GESTIÓN AUTOMATIZADA DE REQUISITOS DE SEGURIDAD EN PROYECTOS DE DESARROLLO DE SISTEMAS DE INFORMACIÓN

Daniel Mellado Fernández
Oficina de Gestión de Proyectos
Centro de Desarrollo del INSS.
Gerencia de Informática de la Seguridad Social

Moisés Rodríguez Monje
Consultor
Kybele Consulting

Eduardo Fernández-Medina Patón
Profesor Titular de Universidad
Universidad de Castilla-La Mancha

Mario Piattini Velthuis
Catedrático de Universidad
Universidad de Castilla-La Mancha

Palabras clave

Seguridad, ingeniería de requisitos de seguridad, ingeniería de requisitos, Métrica, MAGERIT, herramienta de gestión de requisitos.

Resumen

La integración de la seguridad en las primeras fases del desarrollo de Sistemas de Información (SI) es necesario si se quiere construir SI seguros. Sin embargo, en muchos proyectos software la seguridad se trata cuando el sistema ya ha sido diseñado y puesto en operación. Por ello, la denominada Ingeniería de Requisitos de Seguridad es una disciplina que se está erigiendo como una importante rama de la Ingeniería del Software, debido a que se está comprendiendo cada vez más que la seguridad debe abordarse desde el inicio de la fase de requisitos. Pero sin una herramienta CARE (Computer-Aided Requirements Engineering), la aplicación de cualquier metodología o proceso de ingeniería de requisitos resulta difícil de implantar y de que tenga éxito si se tiene que realizar manualmente. Por tanto, en esta comunicación presentamos el prototipo de una herramienta CARE llamada SREPTOOL, que da soporte automatizado al proceso de ingeniería de requisitos de seguridad SREP (Security Requirements Engineering Process) para facilitar su aplicación práctica. SREPTOOL simplifica la gestión de requisitos de seguridad proporcionando una forma guiada, sistemática e intuitiva de tratarlos desde las primeras fases del desarrollo software, simplificando la gestión del repositorio de recursos de seguridad y la integración de los Criterios Comunes (ISO/IEC 15408) así como de los controles de la ISO/IEC 27001 en el proceso de desarrollo software tal y como propone SREP (anteriormente presentado en TECNIMAP'2006). En definitiva, SREPTOOL viene a complementar a Métrica versión 3 y a MAGERIT versión 2 en el tratamiento automatizado de requisitos de seguridad.

1. Introducción

Hoy en día es ampliamente aceptado el principio que establece que la construcción de la seguridad en las etapas tempranas del proceso de desarrollo es más eficaz respecto a los costes y tiene como resultado diseños más robustos [12]. Por tanto, la seguridad en el software está generando cada vez mayor interés entre los ingenieros del software [23] lo cual ha ocasionado que la disciplina de la Ingeniería de Requisitos de Seguridad sea altamente considerada como parte de la Ingeniería de la Seguridad aplicada a los procesos de desarrollo de sistemas software, lo cual hasta la fecha, ha carecido de la atención necesaria [14]. La denominada Ingeniería de Requisitos de Seguridad, proporciona técnicas, métodos y normas aplicables en el ciclo de desarrollo de los SI y que implica el uso de procedimientos repetibles y sistemáticos para asegurar que el conjunto de requisitos obtenidos es completo, consistente y fácilmente comprensible y analizable por parte de los diferentes actores implicados en el desarrollo del sistema a fin de desarrollar Sistemas de Información seguros [13].

A pesar de todas estas consideraciones, todavía existen muchas organizaciones en la actualidad que tienden a prestar poca atención a los requisitos de seguridad. Uno de los motivos es la carencia de herramientas CARE (Computer-Aided Requirements Engineering) que soporten la aplicación de los métodos o metodologías o procesos de ingeniería de requisitos de seguridad, lo que suele implicar que, tal y como se describe en [2], la implantación de este tipo de procesos resulte difícil de implantar y de que tenga éxito por tener que realizarse manualmente.

Después de haber realizado un análisis comparativo de las propuestas existentes sobre requisitos de seguridad y herramientas CARE que los soporten en [17], concluimos que aunque en los últimos años se ha planteado una gran cantidad de propuestas, ninguna de las identificadas alcanzaba un nivel deseado de integración en el ciclo de desarrollo software, ni facilitaban soporte metodológico intuitivo y sistemático para la gestión de requisitos de seguridad a fin de desarrollar sistemas de información seguros y conformes a los estándares de seguridad actualmente más relevantes (como ISO/IEC 15408 [7] principalmente así como ISO/IEC 27001 [9], ISO/IEC 17799 [8] o ISO/IEC 21827 [6]) en lo relativo a la gestión de requisitos de seguridad. Con este fin y partiendo del anteriormente definido concepto de Ingeniería de Requisitos de Seguridad, propusimos el proceso SREP (Security Requirements Engineering Process) [19].

En este artículo describimos el prototipo de una herramienta de gestión de requisitos de seguridad denominada SREPTOOL, que hemos desarrollado para dar soporte automatizado a la aplicación de SREP y que viene a complementar a Métrica versión 3 y a MAGERIT versión 2 en el tratamiento automatizado de requisitos de seguridad. SREPTOOL proporcionará una forma guiada, sistemática e intuitiva para la aplicación del proceso de ingeniería de requisitos de seguridad SREP, asimismo posibilita una sencilla integración con los demás requisitos y con las distintas fases del ciclo de desarrollo, así como facilita el cumplimiento del estándar IEEE 830:1998 [4], ayudándose para ello de las funcionalidades que ofrece *'IBM Rational RequisitePro'* (herramienta CARE que extiende SREPTOOL). Además, este prototipo ayuda en que los sistemas de información desarrollados sean conformes a los estándares de seguridad anteriormente mencionados en lo relativo a la gestión de requisitos de seguridad, sin la necesidad de dominar dichos estándares y reduciendo la participación de expertos de seguridad para conseguirlo, es decir, mejora la eficiencia de SREP. Y adicionalmente, gracias al Repositorio de Recursos de Seguridad que integra SREPTOOL, se facilita la reutilización de artefactos, mejorándose por ende la calidad sucesivamente.

