[21] D. Graham, Introduction to the CLASP Process, 2006.
[22] V. Grassi, R. Mirandola, A. Sabetta, A UML profile to model mobile systems, in: UML 2004, LNCS 3273, 2004, p. 128–142.
[23] GREDIA project. <www.gredia.eu>.
[24] T. Guan, E. Zaluska, D.D. Roure, A grid service infrastructure for mobile devices, in: First International Conference on Semantics, Knowledge, an Grid (SKG 2005), Beijing, China, 2005.
[25] C. Gutiérrez, E. Fernández-Medina, M. Piattini, Security requirements for web services based on SIREN, in: Symposium on Requirements Engineering for Information Security, Paris, France, 2005.
[26] C.B. Haley, J.D. Moffet, R. Laney, B. Nuseibeh, A framework for security requirements engineering, in: Software Engineering for Secure Systems Workshop, Shangai, China, 2006.
[27] M. Humphrey, M.R. Thompson, K.R. Jackson, Security for grids, Lawrence Berkeley National Laboratory, Paper LBNL-54853, 2005.
[28] ISO/IEC 25010, Quality model (ISO/IEC 9126-1), 2009.
[29] ISO/IEC, Information technology – guidelines for the management of IT security – Part 1: concepts and models for IT Security, 1996.
[30] ITU, ITU_T Recommendation X.800. Security Architecture for Open Systems Interconnection for CCITT Applications, 1991.
[31] I. Jacobson, G. Booch, J. Rumbaugh, The Unified Software Development Process. Addison-Wesley Professional, vol. 512,1999.
[32] H. Jameel, U. Kalim, A. Sajjad, S. Lee, T. Jeon, Mobile-to-grid middleware: bridging the gap between mobile and grid environments, in: European Grid Conference EGC 2005, Springer, Amsterdam, The Netherlands, 2005.
[33] J. Jürjens, Towards secure systems development with UMLsec in international conference of fundamental approaches to software engineering (FASE/ETAPS), Springer-Verlag, Genoa, Italy, 2001.
[34] J. Jürjens, Secure Systems Development with UML, Springer, 2004.
[35] J. Jürjens, J. Schreck, P. Bartmann, Model-based security analysis for mobile communications, in: International Conference on Software Engineering, IEEE Computer Society, Leipzig, Germany, 2008.
[36] J. Jurjens, UMLsec: extending UML for secure systems development, in: 5th International Conference on the Unified Modeling Language (UML), Dresden, Germany, 2002.
[37] J. Jurjens, Using UMLsec and goal trees for secure systems development, Communications of the ACM 48 (5) (2002) 1026–1030.
[38] R. Kolonay, M. Sobolewski, Grid interactive service-oriented programming environment, in: Concurrent Engineering: The Worldwide Engineering Grid, Press and Springer Verlag, Tsinghua, China, 2004.
[39] G. Kostopoulos, N. Sklavos, O. Koufopavlou, Cap. 10. State-of-the-Art Security in Grid Computing, in Security in Distributed, Grid, Mobile, and Pervasive Computing, A. Publications, Editor, The University of Alabama, Tuscaloosa, USA, 2007, p. 440.
[40] P. Kruchten, The Rational Unified Process: An Introduction, 2nd ed., Addison-Wesley, 2000. p. 320.
[41] A. Litke, D. Skoutas, T. Varvarigou, Mobile grid computing: changes and challenges of resourse management in a mobile grid environment, in: 5th International Conference on Practical Aspects of Knowledge Management (PAKM 2004), 2004.
[42] T. Lodderstedt, D. Basin, J.R. Doser, SecureUML: A UML-Based Modeling Language for Model-Driven Security, Springer, Dresden, Germany, 2002.
[43] N. Mead, Identifying security requirements using the SQUARE method, Integrating Security and Software Engineering: Advances and Future Visions (2006) 44–69.
[44] D. Mellado, E. Fernández-Medina, M. Piattini, A common criteria based security requirements engineering process for the development of secure information systems, Computer Standards & Interfaces 29 (2) (2007) 244–253.
[45] D. Mellado, E. Fernández-Medina, M. Piattini, Security requirements engineering process for software product lines: a case study, in: The Third International Conference on Software Engineering Advances, IEEE Computer Society, Sliema, Malta, 2008.
[46] H. Mouratidis, A security oriented approach in the development of multiagent systems: applied to the management of the health and social are needs of older people in England, University of Sheffield, 2004.
[47] H. Mouratidis, P. Giorgini, Integrating security and software engineering: advances and future vision, IGI Global, 2006.
[48] OMG, Software & Systems Process Engineering Meta-Model Specification (SPEM) 2.0, 2008.
[49] Open Grid Forum, The open grid services architecture, Version 1.5, 2006.
[50] G. Popp, J. Jürjens, G. Wimmel, R. Breu, Security-critical system development with extended use cases, in: Tenth Asia-Pacific Software Engineering Conference (APSEC'03), IEEE, 2003.
[51] D.G. Rosado, E. Fernández-Medina, 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, LNCS 5833, Gramado, Brasil, 2009.
[52] D.G. Rosado, E. Fernández-Medina, J. López, Obtaining security requirements for a mobile grid system, International Journal of Grid and High Performance Computing 1 (3) (2009) 1–17.
[53] D.G. Rosado, E. Fernández-Medina, 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, Vancouver, Canada, 2009.
[54] D.G. Rosado, E. Fernández-Medina, J. López, 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, Porto, Portugal, 2008.
[55] D.G. Rosado, E. Fernández-Medina, J. López, M. Piattini, PSecGCM: process for the development of secure grid computing based systems with mobile devices, in: International Conference on Availability Reliability and Security (ARES 2008), IEEE Computer Society, Barcelona, Spain, 2008.
[56] D.G. Rosado, E. Fernández-Medina, J. López, M. Piattini, Towards an UML extension of reusable secure use cases for mobile grid systems, IEICE Transactions on Information and Systems, submitted for publication.
[57] L. Røstad, An extended misuse case notation: including vulnerabilities and the insider threat, in: XII Working Conference on Requirements Engineering: Foundation for Software Quality, Luxembourg, 2006.
[58] G. Sindre, A.L. Opdahl, Capturing security requirements by misuse cases, in: 14th Norwegian Informatics Conference (NIK'2001), Tromsø, Norway, 2001.
[59] C. Steel, R. Nagappan, R. Lai, 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. (Chapter 8).
[60] A. Talukder, R. Yavagal, Security issues in mobile computing, in: Mobile Computing, McGraw-Hill Professional, 2006 (Chapter 18).
[61] A. Toval, J. Nicolás, B. Moros, F. García, Requirements reuse for improving information systems security: a practitioner's approach, Requirements Engineering Journal 6 (2002) 205–219.
[62] J. Trujillo, E. Soler, E. Fernández-Medina, M. Piattini, An engineering process for developing secure data warehouses, Information and Software Technology 51 (6) (2009) 1033–1051.
[63] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, S. Tuecke, Security for grid services, in: 12th IEEE International Symposium on High Performance Distributed Computing (HPDC-12 '03), IEEE Computer Society, 2003.
[64] B.D. Win, R. Scandariato, K. Buyens, J. Grégoire, W. Joosen, On the secure software development process: CLASP. SDL and Touchpoints compared, Information and Software Technology 51 (7) (2009) 1152–1171.
[65] J. Yoder, J. Barcalow, Architectural patterns for enabling application security, in: 4th Conference on Patterns Language of Programming (PLop'97), Monticello, IL, USA, 1997.