

SOPORTE AUTOMATIZADO A LA REUTILIZACIÓN DE REQUISITOS

Joaquín Lasheras, Ambrosio Toval, Joaquín Nicolás, Begoña Moros

Grupo de Investigación de Ingeniería del Software. Departamento de Informática y Sistemas. Universidad de Murcia. Campus de Espinardo. 30071. Murcia
jolave@ono.com, {atoval, jnr, bmosos}@um.es

Resumen. La Ingeniería de Requisitos es una disciplina en auge, que ha demostrado su capacidad de influencia en la mejora de la productividad y calidad en los procesos y productos software. Para facilitar su aplicación, es necesario disponer de herramientas CARE (*Computer-Aided Requirements Engineering*) que den un buen soporte a sus distintas actividades. Una de estas actividades, que se ha revelado como una de las más efectivas, es la reutilización de requisitos, para la que, sin embargo la mayoría de herramientas CARE actuales no ofrecen ayuda suficiente. En este artículo presentamos un análisis y una comparativa de algunas de las herramientas CARE más populares, en la búsqueda de soluciones para el soporte automatizado a la reutilización de requisitos. Se define un conjunto de necesidades o características específicas para tratar de forma sistemática con esta actividad y, finalmente, se opta como mejor solución por el desarrollo de un “*add-in*”, un programa integrado con una de las herramientas analizadas. De este modo se consiguen los beneficios ofrecidos por una herramienta convencional de gestión de requisitos, a la vez que se enriquece ésta con las funciones específicas requeridas para la reutilización. La herramienta resultante, denominada *SirenTool* puede ser utilizada como soporte al método de Ingeniería de Requisitos SIREN (“*Simple REuse of software requiremeNts*”). SIREN proporciona un enfoque práctico para obtener y especificar los requisitos de un sistema software, basado en la reutilización de requisitos y en estándares de Ingeniería del Software. Finalmente, mostramos la funcionalidad de la herramienta utilizando los catálogos de requisitos disponibles en SIREN.

Palabras Clave. Ingeniería de Requisitos; Reutilización de Requisitos; CARE.

1. Introducción

A pesar de los recientes avances en relación con las Tecnologías de la Información y de las Comunicaciones, la realidad es que muchos proyectos de desarrollo de software no acaban dando los resultados esperados, se retrasan enormemente, desbordan sus presupuestos iniciales, o simplemente, finalizan plagados de errores. Multitud de artículos en prensa, revistas especializadas e incluso libros sobre el tema, nos muestran esta situación de forma casi cotidiana. En muchos casos, nos encontramos con que la causa es una gestión inapropiada de la especificación de los requisitos del nuevo sistema [4,5,13].

La Ingeniería de Requisitos proporciona técnicas, métodos y normas para abordar esta tarea crítica en el ciclo de desarrollo de sistemas de información. La Ingeniería de Requisitos contempla las actividades de descubrimiento, obtención, representación,

análisis, documentación y mantenimiento de un conjunto de requisitos para un sistema basado en computador. El uso del término “Ingeniería” debería implicar, como se afirma en [9], 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.

Para aprovechar además los beneficios de la reutilización, desde nuestro grupo propusimos el método SIREN [17,20] (*Simple REuse of RequiremeNts*), un enfoque práctico para obtener y especificar los requisitos de un sistema software, basado en la reutilización de requisitos y en estándares de Ingeniería del Software. SIREN incluye un *modelo de proceso en espiral*, unas *plantillas de documentos de requisitos* y un *repositorio* de requisitos reutilizables que se encuentra organizado por *catálogos* y de los cuales actualmente se encuentran disponibles uno relacionado con la seguridad [16,17] y otro con la LOPD, la Ley Orgánica de Protección de Datos personales [21,22].

Es muy ventajoso contar con herramientas *CARE (Computer-Aided Requirements Engineering)* que soporten la aplicación de los métodos y técnicas de Ingeniería de Requisitos. Estas herramientas permiten la definición, organización y gestión de requisitos, facilitando la puesta en práctica del proceso de Ingeniería de Requisitos, o simplemente haciéndolo posible [14]. Hemos detectado, sin embargo, que una de las actividades más efectivas de la Ingeniería de Requisitos, la reutilización de requisitos, no está contemplada de manera explícita en la mayoría de herramientas CARE actuales.