El resto de la comunicación esta organizada de la siguiente forma: en la sección 2, resumimos algunas características básicas de SREP con el fin de comprender la exposición posterior de la herramienta. Posteriormente, en la sección 3, describimos la funcionalidad de SREPTOOL y cómo ha sido implementada. Finalmente en la sección 4 expondremos nuestras conclusiones, junto con las aportaciones de SREPTOOL y el trabajo futuro.

2. El Proceso de Ingeniería de Requisitos de Seguridad SREP

SREP [19] es un proceso basado en activos y dirigido por el riesgo para el establecimiento de requisitos de seguridad en el desarrollo de SI seguros. Básicamente este proceso describe cómo integrar los Criterios Comunes (CC) [7] en el ciclo de desarrollo junto con el uso de un repositorio de recursos de seguridad para facilitar la reutilización de requisitos, activos, amenazas, test y contramedidas. Asimismo, facilita los distintos tipos de trazabilidad (según los conceptos de trazabilidad en [20] que se basan en [3, 13]): pre-trazabilidad y post-trazabilidad; la trazabilidad hacia atrás y hacia delante; las relaciones de trazabilidad entre requisitos y las relaciones de los requisitos con otros artefactos.

Este proceso está centrado en la construcción de conceptos de seguridad en las primeras fases del desarrollo. De manera genérica se puede describir como un ‘add-in’ de actividades (que se descomponen en tareas, donde se generan artefactos de entrada y salida, y con la participación de distintos roles) que se podrían integrar sobre el modelo actual de una Organización dándole a ésta un enfoque en ingeniería de requisitos de seguridad. En [18] describimos la integración de SREP en Métrica v.3 y en [19] se describe más detalladamente cómo SREP se integra en el ciclo de vida del Proceso Unificado [10], que como sabemos está divido en una secuencia de fases y cada fase puede implicar varias iteraciones. De esta manera, el modelo elegido por SREP es un modelo de proceso en espiral y los requisitos de seguridad y sus artefactos asociados (amenazas, etc.) evolucionan a lo largo del ciclo de vida y se tratan a la vez que los otros requisitos funcionales y no funcionales y demás artefactos del proceso de desarrollo software. Al mismo tiempo, los Componentes de los CC y los controles de la ISO/IEC 27001 se introducen en el ciclo de vida de desarrollo software, de manera que SREP usa los diferentes componentes según la fase del ciclo de vida en que se esté y la actividad de SREP correspondiente, aunque las tareas de aseguramiento de la calidad se realizan durante todas las fases y son en estas tareas donde la mayoría de los requisitos de aseguramiento de los CC se incorporan.

El Repositorio de Recursos de Seguridad (RRS) facilita el desarrollo con reutilización de requisitos, lo cual incrementa su calidad, ya que las inconsistencias, errores, ambigüedades y otros problemas se pueden detectar y corregir en proyectos sucesivos [22]. Un meta-modelo de repositorio, describiendo la organización del RRS se muestra en la Fig. 1. Se trata de un meta-modelo dirigido por activos así como por amenazas y objetivos, porque los requisitos pueden obtenerse a través de los objetivos de seguridad o de las amenazas partiendo de los activos.

