D. G. Rosado, E. Fernandez-Medina, and J. Lopez, "Extensión UML para Casos de Uso Reutilizables en entornos Grid Móviles Seguros", *XIV Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2009)*, pp. 331-342, 2009.
NICS Lab. Publications: https://www.nics.uma.es/publications

Extensión UML para Casos de Uso Reutilizables en entornos Grid Móviles Seguros

David G. Rosado[1], Eduardo Fernández-Medina[1] y Javier López[2]

[1] UCLM. Grupo Alarcos – Instituto de Tecnologías y Sistemas de Información. Dep. de Tecnologías y Sistemas de Información – ESI, Paseo de la Universidad 4, 13071 Ciudad Real
{David.GRosado, Eduardo.FdezMedina}@uclm.es
[2] Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga, jlm@lcc.uma.es

Resumen. Los sistemas Grid nos permiten construir sistemas complejos con características diferenciadoras (interoperabilidad entre múltiples dominios de seguridad, autenticación y autorización a través de dominios, sistema dinámico y heterogéneo, etc.). Con el desarrollo de la tecnología wireless y los dispositivos móviles, el Grid llega a ser el candidato perfecto para que los usuarios móviles puedan realizar trabajos complejos, a la vez que añaden nueva capacidad computacional al Grid. Estamos construyendo un proceso completo de desarrollo para sistemas Grid móviles seguros, y una de las actividades es el análisis de requisitos, que está basado en casos de uso reutilizables. En este artículo, presentaremos una extensión UML para casos de uso de seguridad y Grid, los cuales capturan el comportamiento de este tipo de sistemas. Esta extensión UML está siendo aplicado a un caso real para construir diagramas de casos de uso de la aplicación, incorporando los aspectos de seguridad necesarios.

Palabras Claves: Seguridad, Casos de uso de Seguridad, desarrollo seguro, Grid Móvil seguro, Reutilización.

1 Introducción

La creciente necesidad de construir sistemas seguros, debido principalmente a las nuevas vulnerabilidades derivadas del uso de Internet y de las aplicaciones distribuidas en entornos heterogéneos, motiva a la comunidad científica a demandar una clara integración de la seguridad dentro de los procesos de desarrollo [1-4]. Un tipo de sistemas que tiene características diferenciadoras claras [5-7], y donde la seguridad es un factor de suma importancia, son los sistemas basados en Grid Computing. Los procesos de desarrollo genéricos son usados para desarrollar sistemas sin tener en cuenta ni el entorno tecnológico subyacente, ni las características y particularidades de estos sistemas específicos.

La seguridad es un aspecto central en la computación Grid desde el principio, y ha sido considerado como el cambio más significativo de la computación Grid [8, 9]. Además, la seguridad es más difícil de implementar dentro de una plataforma móvil

debido a las limitaciones de recursos de los dispositivos móviles [10]. Por tanto, una infraestructura Grid que soporte la participación de nodos móviles jugará un importante papel en el desarrollo de la computación Grid.

La mayoría de aplicaciones Grid existentes se han construido sin un proceso sistemático de desarrollo, basándose en desarrollos ad-hoc [5, 11]. La falta de métodos de desarrollo adecuados para este tipo de sistemas nos ha motivado a construir una metodología para desarrollarlos [12-14], ofreciendo una guía detallada para analizarlos, diseñarlos e implementarlos. La metodología está fuertemente orientada hacia la reutilización, y especialmente sensibilizada con la seguridad y la utilización de dispositivos móviles en los Grids Computacionales. La reutilización se concentra principalmente i) en la etapa de análisis en la que se parte de un conjunto de casos de uso predefinidos (y diagramas de interacción), y que se integran con los casos de uso identificados para una nueva aplicación y ii) en la etapa de diseño, en la que se parte de una arquitectura que incorpora los servicios de seguridad reutilizables previamente identificados, y se especializa para cada una de las nuevas aplicaciones que sean creadas.

En este artículo, presentaremos una extensión UML para casos de uso Grid reutilizables que pueda ser usada en la actividad de análisis, junto con diagramas de interacción y escenarios, para construir diagramas de casos de uso, integrando los requisitos para las aplicaciones Grid móviles seguras específicas. Esta extensión define los estereotipos necesarios para poder especificar detalles sobre los casos de uso Grid y casos de uso de seguridad que sirven de ayuda en la construcción de diagramas de casos de uso para este tipo de sistemas. También aplicamos esta extensión UML a un caso real donde estos casos de uso reutilizables, almacenados en el repositorio, son usados para construir un diagrama general para esta aplicación. Este caso real es un escenario en el dominio periodístico, dentro del proyecto europeo GREDIA [15], donde trabaja un grupo de investigación de la Universidad de Málaga, ofreciendo controles de seguridad para la infraestructura Grid móvil que le da soporte.

El resto del artículo se organiza como sigue: En la sección 2, presentaremos el trabajo relacionado. En la sección 3, se definirán los estereotipos y asociaciones para los casos de uso Grid y los describiremos formalmente. En la sección 4, aplicaremos los nuevos esterotipos para construir un diagrama de casos de uso en un caso real. Terminaremos proponiendo las conclusiones y algunas líneas de investigación sobre el trabajo futuro en la sección 5.