En este trabajo mostramos una herramienta de gestión de requisitos, que denominamos *SirenTool*, que hemos desarrollado para dar soporte automatizado a la reutilización de requisitos con SIREN. Para construir esta herramienta en primer lugar hemos obtenido una visión del estado del arte en el campo de las herramientas CARE, elaborando una comparativa de las herramientas que consideramos más interesantes para nuestros objetivos. Entre las distintas opciones consideradas, i) utilizar una herramienta de las disponibles en el mercado, ii) adaptar una de estas herramientas o bien iii) construir una herramienta completamente nueva, finalmente decidimos desarrollar *SirenTool* extendiendo la funcionalidad de una herramienta CARE comercial, *Rational RequisitePro*. Para ello construimos *SirenTool* a partir de un análisis de requisitos previamente realizado para especificar los requisitos de una herramienta CARE de soporte a SIREN.

Este artículo se estructura de la siguiente manera: en la Sección 2 proporcionamos una visión general del método SIREN. En la Sección 3 resumimos el estudio realizado sobre el estado del arte en las herramientas CARE. En la Sección 4 mostramos las necesidades específicas asociadas con la reutilización de requisitos y las funcionalidades de la nueva herramienta y, finalmente, en la Sección 5 terminamos con las conclusiones y vías futuras.

2. El Método de Reutilización de Requisitos SIREN

La propuesta SIREN [17,20] se basa en la utilización de un modelo de proceso en espiral, unas plantillas de documentos de requisitos y un repositorio de requisitos

reutilizables, que se encuentra organizado por catálogos. Coincidimos con Robertson & Robertson [12] en que si partimos de un conjunto de requisitos que ya han sido especificados para otros proyectos y/o dominios, podremos mejorar la precisión y completitud de las especificaciones de requisitos del proyecto actual, reduciendo, además, el tiempo de producir dicha especificación. En este apartado resumimos algunas características básicas de SIREN con vistas a comprender la exposición posterior.

Los catálogos de requisitos de SIREN se corresponden con lo que denominamos *perfiles* (o dominios de aplicación “horizontales”, como por ejemplo seguridad, LOPD, etc.) y *dominios* (dominios de aplicación “verticales”, tales como seguros, banca, etc.). Estos catálogos se componen de una jerarquía de documentos de especificación, estructurados de acuerdo a estándares del IEEE [7,8]. Actualmente se ha definido un catálogo de seguridad, con los requisitos procedentes de MAGERIT, la metodología de análisis y gestión de riesgos del Ministerio de Administraciones Públicas [17,20], y otro relacionado con la LOPD [21,22].

Todo requisito en SIREN tiene un conjunto mínimo, común, de atributos [18], aunque dependiendo del catálogo puede tener definidos atributos adicionales. Uno de los atributos de los requisitos son las relaciones de *trazabilidad exclusiva*, que implican que sólo uno de los requisitos que mantengan esta relación podrá ser incluido en un mismo proyecto. Un ejemplo de dos requisitos exclusivos extraído del catálogo de la LOPD sería el siguiente:

“La configuración del firewall será screened host”, es exclusivo al requisito “La configuración del firewall será screened subnet”.

Las relaciones de *trazabilidad inclusiva*, por otra parte, serían aquellas relaciones según las cuales un requisito depende de otro, siendo recomendable que éste no sea incluido hasta que no esté incluido aquel del que depende. Las *relaciones padre e hijo* entre requisitos, por último, son relaciones en las que los requisitos hijos refinan al requisito padre y su definición esta siempre a continuación del requisito padre y en el mismo documento.

En SIREN tenemos *requisitos parametrizados*, que son aquellos que contienen alguna parte que se tiene que adaptar a cada aplicación o sistema, teniendo que ser instanciados en el momento de la reutilización. Un ejemplo extraído del catálogo de la LOPD sería el siguiente:

El responsable del fichero elegirá un [soporte físico] para la realización de copias de seguridad.

El modelo de proceso de SIREN es básicamente una adaptación del modelo de proceso en espiral propuesto en [9], al que hemos añadido las siguientes actividades (ver Fig. 1):

- *Selección de Requisitos.* Se proporcionan las plantillas de los documentos de especificación rellenas con requisitos del repositorio. En los dominios y perfiles relevantes para el proyecto actual, el analista junto con el resto de

*stakeholders*¹ seleccionarán los requisitos del repositorio que sean adecuados para la aplicación concreta que se está desarrollando. Nótese que cuando se reutiliza un requisito también se seleccionarán todos los requisitos relacionados con él mediante dependencias inclusivas. Además, se deberá asegurar que no se incluya en la especificación dos requisitos que sean exclusivos.