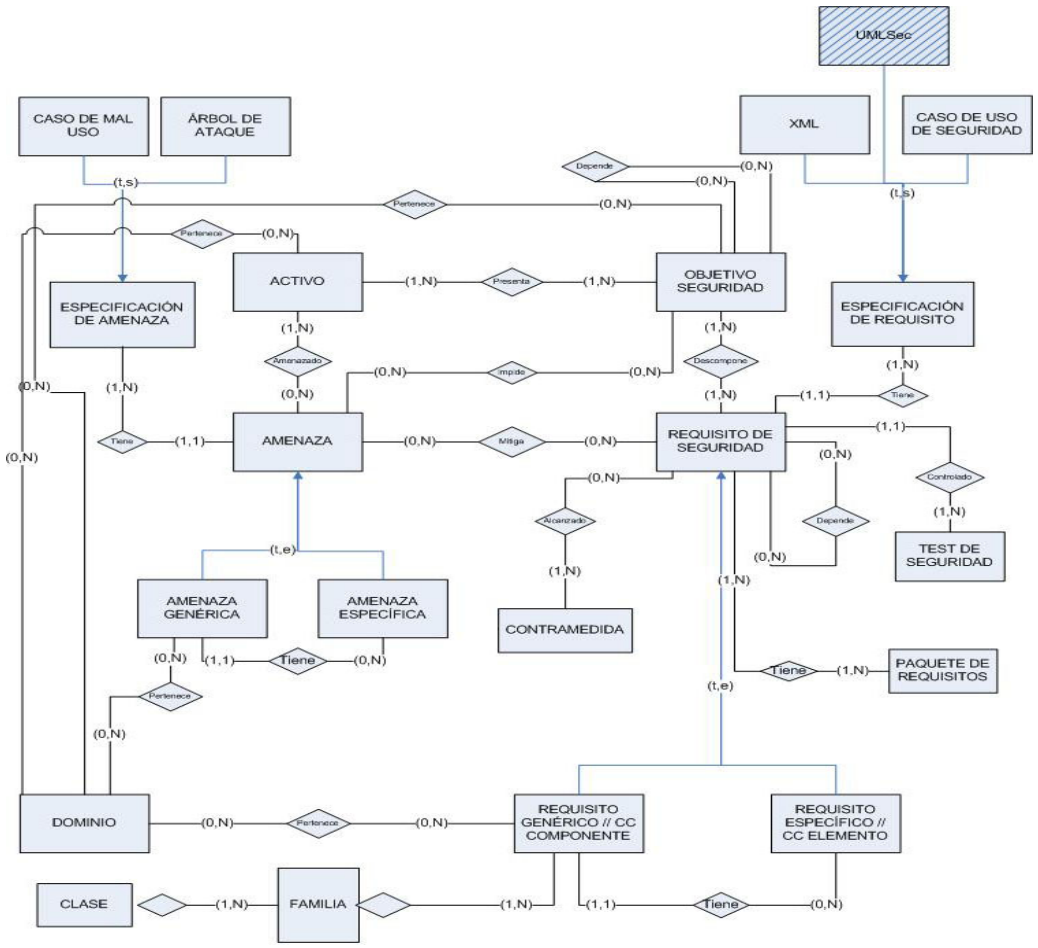

Fig. 1 Meta-modelo del Repositorio de Recursos de Seguridad

3. La Herramienta SREPTOOL

En esta sección describimos brevemente tanto la tecnología utilizada como la funcionalidad de SREPTOOL, que viene a complementar a MÉTRICA versión 3 y a su interfaz de seguridad con MAGERIT versión 2 en el tratamiento automatizado de requisitos de seguridad, y donde explicamos como SREPTOOL facilita la aplicación de SREP, automatizándolo y por tanto complementando a MÉTRICA en las primeras fases (procesos), quedándose fuera del alcance de esta comunicación una más extensa y precisa descripción tanto de SREPTOOL como de su acoplamiento con MÉTRICA en todos sus procesos en un proyecto de la Administración Pública.

3.1 Elección de herramienta CARE

Hubo que tomar la decisión inicial de desarrollar una herramienta nueva o extender una existente, y dadas las características del proceso SREP y los objetivos que pretende su aplicación, se consideró más adecuado extender una herramienta CARE existente y centrarse en ampliarla para que automatizase la gestión de requisitos de seguridad que describe SREP.

Realizamos una primera selección de herramientas que cumplían con la mayoría de las

funciones y que han demostrado con su grado de penetración en el mercado que proporcionan soluciones efectivas, tal y como se recoge en diversos estudios al respecto, como [1, 5, 21]. Nuestra lista seleccionada fue: RequisitePro, IRqA, DOORS y Caliber-RM. Finalmente, se decidió elegir extender IBM/Rational RequisitePro como soporte para nuestro prototipo, debido fundamentalmente a los siguientes factores:

- Extensibilidad. RequisitePro facilita un API basada en COM que permite acceder a los datos almacenados en éste (proyectos, requisitos, atributos, etc.) tanto para consultarlos como para modificarlos. Así como controlar la interfaz de usuario de RequisitePro y también los documentos de Microsoft Word. Lo cual, a pesar de ser algo más limitada que en las otras herramientas, nos resultaba más claro y sencillo de adaptar a las necesidades de SREPTOOL.

- Integración automatizada con el resto de actividades del ciclo de vida. RequisitePro al estar integrada en el paquete "*Rational Suite AnalystStudio*" facilitaba un aspecto clave para SREP, la integración no sólo con los otros requisitos, sino con otros artefactos del ciclo de vida (como su integración con elementos de modelado de *Rational Rose*).

- Experiencia previa e integración en el entorno corporativo de desarrollo. La herramienta RequisitePro ha sido ampliamente utilizada como herramienta de soporte en proyectos previos al desarrollo de SREPTOOL (utilizándose por ejemplo en [16]), al ser la herramienta corporativa de la Gerencia de Informática de la Seguridad Social.

- Facilidad de uso y Multiusuario. Una de sus características más destacadas es su integración con el procesador de textos Microsoft Word, así como ver todas sus funciones en una sola vista. Además proporciona la posibilidad de acceso multiusuario al proyecto y una interfaz web colaborativa.

- Trazabilidad. RequisitePro permite la creación de relaciones de trazabilidad entre distintos tipos de requisitos, y se visualiza a través de una matriz de trazabilidad.

- Otros factores destacables. RequisitePro permite cierta reutilización utilizando plantillas de documentos. Asimismo, su repositorio está basado en una base de datos relacional comercial (MS-Access, Oracle, MS-SQLServer) y ofrece control de versiones de los requisitos.

3.2 Tecnología utilizada

Para la creación del prototipo se ha utilizado una biblioteca *dll ActiveX*, que se enlaza con RequisitePro en Windows (2000, XP o Vista). De esta manera, los objetos de RequisitePro son visibles desde SREPTOOL y por otro lado, los artefactos generados por SREPTOOL son visibles desde RequisitePro. Así, la funcionalidad de nuestra herramienta esta accesible desde la ventana principal de Rational RequisitePro, a través del menú *Tools→SREPTOOL*. Para la integración con RequisitePro, SREPTOOL se ha desarrollado como un add-in de dicha herramienta CARE, para lo cual se ha utilizado la interfaz de extensibilidad de RequisitePro, en concreto el *RequisitePro Extensibility Server* (RPX) que permite acceder a los datos almacenados en RequisitePro y el *RqGUIApp library* que controla la interfaz de usuario del RequisitePro y permite también controlar los documentos de Microsoft Word en los que se recogen los requisitos de seguridad y demás elementos relacionados (amenazas, activos, etc.). Además, utiliza una base de datos Microsoft SQL Server. Por último, la arquitectura de SREPTOOL es una

arquitectura de tres capas y uno o dos niveles. Se independizan la capa de Presentación, la capa de Negocio o Dominio y la capa de Datos o Persistencia (Tres capas), y la capa de persistencia se encuentra en un Servidor de Base de Datos (dos niveles).

Asimismo, para su instalación el primer paso a realizar es ejecutar el instalador que la herramienta incluye, el cual ubicará automáticamente SREPTOOL en el ordenador del usuario. A continuación es necesario seguir los pasos especificados en el archivo LEEME.txt para poder integrar la herramienta en cada uno de los proyectos de RequisitePro. Además, su interfaz está tanto en inglés como en castellano.

3.3 Funcionalidad

El prototipo de SREPTOOL permite la aplicación del proceso SREP en el desarrollo de un proyecto dando soporte automatizado a sus nueve actividades. A continuación se explica el funcionamiento del prototipo y cómo SREPTOOL facilita la realización de cada una de estas actividades, ya que va guiando la actuación de los distintos roles y responsables, así como ayuda en la generación de la documentación necesaria.