2 Antecedentes

La idea de desarrollar software mediante procesos de desarrollo sistemáticos para mejorar la calidad del software no es nueva [16-18]. Sin embargo, todavía hay muchos sistemas de información, tal como los Grid computing, que no son desarrollados mediante metodologías adaptadas a sus características más diferenciadoras [5]. De hecho, no hemos encontrado propuestas para el desarrollo sistemático de sistemas Grid, a pesar de la demanda de estos sistemas en la comunidad científica.

Por ejemplo, los autores en [19] presentan una metodología para la integración de la seguridad en los sistemas software. Esta metodología está basada en el Proceso Unificado [18] y es llamada Proceso Unificado Seguro (SUP – Secure Unified Process) . El problema es que sólo ofrece una solución a muy alto nivel sin ofrecer "mecanismos prácticos" (por ejemplo, artefactos de seguridad específicos de Grid, o una arquitectura de seguridad de referencia) que permita implementar su propuesta en un corto espacio de tiempo y con un mínimo esfuerzo. Otra propuesta [20, 21] se concentra en proporcionar semánticas formales en UML para integrar consideraciones de seguridad dentro del proceso de diseño software. La propuesta presenta UMLSec, que es una extensión de UML y permite expresar información relevante de seguridad. UMLSec y nuestra propuesta son compatibles, mientras que los modelos desde UMLSec pueden ser usados para especificar aspectos de seguridad generales de los sistemas, nuestra propuesta podría ser usada para especificar características de seguridad para entornos Grid.

Por otro lado, la actual arquitectura Grid no tienen en cuenta los entornos móviles debido a que los dispositivos móviles no han sido considerados como recursos de computación válidos o interfaces en la comunidad Grid. Es actualmente cuando se está prestando más atención a integrar estas dos tecnologías emergentes, computación Grid y computación móvil, como muestran algunos trabajos en [22-25], aunque ninguno elabora la forma de incorporar los dispositivos móviles en la actual arquitectura Grid. La metodología que proponemos considera por un lado, la incorporación de dispositivos móviles como un recurso más del Grid, y por otro lado, esta incorporación se realiza desde el principio del desarrollo sistemático, considerando todos los aspectos de seguridad y limitaciones de estos dispositivos.

3 Extensión de casos de uso para entornos Grid móviles seguros

La estructura de la metodología que estamos definiendo sigue el ciclo clásico, donde tenemos una etapa de planificación, una de desarrollo que incluye análisis, diseño y construcción, y finalmente, de una etapa de mantenimiento, sin embargo, está especialmente diseñada para este tipo de sistemas, con características tan particulares. Detalle sobre las actividades de la metodología pueden encontrarse en [12-14].

La actividad de análisis está centrada en casos de uso, donde se define el comportamiento, acciones e interacciones con los implicados en el sistema (actores), obteniendo una primera aproximación a las necesidades y requisitos (funcionales y no funcionales) del sistema a construir. Esta actividad se apoya en la reutilización de casos de uso Grid, almacenados en el repositorio de donde se obtienen casos de uso correctos que definen un comportamiento común del sistema Grid, y que son muy utilizados en la mayoría de diagramas de casos de uso que se construyen para diferentes sistemas Grid.

Para definir diagramas de casos de uso reutilizables específicos para los sistemas Grid móviles, necesitamos extender el metamodelo de UML 2.0 y definir un nuevo perfil con nuevos estereotipos. Un estereotipo es una extensión del vocabulario de UML que permite crear nuevos bloques de construcción derivados de los existentes, pero específicos a un dominio concreto, en nuestro caso al dominio Grid computing.

En esta sección, presentamos la extensión GridUCSec-Profile mediante la cual es posible representar características Grid móvil específicas y aspectos de seguridad para los diagramas de casos de uso, obteniendo como resultado, diagramas de casos de uso para entornos Grid móviles seguros y reutilizables. Esta extensión ha sido construida como un perfil UML, el cual es un mecanismo de extensibilidad que permite adaptar las metaclases de un modelo haciendo posible la incorporación de nuevos elementos en un dominio.

3.1 Extensión UML: GridUCSec-Profile

Para la representación de casos de uso Grid y casos de uso de seguridad, ha sido definido un conjunto de estereotipos, los cuales han sido agrupados en paquetes, *GridUCSec* y *TypesGridUCSec* que son parte de GridUCSec-Profile.

El paquete *GridUCSec* (ver Fig. 1) está compuesto de casos de uso Grid, casos de uso de seguridad, casos de mal uso, asociaciones de permiso, protección, amenaza y mitigación, junto con los actores involucrados, todos ellos necesarios para capturar el mayor número de aspectos de seguridad para sistemas Grid. Este paquete tiene 11 estereotipos: 4 especializan a *UseCase*, 2 especializan a *Actor*, y 5 especializan a *DirectedRelationship* y *NamedElement*. Los estereotipos que componen este paquete serán definidos en las siguientes subsecciones.