- *Obtención de requisitos específicos.* De las reuniones entre los analistas y el resto de *stakeholders* se obtienen los requisitos informales y específicos del sistema actual. Estos requisitos no están inicialmente en el repositorio.
- *Mejora del Repositorio.* El repositorio no se debe considerar como un producto final, estático, sino como un producto en continua evolución, incluyendo nuevos requisitos reutilizables y mejorando la calidad de los existentes.

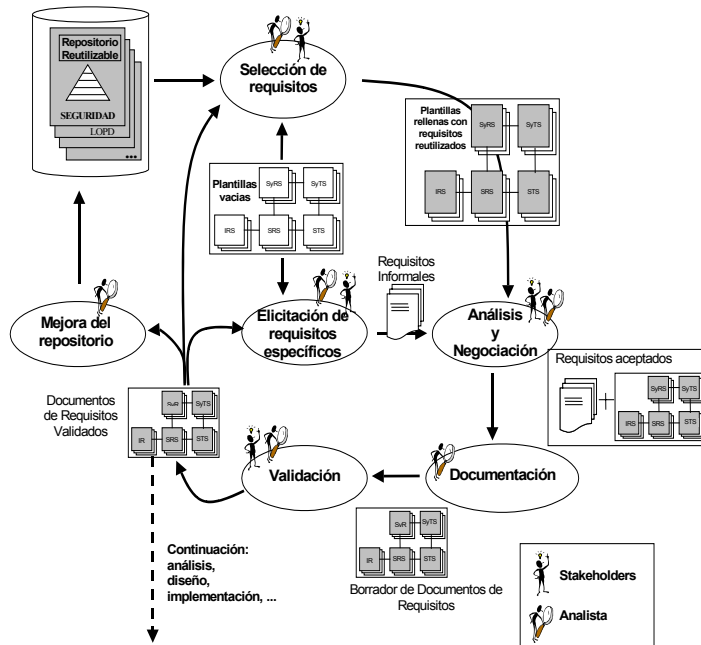


Fig. 1. Modelo de proceso de SIREN para la reutilización de requisitos.

3. Análisis de Herramientas CARE

3.1. Marco de comparación

Un paso importante en la realización de la herramienta consistía en analizar la situación actual con respecto a las herramientas CARE y la reutilización. En primer

¹ Son todas aquellas personas que son afectadas por el sistema, las cuáles tienen una influencia directa o indirecta con los requisitos del sistema.

lugar, surge la necesidad de establecer un marco para realizar una comparativa entre herramientas. Para definir este marco tuvimos en cuenta dos aspectos: (i) las necesidades generales de una herramienta de gestión de requisitos, para lo cual nos basamos sobre todo en la encuesta sobre herramientas de gestión de requisitos realizada por INCOSE [6], y (ii) las necesidades específicas de reutilización y del método SIREN.

Tras una primera selección elegimos para la comparativa las herramientas Caliber-RM v. 4.0 [2], DOORS v. 5.2 [3] y RequisitePro v. 7.0 [11]. Estas herramientas se eligieron puesto que, además de presentar características deseables para nuestros objetivos, están ampliamente difundidas (sobre todo RequisitePro y DOORS, aunque Caliber-RM está extendiéndose rápidamente) y son muy reconocidas, como demuestra el hecho de que aparecían siempre en todas las comparativas que estudiamos, como por ejemplo en [1,6,15]. Además tienen un amplio soporte de las empresas que las desarrollan, y lo que es más importante tienen la posibilidad de ampliar la funcionalidad. Las características más destacadas de estas herramientas son las siguientes:

- RequisitePro es una herramienta centrada en documentos, que almacena los requisitos asociándolos a documentos (aunque también permite guardarlos directamente en la base de datos), mientras que las otras dos herramientas están orientadas a requisitos.
- DOORS, a diferencia del resto de las herramientas, considera los requisitos como objetos y los documentos como módulos. Tiene pues una orientación basada en objetos, frente a RequisitePro y Caliber-RM, que manejan solamente requisitos y sus atributos.
- Caliber-RM era de las tres la que presentaba unas interfaces más prácticas y amigables y está enfocada sobre todo al trabajo en web.

3.2. Cuadro resumen de la comparativa

En la Tabla 1 sintetizamos las características que hemos considerado necesarias para nuestro proceso SIREN. El análisis completo realizado para las tres herramientas en estudio se puede ver en los anexos de [18].