En primer lugar se especifican los requisitos funcionales y demás requisitos no funcionales (salvo los de seguridad) para el proyecto en desarrollo a través de las funcionalidades básicas de IBM/Rational RequisitePro; posteriormente, accediendo al menú de Herramientas y a la opción de SREPTOOL se podrían elicitar los requisitos de seguridad asociados al proyecto. Dada la naturaleza iterativa del proceso SREP, no es necesario que esto sea necesariamente así, pero si los requisitos están definidos previamente desde RequisitePro, al llegar a la actividad 6 de SREP “Elicitación de Requisitos de Seguridad”, se podrán relacionar los requisitos de seguridad con los requisitos funcionales.

Una vez arrancado RequisitePro, y ya también arrancado el gestor de base de datos, el usuario deberá autenticarse y seleccionar un rol para poder arrancar el prototipo. Dependiendo del rol con el que el usuario se identifique ante el sistema, dicho usuario podrá realizar unas determinadas actividades tal y como se determina en los roles que se determinan en SREP, por ejemplo el responsable de la actividad (el ingeniero de requisitos, normalmente) junto con el asegurador de calidad validarán los artefactos en cada actividad y permitirán o no pasar a la siguiente actividad o volver a alguna anterior. Una vez autorizado, el usuario, y según sus permisos, podrá elegir entre crear un nuevo proyecto o abrir un proyecto ya existente. Esta segunda opción representará la situación en la que ya exista otra iteración de SREP guardada y el usuario desee realizar un refinamiento.

En la Fig. 2, se observa la interfaz principal de SREPTOOL. A través del menú Ver, se permite ver los distintos documentos generados. Hay una pestaña por cada actividad de SREP y tres botones en la parte inferior comunes para todas las pestañas, que permiten validar la actividad, generar el informe de la actividad o salir.

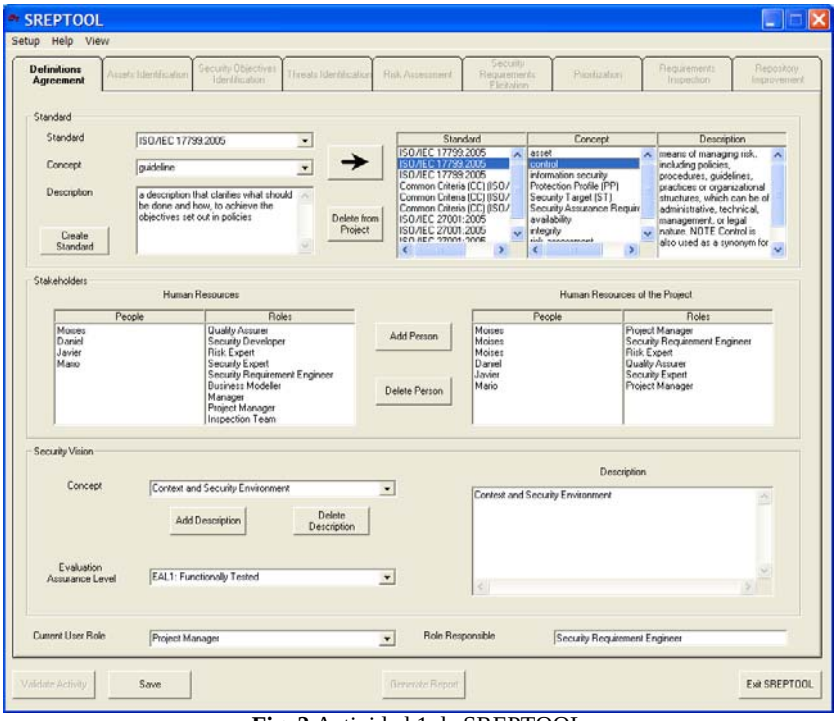

Fig. 2 Actividad 1 de SREPTOOL

Actividad 1: Acuerdo de las definiciones

Tal y como se observa en la Fig. 2, el usuario puede seleccionar los conceptos y definiciones de seguridad que desea tener en cuenta para su proyecto. Para ello tan solo tiene que seleccionar el estándar y dentro de él los conceptos que le resulten necesarios. Por otro lado, en esta actividad se seleccionan los stakeholders (partes interesadas o participantes) de entre el personal disponible, asignándole a cada uno de ellos el rol que va a desempeñar. Además en esta primera actividad el usuario puede definir el nivel de aseguramiento que desea aplicar al proyecto en desarrollo, así como recoger los artefactos de entrada de SREP en la Visión de Seguridad, como la Política de Seguridad Organizacional. Finalmente, al pulsar el botón “Generar Informe”, se crea el “Documento de Visión de Seguridad” de manera automática con los datos que el usuario ha introducido en esta actividad.

Actividad 2: Identificación de activos

En esta actividad el usuario podrá seleccionar el Dominio de seguridad al cuál pertenece el proyecto en desarrollo. En función de dicho dominio, podrá seleccionar aquellos activos que considere relevantes para su proyecto. Por otro lado, el usuario podrá crear un nuevo Dominio y agregarle nuevos activos creados o activos pertenecientes a otros dominios relacionados.

Actividad 3: Identificación de objetivos de seguridad

En esta actividad el usuario puede elegir para cada uno de los activos seleccionados en la actividad anterior, uno/s de los objetivos de seguridad de los que ese activo tenga registrados en el repositorio. Además para cada uno de los objetivos de seguridad introducidos en el proyecto, el usuario puede establecer una valoración de dichos objetivos. Por otro lado, esta actividad facilita la creación de nuevos objetivos de

seguridad por parte del usuario, así como la creación de dependencias entre dichos objetivos. El usuario puede crear un nuevo objetivo de seguridad y asociárselo tanto a alguno de los activos como al dominio de la aplicación. Finalmente, si se pulsa el botón "Generar Informe", se plasmarán todos los datos en el "Documento de Objetivos de Seguridad".