El paquete TypesGridUCSec (ver Fig. 1) define los tipos de datos para los valores etiquetados de los estereotipos de GridUCSec-Profile, como son el nivel de protección y riesgo, tipos de permiso, de requisito, de activos, de ataque, etc. Este paquete está compuesto de 9 estereotipos que especializan la clase *Enumeration*.

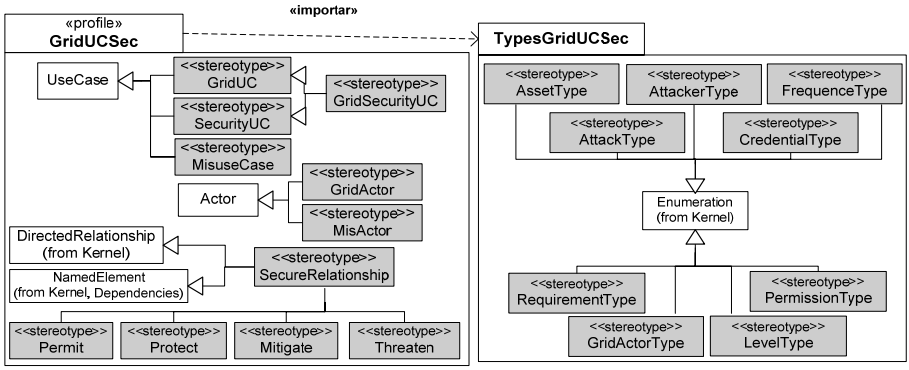

Fig. 1. Metamodelo de GridUCSec-Profile y TypesGridUCSec

En la Fig. 2, podemos ver las asociaciones entre estos nuevos estereotipos definidos para construir diagramas de casos de uso para sistemas Grid móviles seguros. Los estereotipos definidos para casos de uso se asocian con los estereotipos definidos para las relaciones entre casos de uso, indicando la acción (proteger, mitigar, permitir o amenazar) que ejerce un caso de uso sobre otro caso de uso mediante estos estereotipos de relación. Así, por ejemplo, el estereotipo «*SecurityUC*» se asocia con «*UseCase*» a través del estereotipo de relación «*Protect*» indicando que

el caso de uso de seguridad puede establecer una relación de protección con otro caso de uso. O dicho de otra forma, el estereotipo de relación «*Protect*» relaciona un caso de uso de seguridad con otro caso de uso para establecer la relación de protección. Hay varias restricciones que debemos definir y también, debemos describir en profundidad estas asociaciones de forma detallada para construir un perfil UML completo y correcto en el contexto de casos de uso para aplicaciones Grid. Esta descripción detallada será mostrada en la siguiente subsección.

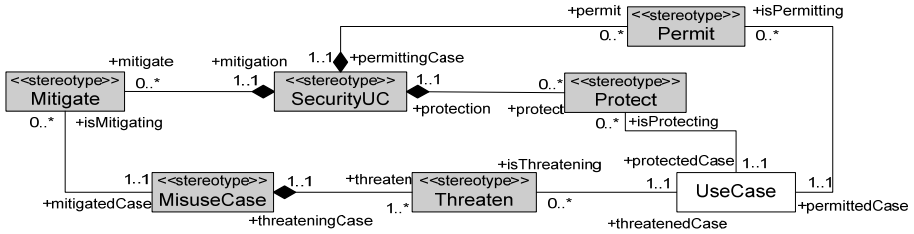

Fig. 2. Relación entre UseCase y DirectedRelationship.

3.2. Descripción detallada de estereotipos para el paquete GridUCSec

Una vez identificados los estereotipos y asociaciones entre ellos, ahora damos una descripción detallada de cada uno de ellos, describiendo los valores etiquetados y restricciones que definen su semántica. Usamos una plantilla, mostrada en la Tabla 1, basada en la especificación UML 2.0 [26], para describir formalmente los estereotipos anteriormente definidos. Esta plantilla simplifica, extiende y proporciona descripciones de cada uno de los estereotipos, facilitando su reutilización. Los elementos de esta plantilla son: *Descripción*: Indica el propósito y significado para los distintos usuarios del estereotipo. *Generalización*: Identifica el elemento del metamodelo de UML (clase base) desde la cual es heredado el estereotipo. *Asociaciones*: Se identifican las relaciones que tiene el estereotipo, tanto con otros estereotipos como con los elementos de UML. *Notación*: Corresponde a un icono que se asocia al estereotipo para su representación gráfica. *Valores Etiquetados*: Se identifican los atributos asociados al estereotipo. *Restricciones*: Corresponde a la descripción de un conjunto de limitaciones que tiene el estereotipo y la forma en que el estereotipo se relaciona con el resto de los estereotipos y con los elementos de UML. Estas restricciones se describen en forma textual y son definidas mediante expresiones OCL.