3.3. Conclusiones del Análisis y Decisiones de Diseño

Las tres herramientas elegidas proporcionan casi todas las necesidades básicas exigibles a una herramienta de gestión de requisitos para que sea incorporada por las empresas. Sin embargo, el estudio realizado revela que ninguna de ellas cubre todas las necesidades identificadas para dar soporte automatizado a la reutilización de requisitos. Las limitaciones que presentan en aspectos que son fundamentales, por ejemplo, en SIREN, hacen que no podamos elegir una sin considerar previamente la capacidad de extensibilidad que ofrecen. Es común a las tres herramientas el hecho de no tener la capacidad para definir e instanciar requisitos parametrizados, ni tampoco el soporte para mantener requisitos exclusivos. Por otra parte, en este estudio pudimos apreciar que la reutilización de requisitos, de forma expresa, no es soportada por ninguna de ellas. La única manera de reutilizar es mediante las funcionalidades del

menú de Edición de las herramientas, esto es, “copiando y pegando” requisitos definidos en otros proyectos, con lo que no se tienen en cuenta las relaciones de traza automáticamente. Las herramientas CARE estudiadas no proporcionan utilidades para definir criterios de búsqueda y poder seleccionar del resultado aquellos requisitos que sean adecuados para el proyecto actual.

CARACTERÍSTICAS NECESARIAS	HERRAMIENTAS ESTUDIADAS		
	CALIBER-RM V.4	DOORS V.5.2	REQUISITEPRO V2002
<i>Importación de requisitos</i>	Sí, Word y ficheros delimitados.	Sí, Word y ficheros delimitados.	Sí, Word y CSV.
<i>Clasificación de requisitos</i>	Sí, se definen tipos de requisitos	Sí, se definen objetos (requisitos como objetos).	Sí, se definen tipos de requisitos
<i>Req. parametrizados</i>	No	No	No
<i>Soporte selección y reutilización de requisitos</i>	No. Sólo copiar y pegar dentro y entre proyectos.	No. Sólo copiar y pegar módulos.	No, sólo copiar y pegar dentro del mismo proyecto.
<i>Asignación de propiedades de rendimiento, etc.</i>	Sí, atributos asociados a tipos de requisitos.	Sí, por medio de atributos asociados a objetos.	Sí, por medio de atributos asociados a tipos de requisitos.
<i>Creación de nuevos tipos de requisitos y atributos</i>	Sí	a nivel de módulos	Sí
<i>Identificación de inconsistencias</i>	Sí, de forma gráfica con matriz de trazabilidad.	Sí, de forma gráfica con matriz de trazabilidad.	Sí, de forma gráfica con matriz de trazabilidad.
<i>Visibilidad de enlaces de trazabilidad</i>	Sí, de forma gráfica. Por tipos de requisitos y para un requisito sólo.	Sí, de forma gráfica, con todos los requisitos.	Sí, de forma gráfica, por tipos de requisitos.
<i>Requisitos exclusivos</i>	No	No	No
<i>Historia de los cambios de los requisitos</i>	Sí, automáticamente.	Sí, automáticamente.	Sí, automáticamente.
<i>Control de versiones/líneas base</i>	Sí, pudiendo compararlas.	Sí, pudiendo compararlas.	Sí, pero no comparables.
<i>Control de acceso</i>	Sí, grupos y usuarios.	Sí, grupos y usuarios.	Sí, grupos y usuarios.
<i>Especificación estándar de salida al formato de documentos SIREN</i>	Sí, la salida se puede adaptar a diferentes formatos	Sí, la salida se puede adaptar a diferentes formatos	Sí, puesto que trabaja con documentos similares
<i>Posibilidad de ampliación de funcionalidad</i>	Sí, API basado en componentes COM y JAVA.	Sí, API con lenguaje de extensión (DXL).	Sí, API basado en componentes COM.
<i>Práctica y sencilla de utilizar. Interfaz amigable</i>	Sí, lo mejor su presentación de los atributos de los requisitos.	No, difícil de seguir, organizada por módulos y objetos.	Sí, requisitos, vistas y documentos integrados en una sola vista.

Tabla 1. Resumen de la comparativa de las tres herramientas en estudio.

RequisitePro permite la reutilización utilizando plantillas de documentos, al igual que la herramienta REM [10], que soporta reutilización utilizando cualquier documento como “documento base”. Sin embargo, en SIREN buscamos una reutilización con distintos grados de granularidad, permitiendo al usuario desde reutilizar un solo requisito procedente de alguno de los catálogos disponibles a reutilizar catálogos completos.