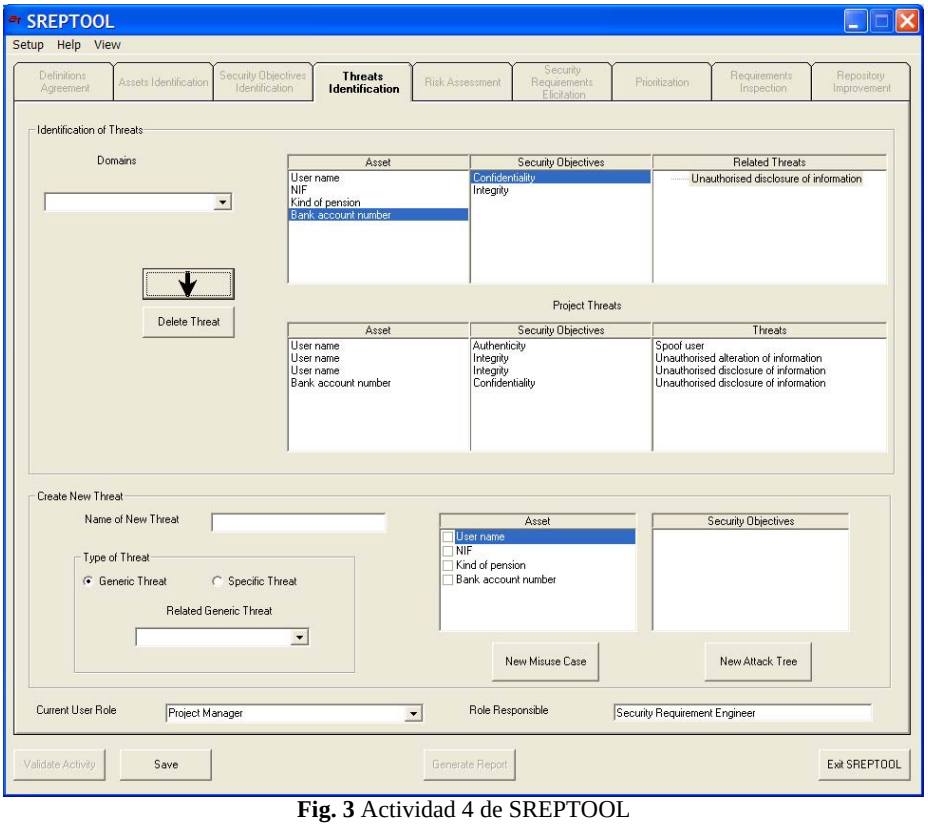

Fig. 3 Actividad 4 de SREPTOOL

Actividad 4: Identificación de amenazas

En esta actividad el usuario podrá ver cuales son las amenazas más probables (que el repositorio de la organización tiene registradas) para cada uno de los activos de su proyecto según sus objetivos de seguridad asociados. De esta manera, podrá seleccionar para su proyecto aquellas amenazas que considere de mayor relevancia. Como se observa en la Fig. 3, el usuario habría seleccionado para el activo "Número de cuenta bancaria" y el objetivo de seguridad "confidencialidad", proponiéndole SREPTOOL como amenaza única "Acceso no autorizado a la información".

Sin embargo, pudiera ser que las amenazas que el usuario quiera asociar a los activos u objetivos de su proyecto no se encontrasen en el Repositorio de Recursos de Seguridad. En este caso, el usuario podrá introducir una nueva amenaza mediante la instanciación de un "nuevo Caso de Mal Uso" o de un "nuevo Árbol de Ataque", al pinchar sobre dichos botones. De manera que rellenando una plantilla, podrá especificar: Nombre e Id del Caso de mal Uso; Casos de Uso Relacionados; Probabilidad de que se produzca la amenaza; Resumen, Precondiciones y Postcondiciones; Interacciones que se producen en el Caso de mal Uso; Tipo de amenaza (Genérica o Específica). Finalmente, si el usuario pulsa en esta actividad el botón de generación de informe, se creará el documento de "Definición del Problema de Seguridad".

Actividad 5: Valoración del riesgo

Una vez que se han identificado las amenazas en la actividad anterior, ahora el usuario puede estimar el impacto que produce la materialización de cada una de ellas. Una vez calculado el impacto que produce la amenaza, se puede estimar el riesgo aproximado de dicha amenaza teniendo en cuenta la frecuencia con que dicha amenaza se manifiesta.

Para realizar el cálculo del impacto y el riesgo, el prototipo utiliza una de las técnicas propuestas por MAGERIT v.2 [15] basada en el análisis mediante tablas. Por ejemplo, si el usuario ha calculado el impacto que tiene la amenaza "Acceso no autorizado a la información" sobre el activo "Número de cuenta bancario". Una vez que determina que la degradación que produce la amenaza es "Alta" y la valoración del objetivo de seguridad lo había determinado como "Alto", el resultado es un impacto "Muy Alto". A continuación si el usuario ha seleccionado una frecuencia para la amenaza de valor "Poco Frecuente". El resultado será un riesgo de valor "Muy Alto". Por último, en esta actividad se genera el "Documento de Valoración del Riesgo" al pulsarse el botón "Generar Informe".