Tabla 1. Descripción detalla de los estereotipos para el paquete GridUCSec (Los números en la primera columna indican: 1) Descripción; 2) Generalización; 3) Asociaciones; 4) Notación; 5) Valores Etiquetados; y 6) Restricciones)

	«GridUC»	A	«SecurityUC»	B
1	Especifica los requisitos del sistema Grid y representa el comportamiento común, las funciones y las relaciones para esta clase de		Especifica los requisitos de seguridad del sistema, describiendo las tareas de seguridad que los usuarios deben ser capaces de desempeñar	

	sistemas.	por medio del sistema.
2	Classifier::BehavioredClassifier::UseCase	
3	-isPermitting:Permit[0..*]. Referencia la relación Permit que está permitiendo a este UC. -isProtecting:Protect[0..*]. Referencia la relación Protect que está protegiendo a este UC. -isThreatening:Threaten[0..*]. Referencia la relación Threaten que está amenazando a este UC.	-mitigate:Mitigate[0..*]. Referencia la relación Mitigate que pertenece a este UC de seguridad. -permit:Permit[0..*]. Referencia a la relación Permit que pertenece a este UC de seguridad. -protect:Protect[0..*]. Referencia a la relación Protect que pertenece a este UC de seguridad.
4		
5	GridRequirement, ProtectionLevel, SecurityDependence, InvolvedAsset	SecurityRequirement, InvolvedAsset, SecurityDegree
6	context GridUC inv: self.GridRequirement→size()=1 inv: self.InvolvedAsset→size()=1 inv: self.ProtectionLevel→size()=1 inv: self.SecurityDependence→size()=1 inv: self.MisActor→size()=0 inv: self.GridActor→size()>=0	context SecurityUC inv: self.Threaten→size()=0 inv: self.SecurityRequirement→size()=1 inv: self.InvolvedAsset→size()=1 inv: self.SecurityDegree→size()=1 inv: (self.mitigation→size() + self.permit→size()+ self.protect→size())>=1 inv: self.MisActor→size()=0 inv: self.GridActor→size()>=0
	«GridSecurityUC» **C**	**«MisuseCase»** **D**
1	Representan características de seguridad específicas de sistemas Grid. Especializa a los UC de seguridad comunes de otras aplicaciones, proporcionando características únicas para entornos Grid.	Representa una secuencia de acciones, incluyendo variantes, que un sistema u otra entidad puede desempeñar, interactuando con atacantes de la entidad y causando daño si la secuencia se completa.
2	- UseCase::SecurityUC; - UseCase::GridUC	- Classifier::BehavioredClassifier::UseCase
3	- Hereda asociaciones de SecurityUC. - Sólo hereda de GridUC la asociación isPermitting.	- threaten:Threaten [1..*]. Referencia la relación Threaten que pertenece a este caso de mal uso. - isMitigating:Mitigate [0..*]. Referencia la relación Mitigate que está mitigando este caso de mal uso.
4		
5	InvolvedAsset, SecurityRequirement, SecurityDegree, SecurityDependence	InvolvedAsset, ImpactLevel, RiskLevel, ThreatLikelihood, KindAttack
6	context GridSecurityUC inv: self.SecurityRequirement→size()=1 inv: self.InvolvedAsset→size()=1 inv: self.ProtectionLevel→size()=1 inv: self.SecurityDependence→size()=1. inv: self.Threaten→size()=0 inv: self.Protect→size()=0	context MisuseCase inv: self.Permit→size()=0 inv: self.Protect→size()=0 inv: self.MisActor→size()>=0 inv: self.GridActor→size()=0 inv: self.KindAttack→size()=1 inv: self.InvolvedAsset→size()=1 inv: self.ImpactLevel→size()=1 inv: self.RiskLevel→size()=1 inv: self.ThreatLikelihood→size()=1
	«Protect» **E**	**«Permit»** **F**
1	Esta relación especifica que el comportamiento de un UC podría ser protegido por el comportamiento de un UC de seguridad.	Esta relación especifica que el comportamiento de un UC podría ser permitido por el comportamiento de un UC de seguridad..
2	Element::Relationship::DirectedRelationship:: SecureRelationship Element::NamedElement:: SecureRelationship	
3	- protection:SecurityUC [1..1]. Referencia el UC que representa la protección y posee la relación protect.	- permittingCase:SecurityUC [1..1]. Referencia el UC que representa el permiso y posee la relación permit (SecurityUC o GridSecurityUC).