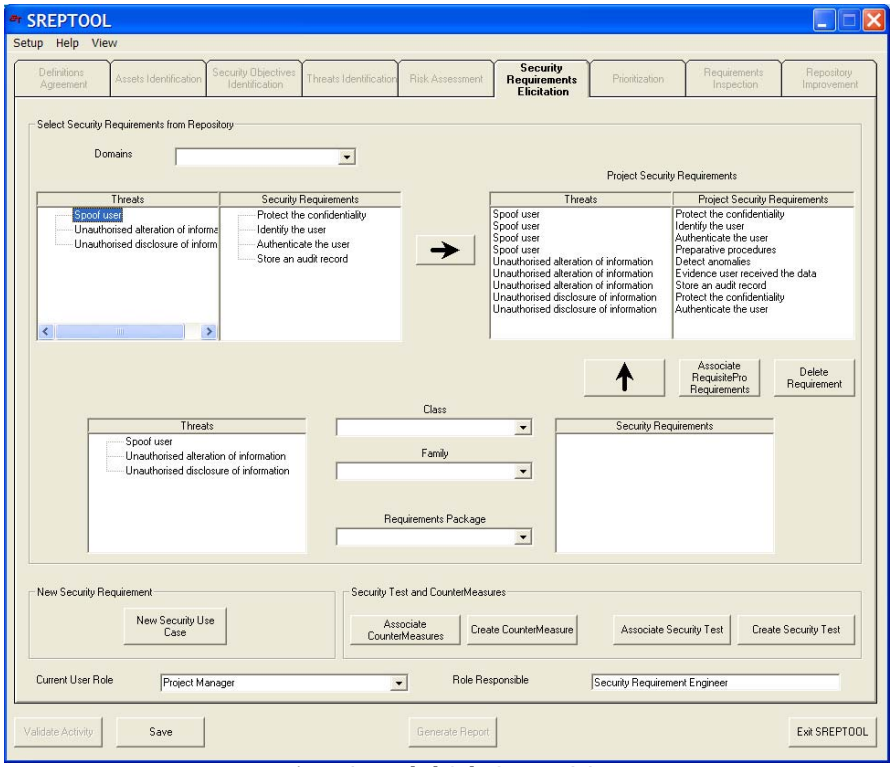

Fig. 4 Actividad 6 de SREPTOOL

Actividad 6: Elicitación de Requisitos de Seguridad

Esta actividad es una de las principales de SREPTOOL. Una vez se han seleccionado las amenazas relevantes para el proyecto, el usuario puede seleccionar aquellos requisitos de seguridad que quiere implantar. Para ello el usuario tiene tres opciones como se observa en la Fig. 4:

- Teniendo seleccionada una amenaza, el prototipo mostrará los requisitos que hay en el RRS para esa amenaza. El usuario solo tendrá que seleccionar

aquellos requisitos que considere relevantes.

- Seleccionando una clase y una de sus familias de los Criterios Comunes, el prototipo mostrará los requisitos de seguridad asociados a dicha familia. El usuario podrá seleccionar y añadir a su proyecto los requisitos deseados.
- Seleccionando uno de los paquetes de requisitos y dentro del paquete aquellos requisitos deseados.

Por otro lado, el usuario puede introducir nuevos requisitos de seguridad que no se encuentren en el repositorio mediante la plantilla que se muestra en la Fig. 5. En esta pantalla se podrá introducir: El Nombre e Id del Caso de Uso Seguro; Aquellas amenazas y objetivos con las que está relacionado el requisito que instancia el caso de uso seguro; Los requisitos de seguridad que son excluyentes respecto al requisito de seguridad que se está instanciando; El tipo de requisito de seguridad (genérico o específico) y en caso de que sea específico, a que requisito genérico pertenece; La clase, familia y paquete de requisitos a los que se quiere asociar el nuevo requisito de seguridad; Precondiciones, Postcondiciones e Interacciones del caso. Además, en esta actividad se recogerán en la herramienta las dependencias entre los requisitos de seguridad y los requisitos introducidos en el RequisitePro, también le permite asociar tanto contramedidas como test de seguridad a los requisitos de seguridad elicitados.

Además, el usuario puede seleccionar o crear Contramedidas o Test de Seguridad. Finalmente, con el botón “Generar Informe” se crea el “Documento de Especificación de Requisitos de Seguridad”.

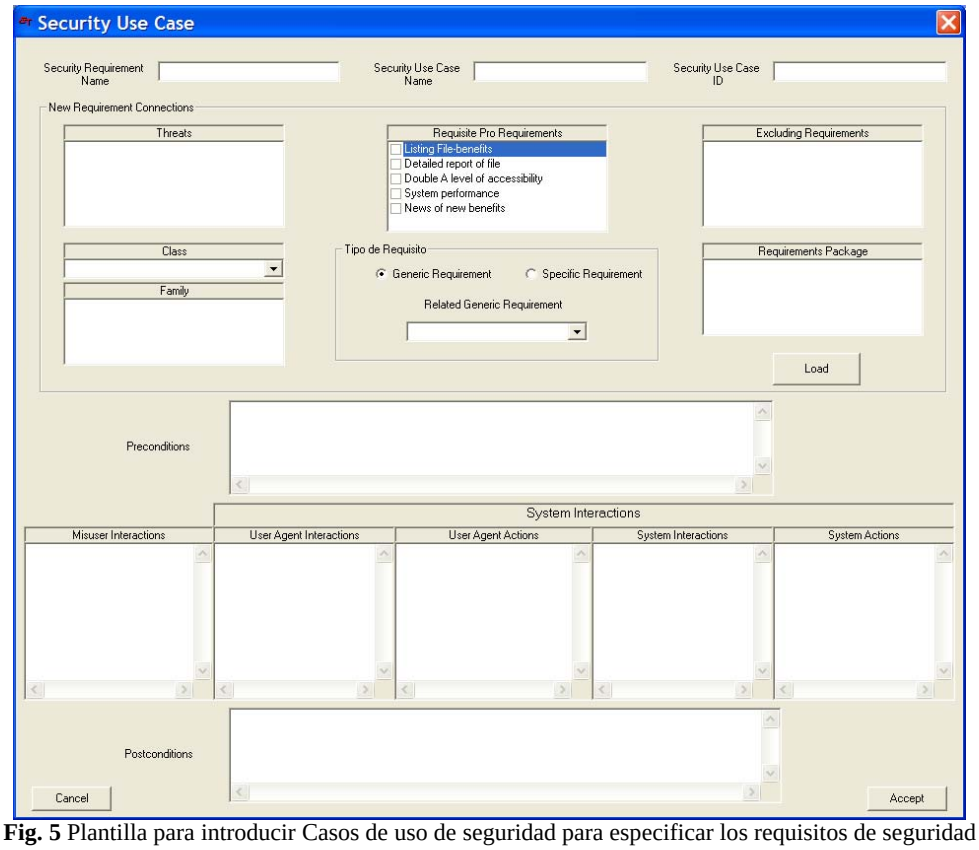

Fig. 5 Plantilla para introducir Casos de uso de seguridad para especificar los requisitos de seguridad

Actividad 7: Priorización

Esta actividad tiene por objetivo automatizar la priorización de requisitos de seguridad en función del riesgo de las amenazas que mitigan y se tendrá en cuenta las dependencias con otros requisitos. Para cada uno de los requisitos de seguridad establecidos en el proyecto, el usuario podrá seleccionar el valor de prioridad que desea asignarle (Critico, Estándar, Óptimo). Una vez seleccionadas las prioridades de todos los requisitos, al pulsar el botón priorizar se ordenarán los requisitos de mayor a menor prioridad.

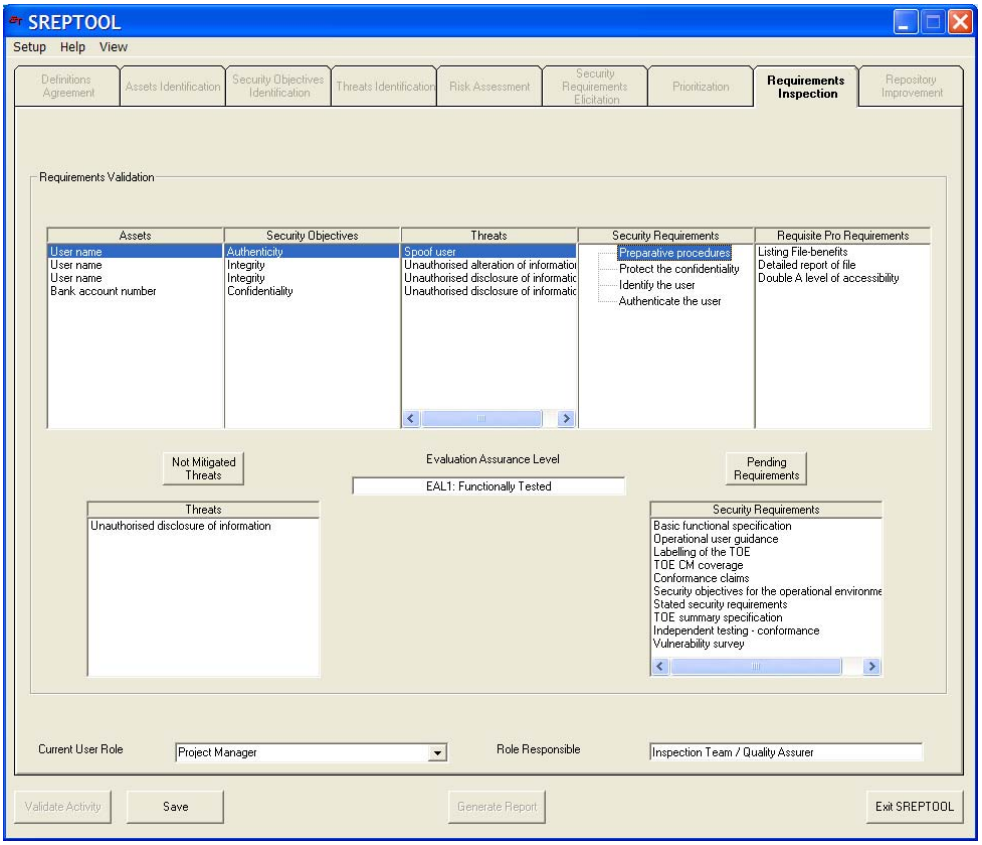

Fig. 6 Actividad 8 de SREPTOOL

Actividad 8: Inspección de requisitos

En esta actividad, el prototipo facilita al usuario la verificación y validación de los requisitos de seguridad, Para ello la aplicación facilita la trazabilidad entre activos, objetivos de seguridad, amenazas y requisitos de seguridad identificados. Además, SREPTOOL permite identificar la existencia de amenazas identificadas en la actividad 4 que no han sido mitigadas mediante ningún requisito de seguridad. Y de acuerdo al nivel de aseguramiento establecido en la primera actividad, la aplicación permite revisar los requisitos de aseguramiento pendientes. En la Fig. 6 se aprecia cómo el prototipo nos indica que para la amenaza “Revelación no autorizada de información”, el usuario no ha especificado ningún requisito de seguridad. Por otro lado, SREPTOOL muestra cuatro requisitos de aseguramiento pendientes, que de acuerdo al nivel EAL1, el usuario no había añadido a su proyecto. Por último, el prototipo permite la generación automática del “Documento de Fundamentación de los Requisitos de Seguridad”.

Actividad 9: Mejora del repositorio.

En esta última actividad, el prototipo permite añadir al repositorio (RRS) todos aquellos recursos de seguridad que el usuario haya creado durante la iteración. SREPTOOL presenta en la parte superior y clasificados por categorías, todos aquellos recursos de seguridad que han sido creados nuevos. El usuario (en este caso el equipo de inspección o el asegurador de la calidad) ha de seleccionar aquellos que considere interesantes de ser introducidos en el RRS y pulsar el botón "Añadir al Repositorio". De esta manera, en la parte inferior se mostrará el número de elementos de cada categoría que se han creado en la iteración, y cuantos de ellos han sido introducidos en el repositorio. Por último, el prototipo generará el "Documento de Declaración de Seguridad" conforme a los Criterios Comunes (ISO/IEC 15408), que contendrá toda la información de los artefactos generados por SREPTOOL en anteriores actividades.

4. Conclusiones

En nuestros días, la seguridad en el software está generando cada vez mayor interés Existen diversas herramientas CARE interesantes, algunas de ellas han sido descritas y comparadas en este trabajo, aunque presentan algunas limitaciones en lo relativo a la gestión de requisitos de seguridad expuestas anteriormente.