	- protectedCase:UseCase [1..1]. Referencia el UC que está siendo protegido.	- permittedCase: UseCase [1..1]. Referencia el UC que está siendo permitido (UseCase).
4	<<protect>>	<<permit>>
5	InvolvedAsset, ProtectionLevel, KindAttack	PermissionCondition, KindPermission
6	context Protect inv: (self.protectedCase→isTypeOf(UseCase) or self.protectedCase→isTypeOf(GridUC)) inv: (self.protection→isTypeOf(SecurityUC) or self.protection→isTypeOf(GridSecurityUC)) inv: self.InvolvedAsset→size()=1 inv: self.ProtectionLevel→size()=1 inv: self.KindAttack→size()=1 inv: self.InvolvedAsset= self.protectedCase.InvolvedAsset inv: self.IKindAttack= self.protectedCase.KindAttack	context Permit inv: (self.permitedCase→isTypeOf(GridUC) or self.permitedCase→ isTypeOf(GridSecurityUC)) inv: (self.permittingCase→ isTypeOf(SecurityUC) or self.permittingCase→ isTypeOf(GridSecurityUC)) inv: self.KindPermission→size()=1
	«Mitigate» G	**«Threaten» H**
1	Esta relación especifica que el comportamiento de un caso de mal uso podría ser mitigado por el comportamiento de un UC de seguridad.	Esta relación especifica que el comportamiento de un UC podría ser amenazado por el comportamiento de un caso de mal uso.
2	Element::Relationship::DirectedRelationship:: SecureRelationship Element::NamedElement:: SecureRelationship	
3	- mitigation:SecurityUC [1..1].]. Referencia el UC que representa la mitigación y posee la relación mitigate (SecurityUC o GridSecurityUC) - mitigatedCase: MisuseCase [1..1]. Referencia el UC que está siendo mitigado (MisuseCase).	- threatenedCase: UseCase [1..1]. Referencia el UC que está siendo amenazado (UseCase o GridUC). - threateningCase: MisuseCase [1..1]. Referencia el UC que representa la amenaza y posee la relación threaten.
4	<<mitigate>>	<<threaten>>
5	SuccessPercentage, KindCountermeasure	SuccessPercentage, KindVulnerability, KindAttack
6	context Mitigate inv: (self.mitigation→isTypeOf(SecurityUC) or self.mitigation→ isTypeOf(GridSecurityUC)) inv: self.SuccessPercentage→size()=1 inv: self.KindCountermeasure→size()=1	context Threaten inv: (self.threatenedCase→isTypeOf(UseCase) or self.threatenedCase→isTypeOf(GridUC)) inv: self.SuccessPercentage→size()=1 inv: self.KindVulnerability→size()=1 inv: self.KindAttack→size()=1
	«GridActor» I	**«MisActor» J**
1	Este actor especifica el rol jugado por un usuario Grid u otro sistema grid que interactúa con el sistema. **4** «GridActor»	Este actor especifica el rol jugado por un atacante o cualquier otro ataque que interactúa con el sistema. **4** «MisActor»
2	Classifier::Actor	Classifier::Actor
3	asociaciones con UseCase, GridUC, SecurityUC y GridSecurityUC	Sólo tiene asociaciones con MisuseCase
5	KindGridCredential, KindGridActor, KindRole, OrganizationName	KindMisActor, HarmDegree
6	context GridActor inv: self.KindGridCredential→size()>=0 inv: self.KindGridActor→size()=1 inv: self.KindRole→size()=1 inv: self.OrganizationName→size()>=1 inv: self.MisuseCase→size()=0	context Misuser inv: self.MisuseCase→size()>=1 inv: self.GridUC→size()=0 inv: self.SecurityUC→size()=0 inv: self.KindMisuser→size()=1 inv: self.HarmDegree→size()=1

3.3. Valores Etiquetados y estereotipos del paquete TypesGridUCSec

Presentamos en la Tabla 2, la descripción detallada de cada uno de los valores etiquetados definidos en el paquete GridUCSec. Para definir los valores etiquetados damos una pequeña descripción e identificamos los valores posibles de cada tipo asociados con el valor etiquetado (en la tabla se indica el número del tipo, que se puede ver en la Tabla 3).

Tabla 2. Valores Etiquetados de los estereotipos definidos en el paquete GridUCSec

Valor Etiquetado	Descripción	Nº Tipo
GridRequirement	Contiene los tipos de requisitos involucrados en el caso de uso.	9
HarmDegree	Define el grado de daño que un atacante puede causar al sistema.	7
ImpactLevel	Indica el nivel de impacto en el sistema que puede causar si alguna de las amenazas lleva a cabo un ataque con éxito.	7
InvolvedAsset	Identifica los activos que deben ser protegidos y que toman parte en la realización del caso de uso.	1
KindAttack	Describe el tipo de ataque que es llevado a cabo sobre el sistema abriendo un hueco de seguridad.	2
Kind Countermeasure	Describe las decisiones de cómo proteger la seguridad y privacidad del atacante potencial y de las vulnerabilidades.	10
KindGridActor	Define el tipo de actor Grid que interactúa con el sistema.	6
KindGrid Credential	Cuando un actor Grid quiere interactuar con el sistema, éste debe presentar o soportar un tipo de credencial de seguridad.	4
KindMisActor	El tipo de atacante que accede al sistema con el propósito de dañarlo.	3
KindPermission	Indica el tipo de permiso concedido para la realización de un UC.	8
KindRole	Indica el role de un actor en el sistema, así los privilegios asociados a un actor son conocidos.	10
Kind Vulnerability	Identifica los tipos de vulnerabilidades encontradas en el sistema y que son posibles candidatas de ataques.	10
OrganizationName	Todos los usuarios deben pertenecer a alguna organización.	10
Permission Condition	Define las condiciones necesarias bajo las cuales un caso de uso permite la realización de otro caso de uso.	10
ProtectionLevel	Indica el nivel de protección que el caso de uso debe tener.	7
RiskLevel	El nivel de riesgo es considerado en función de amenazas y vulnerabilidades de los activos.	7
SecurityDegree	Describe el grado de seguridad aportado por este UC al sistema.	7
Security Dependence	Indica si un caso de uso necesita ser asegurado porque su comportamiento puede generar un riesgo para el sistema.	7
Security Requirement	Contienen los tipos de requisitos de seguridad que intervienen en los casos de uso de seguridad.	9
Success Percentage	Indica el porcentaje de que una cierta acción (de seguridad, de ataque, etc.) tenga éxito.	7
ThreatLikelihood	Es la probabilidad que una amenaza sea llevada a cabo.	5