Por ello, el prototipo que se presenta (SREPTOOL) proporciona una forma guiada, sistemática e intuitiva para la aplicación del proceso de ingeniería de requisitos de seguridad SREP, asimismo posibilita una sencilla integración con los demás requisitos y con las distintas fases del ciclo de desarrollo, así como facilita el cumplimiento del estándar IEEE 830:1998, ayudándose para ello de las funcionalidades que ofrece '*IBM Rational RequisitePro'*. Además, este prototipo facilita que los sistemas de información desarrollados sean conformes a los estándares de seguridad actualmente más relevantes relativos a la gestión de requisitos de seguridad (como ISO/IEC 15408, ISO/IEC 27001, ISO/IEC 17799 o ISO/IEC 21827), sin la necesidad de dominar dichos estándares y reduciendo la participación de expertos de seguridad para conseguirlo. Asimismo, gracias al Repositorio de Recursos de Seguridad que integra, se facilita la reutilización de artefactos, mejorándose por ende la calidad sucesivamente.

Además, existe un conjunto de aspectos proyectados para el futuro del prototipo que permitirán aumentar el nivel de automatización de la aplicación de SREP y por tanto una mejor eficacia en el proceso de ingeniería de requisitos, entre los cuales destacamos los siguientes: extender el tipo de especificaciones de requisitos soportadas, para que soporte UMLSec [11]; extender la herramienta para que pueda ser soportada en otras herramientas CARE; automatizar la creación de los casos de uso de seguridad utilizando los casos de mal uso creados en la actividad 4 de SREP.

Agradecimientos

Esta comunicación ha sido desarrollada en el contexto del proyecto ESFINGE (TIN2006-15175-C05-05) del Ministerio de Educación y Ciencia, y de los proyectos MISTICO (PBC-06-0082) y DIMENSIONS (PBC-05-012-2) de la Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla- La Mancha y el FEDER.

Referencias Bibliográficas

1. *Atlantic_Systems, Requirements tools (http://www.volere.co.uk/tools.htm). 2006.*

2. *Davis, A., Tracing: A Simple Necessity Neglected. IEEE Software, **12**(5) (1995).*
3. *Gotel, O.C.Z. and Finkelstein, A.C.W. An analysis of the requirements traceability problem. in First International Conference on Requirements Engineering (ICRE'94). 1994: IEEE CS Press.*
4. *IEEE, IEEE 830: 1998 Recommended Practice for Software Requirements Specifications. 1998.*
5. *INCOSE, The International Council on Systems Engineering Requirements Management Tools Survey (http://www.incose.org). 2006.*
6. *ISO/IEC, ISO/IEC 21827:2002 Information technology -- Systems Security Engineering -- Capability Maturity Model (SSE-CMM). 2002.*
7. *ISO/IEC, ISO/IEC 15408:2005 Information technology - Security techniques - Evaluation criteria for IT security, (Common Criteria v3.0). 2005.*
8. *ISO/IEC, ISO/IEC 17799 Information technology - Security techniques - Code of practice for information security management. 2005.*
9. *ISO/IEC, ISO/IEC 27001:2005 Information technology -- Security techniques -- Information security management systems -- Requirements. 2005.*
10. *Jacobson, I., Booch, G., and Rumbaugh, J., The Unified Software Development Process. 1999, Boston: Addison-Wesley Longman Publishing Co.*
11. *Jürjens, J., UMLsec: extending UML for secure systems development. UML 2002 - The Unified Modeling Language. Model Engineering, Languages,Concepts, and Tools. 5th International Conference., 2002. **LNCS 2460**: p. 412-425.*
12. *Kim., H.-K., Automatic Translation Form Requirements Model into Use Cases Modeling on UML. ICCSA 2005, LNCS, 2005: p. 769-777.*
13. *Kotonya, G. and Sommerville, I., Requirements Engineering Process and Techniques. Hardcover ed. 1998, UK: John Willey & Sons. 294.*
14. *Lamsweerde, A.v. Elaborating Security Requirements by Construction of Intentional Anti-Models. in 26th International Conference on Software Engineering. 2004. Edinburgh: ACM-IEEE.*
15. *López, F., Amutio, M.A., Candau, J., and Mañas, J.A., Methodology for Information Systems Risk Analysis and Management. 2005: Ministry of Public Administration.*
16. *Mellado, D., Fernández-Medina, E., and Piattini, M., Applying a Security Requirements Engineering Process. 11th European Symposium on Research in Computer Security (ESORICS 2006), 2006. **Springer LNCS 4189**: p. 192-206.*
17. *Mellado, D., Fernández-Medina, E., and Piattini, M., A Comparative Study of Proposals for Establishing Security Requirements for the Development of Secure Information Systems. The 2006 International Conference on Computational Science and its Applications (ICCSA 2006), Springer LNCS 3982, 2006. **3**: p. 1044-1053.*
18. *Mellado, D., Fernández-Medina, E., and Piattini, M., Complementando a Métrica: Proceso de Ingeniería de Requisitos de Seguridad para el Desarrollo de Sistemas de Información Seguros. IX Jornadas sobre Tecnologías de la Información para la Modernización de las Administraciones Públicas (TECNIMAP 2006), 2006: p. Tema 3, comunicación sexta (nº 48).*
19. *Mellado, D., Fernández-Medina, E., and Piattini, M., A Common Criteria Based Security Requirements Engineering Process for the Development of Secure Information Systems. Computer Standards and Interfaces, 2007. **29**(2): p. 244 - 253.*
20. *Pinheiro, F.A., Requirements Traceability, in Perspectives on software requirements, Sampaio, J.C. and Horacio, J., Editors. 2004.*
21. *TCP, Comparative Study Between Requirements Management and Engineering Tools (http://www.irqaonline.com/downloadarea/documents.htm). 2004.*
22. *Toval, A., Nicolás, J., Moros, B., and García, F., Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach. Requirements Engineering, **6**(4) (2002). p. 205-219.*
23. *Viega, J. and McGraw, G., Building Secure Software: How to Avoid Security Problems the Right Way. 2002, Boston: Addison-Wesley.*