Para la definición de los estereotipos de la extensión GridUCSec, ha sido necesario definir tipos de datos, generales y específicos Grid, que representan los valores etiquetados y atributos de esos nuevos estereotipos (ver Tabla 3). Estos tipos están disponibles en el GridUCSec-Profile para que los atributos de los nuevos estereotipos puedan ser definidos y reutilizados.

Tabla 3. Valores de los Tipos definidos en el paquete TypesGridUCSec

Nº	Tipo	Valores
1	AssetType	Accounting, Credential, Data, General, Identity, Message, Resource, User.
2	AttackType	AccesControlAtt, ColludingAtt, DefeatingAtt, DoSAtt, EavesdroppingAtt, IntruderAtt, MaliciousAtt, MasqueradingAtt,ObjectReuseAtt, SniffingAtt
3	AttackerType	hacker, cracker, script kiddies, newbies, lamers, virus, Trojan, …
4	CredentialType	UserPass, X509, Kerberos, SAML, PIN, Biometric
5	FrequencyType	VFrequent, Frequent, Normal, Rare
6	GridActorType	User, MobileUser, Service, MobileResource, Resource, VO, Host
7	LevelType	VHigh, High, Medium, Low, VLow
8	PermissionType	Execute, CheckExecute, Interact, Include, Extend, Protect, Mitigate
9	Requirement Type	Accounting, Anonymity,Antivirus, Authentication, AuthenticationMutual, Authorization&AC, Availability, Confidentiality, Credential, Delegation, Firewall&IntrusionPrevention, Integration, Integrity, Interoperability, MappingIdentity, MultipleImplementation, NonRepudiation, Privacy, Revocation, Scalability, SSO, Trust, Usability
10	String	Cadena de caracteres

4 Ejemplo

La metodología que proponemos está siendo validada mediante su aplicación en un sistema real, una aplicación de negocio en el dominio multimedia y noticias, definido para el proyecto europeo GREDIA [15]. Queremos construir un sistema que ofrezca al periodista (en continuo movimiento) con un equipo ligero, la posibilidad de capturar y transmitir contenidos de información, manteniendo en todo momento los controles de seguridad necesarios.

Usando la extensión GridUCSec-Profile definida en este artículo, construiremos un diagrama de casos de uso para esta aplicación ayudándonos de los casos de uso reutilizables almacenados en el repositorio usado por la metodología propuesta. Por motivos de espacio y debido a su complejidad, sólo mostramos un caso de uso (Búsqueda Noticias) para este caso real de todos los definidos en el proyecto GREDIA (ver Fig. 3).

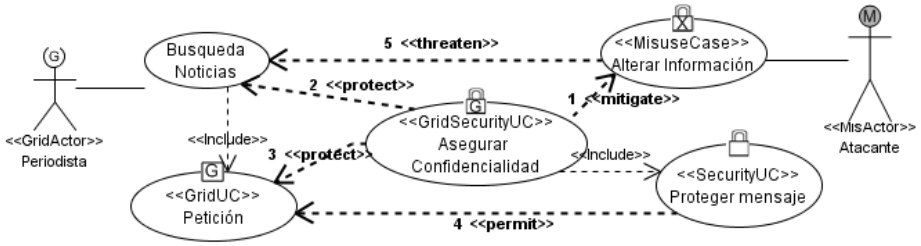

Fig. 3. Usando casos de uso reutilizables del repositorio para construir el diagrama final de la aplicación.

En este diagrama, el periodista («GridActor») hace una búsqueda de noticias (UseCase) a través del sistema para familiarizarse con el tema a tratar. Esta acción requiere hacer una petición al sistema Grid, representada por el caso de uso "Petición" («GridUC») que es reutilizable, al tratarse de un caso de uso común en sistemas Grid.

En este escenario, un atacante puede alterar la información que se trasmite a o desde el sistema Grid, por lo que es conveniente proteger los mensajes y el contenido de posible ataques. Todo esto se modela mediante un atacante («MisActor») que realiza el ataque de alterar la información («MisuseCase») sobre la búsqueda de noticias realizada por el periodista («GridActor»). Disponemos de un caso de uso de seguridad reutilizable que asegura la confidencialidad («GridSecurityUC») y mitiga el ataque, protegiendo la búsqueda de noticias y la petición en el Grid, a la vez que permite la realización de las peticiones, una vez haya sido protegido el mensaje. Este diagrama de casos de uso debe ser apoyado por diagramas de secuencia, que también estarán disponibles en el repositorio, para facilitar su comprensión y entender mejor las interacciones entre casos de uso y actores.

En la Tabla 4, mostramos la información relacionada con el diagrama de la

Fig. 3., describiendo los casos de uso y atributos siguiendo la extensión GridUCSec-Profile para este ejemplo. Todos los detalles que aparecen en la tabla deben ser definidos a la vez que construimos el diagrama, y son importantes para tomar decisiones de seguridad (definir servicios, mecanismos, etc.) en la actividad de diseño, para establecer la relación de trazabilidad entre casos de uso y servicios dentro de la arquitectura, o para el procesamiento automático en alguna herramienta CASE.

Tabla 4. Aplicación de GridUCSec-profile a un caso de estudio

	Estereotipos Asociaciones	**Asociaciones (Asoc) y Valores etiquetados (VEti) para los estereotipos de asociaciones**		
1	mitigate: Mitigate	Asoc	mitigation: Asegurar Confidencialidad mitigatedCase: Alterar información	
		VEti	KindCountermeasure: cifrar mensaje	SuccessPercentage: {High}
2	protect: Protect	Asoc	protection: Asegurar Confidencialidad	protectedCase: Búsqueda noticias
		VEti	InvolvedAsset: {Message} KindAttack: {MasqueradingAtt, EavesdroppingAtt}	ProtectionLevel: {High}
3	protect: Protect	Asoc	protection: Asegurar Confidencialidad	protectedCase: Petición
		VEti	InvolvedAsset: {Message} KindAttack: {MasqueradingAtt}	ProtectionLevel: {High}
4	permit: Permit	Asoc	permittingCase: Proteger mensaje	permittedCase: Petición
		VEti	PermissionCondition: todos los mensajes cifrados KindPermission: Execute	
5	threaten: Threaten	Asoc	threateningCase: Alterar información	threatenedCase: Búsqueda noticias
		VEti	SuccessPercentage: {High} KindVulnerability: mensajes por red inalámbrica KindAttack: {MasqueradingAtt, EavesdroppingAtt}	

Estereotipos	**Valores Etiquetados**	
«GridSecurityUC» Asegurar Confidencialidad	InvolvedAsset: {Message} SecurityDegree: {High}	Securityrequirement: {Confidentiality} SecurityDependence: {VLow}
«SecurityUC» Proteger mensaje	InvolvedAsset: {Message} SecurityDegree: {High}	SecurityRequirement: {Confidentiality}
«GridUC» Petición	InvolvedAsset: {Message} ProtectionLevel: {Medium}	GridRequirement: {Interoperatibility} SecurityDependence: {Medium}
«MisuseCase» Alteración información	ImpactLevel: {High} RiskLevel: {High}	InvolvedAsset:{Message,Identity,Data} KindAttack: {MasqueradingAtt} ThreatLikelihood: {Frequent}
«GridActor» Periodista	KindRole: periodista OrganizationName: News	KindGridActor: {Mobile User} KindGridCredential: {UserPass}
«MisActor» Atacante	KindMisActor: hacker	HarmDegree: {Medium}

5 Conclusiones y trabajo futuro

Una metodología de desarrollo para sistemas Grid móviles seguros nos sirve de guía para obtener un software de calidad, ofreciendo los métodos, técnicas y herramientas que faciliten la labor a todo el equipo involucrado en el desarrollo del software. Estudiando las necesidades y particularidades de los sistemas Grid móviles, fue necesario definir una extensión UML para casos de uso que capture el comportamiento, las funciones, las propiedades y necesidades que surgen en este tipo de sistemas. Esta extensión enriquece los diagramas de casos de uso con aspectos de seguridad y define valores que son esenciales a la hora de interpretar el diagrama de casos de uso en las sucesivas actividades del proceso de desarrollo.

La reutilización de elementos es una característica fundamental del proceso y ofrece a los desarrolladores e implicados en el desarrollo, soluciones construidas y probadas, mejorando la calidad del producto final, ahorrando tiempo y esfuerzo y mejorando la productividad en el proceso de desarrollo. Gracias a la extensión UML para casos de uso, podemos analizar los requisitos de seguridad del sistema desde las primeras etapas de desarrollo, y capturar toda la información de seguridad relevante que será necesaria en las siguientes actividades del proceso de desarrollo.

Como trabajo futuro se pretende completar el detalle de la metodología (actividades, tareas, etc.) a través del método de investigación-acción, integrar técnicas de ingeniería de requisitos de seguridad (UMLSec, etc.), y definir la trazabilidad de artefactos entre actividades.

Acknowledgments. Esta investigación es parte de los siguientes proyectos: QUASIMODO (PAC08-0157-0668) financiado por la "Viceconsejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha" (España), y ESFINGE (TIN2006-15175-C05-05) concedida por la "Dirección General de Investigación del Ministerio de Educación y Ciencia" (España). Agradecimiento especial a GREDIA (FP6-IST-034363) financiado por la Comisión Europea.

References

1. Bass, L., Bachmann, F., Ellison, R.J., Moore, A.P., Klein, M.: Security and survivability reasoning frameworks and architectural design tactics. SEI (2004)
2. Jürjens, J.: Secure Systems Development with UML. Springer-Verlag (2004)
3. Lodderstedt, T., Basin, D., Doser, J.: SecureUML: A UML-Based Modeling Language for Model-Driven Security. 5th International Conference on the Unified Modeling Language (UML), 2002, Vol. 2460. Springer, Dresden, Germany (2002) 426--441
4. Mouratidis, H., Giorgini, P.: Integrating Security and Software Engineering: Advances and Future Vision. IGI Global (2006)
5. Kolonay, R., Sobolewski, M.: Grid Interactive Service-oriented Programming Environment. Concurrent Engineering: The Worldwide Engineering Grid. Press and Springer Verlag, Tsinghua, China (2004) 97–102
6. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: Grid services for distributed system integration. Computer 35 (2002) 37-46
7. Foster, I., Kesselman, C.: Globus: A Toolkit-Based Grid Architecture. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann (1999) 259-278

8. Humphrey, M., Thompson, M.R., Jackson, K.R.: Security for Grids. Lawrence Berkeley National Laboratory. Paper LBNL-54853 (2005)
9. Chakrabarti, A., Damodaran, A., Sengupta, S.: Grid Computing Security: A Taxonomy. IEEE Security & Privacy 6 (2008) 44-51
10. Bradford, P.G., Grizzell, B.M., Jay, G.T., Jenkins, J.T.: Cap. 4. Pragmatic Security for Constrained Wireless Networks. In: Publications, A. (ed.): Security in Distributed, Grid, Mobile, and Pervasive Computing, University of Alabama, USA (2007) 440
11. Dail, H., Sievert, O., Berman, F., Casanova, H., YarKhan, A., Vadhiyar, S., Dongarra, J., Liu, C., Yang, L., Angulo, D., Foster, I.: Scheduling In The Grid Application Development Software Project. Grid resource management:state of the art and future trends (2004) 73-98
12. Rosado, D.G., Fernández-Medina, E., López, J., Piattini, M.: PSecGCM: Process for the development of Secure Grid Computing based Systems with Mobile devices. International Conference on Availability, Reliability and Security (ARES'08). IEEE, Barcelona, Spain (2008) 136-142
13. Rosado, D.G., Fernández-Medina, E., López, J., Piattini, M.: Engineering Process Based On Grid Use Cases For Mobile Grid Systems. The Third International Conference on Software and Data Technologies- ICSOFT 2008, Porto, Portugal (2008) 146-151
14. Rosado, D.G., Fernández-Medina, E., López, J.: Obtaining Security Requirements for a Mobile Grid System. International Journal of Grid and High Performance Computing (2009) (to be published in April 1, 2009)
15. GREDIA, www.gredia.eu
16. Bell, D.E., LaPadula, L.J.: Secure Computer System: Unified Exposition and Multics Interpretation. Technical Report MTR-2997. Bedford, MA (1976)
17. Baskerville, R.: Information systems security design methods: implications for information systems development. ACM Computing Surveys 25 (1993) 375 - 414
18. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley Professional (1999)
19. Steel, C., Nagappan, R., Lai, R.: Chapter 8. The Alchemy of Security Design Methodology, Patterns, and Reality Checks. Core Security Patterns:Best Practices and Strategies for J2EE, Web Services, and Identity Management. Prentice Hall (2005) 1088
20. Jurjens, J.: Towards Development of Secure Systems Using UMLsec. In: LNCS, S.-V. (ed.): Fundamental Approaches to Software Engineering (FASE/ETAPS) (2001)
21. Jurjens, J.: UMLsec: Extending UML for Secure Systems Development. In: Springer (ed.): 5th International Conference on the Unified Modeling Language (UML), Vol. 2460, Dresden, Germany (2002) 1-9
22. Hans A. Franke, Fernando L. Koch, Carlos O. Rolim, Carlos B. Westphall, Douglas O. Balen: Grid-M: Middleware to Integrate Mobile Devices, Sensors and Grid Computing. Third International Conference on Wireless and Mobile Communications (ICWMC'07) Guadeloupe, French Caribbean (2007) 19
23. Isaiadis, S., Getov, V.: Integrating Mobile Devices into the Grid: Design Considerations and Evaluation. In: Jose C. and Medeiros, P.D. (ed.): 11th International Euro-Par Conference (Euro-Par 2005). LNCS (3648), Lisbon, Portugal (2005) 1080-1088
24. Jameel, H., Kalim, U., Sajjad, A., Lee, S., Jeon, T.: Mobile-To-Grid Middleware: Bridging the gap between mobile and Grid environments. European Grid Conference EGC 2005, Vol. 3470/2005. Springer, Amsterdam, The Netherlands (2005) 932-941
25. Phan, T., Huang, L., Dulan, C.: Challenge: Integrating Mobile Wireless Devices Into the Computational Grid. 8th annual international conference on Mobile computing and networking (MobiCom'02). ACM Press, Atlanta, Georgia, USA (2002) 271 - 278
26. OMG: OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2. (2007)