En cuanto a la definición de las plantillas de documentos de requisitos propuestas en el método SIREN, Caliber-RM y DOORS no trabajan directamente sobre documentos, aunque se puede establecer el formato para la salida. En este punto RequisitePro difiere de ellas puesto que trabaja directamente sobre documentos, a los cuáles se les asocian los requisitos, cuyas plantillas iniciales pueden ser definidas por el usuario. De esta manera, RequisitePro se adapta al modo de trabajo propuesto en el modelo de proceso de SIREN

Se decidió elegir RequisitePro como soporte para nuestra herramienta, ampliando la funcionalidad para dotarla de las capacidades de las que carece, frente a la opción de empezar a diseñar una nueva herramienta desde el principio. A la hora de elegir RequisitePro tuvimos en cuenta los siguientes factores:

- Extensibilidad. Aunque las tres herramientas disponen de un API mediante el cual su funcionalidad puede ser ampliada, el API de RequisitePro, a pesar de ser algo más limitado que el de las otras dos, nos resultaba más claro y sencillo de utilizar, sin dejar por ello de ser menos eficiente que las otras opciones, para las funcionalidades a implementar.
- Experiencia previa. La herramienta RequisitePro ha sido utilizada como herramienta de soporte en los trabajos previos del desarrollo de SIREN [17,19,21].
- Integración con otras herramientas. RequisitePro está integrada con las herramientas del paquete “*Rational Suite AnalystStudio*”, lo que puede resultar muy interesante para añadir futuras características a SirenTool.
- Facilidad de uso. Tanto Caliber-RM como RequisitePro son dos herramientas muy prácticas y sencillas de utilizar. Una de las características que hacía más atractiva la herramienta RequisitePro es la integración de todas sus funciones en una sola vista y el uso del procesador de texto Microsoft Word.

4. La Herramienta SirenTool

Después del estudio de las herramientas comerciales se elaboró la especificación de requisitos (SRS) de la herramienta CARE de soporte a SIREN [18]. A partir del SRS se desarrolló un prototipo de *SirenTool*, que es el nombre que le damos a la solución completa, como resultado de la integración de las necesidades propias del proceso SIREN en la herramienta RequisitePro, gracias a su capacidad de extensibilidad. Las funcionalidades añadidas están disponibles a través de una interfaz programada con VisualBasic a la que denominamos “SIRENA” (*SIREN Automated*). En RequisitePro esta interfaz aparecerá como una opción nueva en el menú *Tools*.

En la Sección 4.1 presentamos brevemente la capacidad de extensibilidad (“Extensibility Interface”, RPX) de RequisitePro, mientras que en la Sección 4.2 se muestra la funcionalidad básica de SirenTool.

4.1. La Interfaz de Extensibilidad de RequisitePro

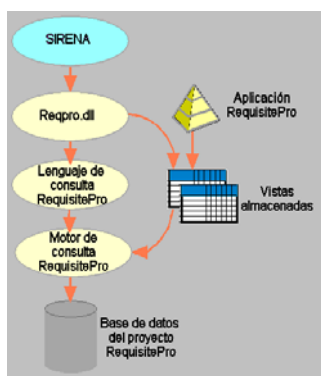


Fig. 2. Interfaz RPX

La interfaz de extensibilidad (*Extensibility Interface*) de RequisitePro es una biblioteca basada en el modelo de componentes de objetos (COM) que permite acceder a los datos almacenados en RequisitePro (proyectos, requisitos, atributos, etc.) tanto para consultarlos como para modificarlos.

La interfaz RPX incluye (Fig. 2): la biblioteca de objetos (*repro.dll*) para el acceso a los datos de un proyecto creado en RequisitePro, el lenguaje de consultas RequisitePro y un motor de consultas, que construye sentencias SQL (a partir de las consultas expresadas en el lenguaje propio de RequisitePro) y las ejecuta sobre la base de datos.

4.2. Funcionalidad de la herramienta

La nueva interfaz se abre desde la ventana principal de RequisitePro, a través del menú *Tools* y aparece con el nombre de *SIRENA* (Fig.3). La ventana principal de la aplicación (Fig. 3) permite hacer las búsquedas (según diversos criterios) de los requisitos a reutilizar procedentes de los catálogos definidos en SIREN. A continuación se muestra, en otra ventana, el resultado de la búsqueda donde se pueden seleccionar los requisitos que deben formar parte del proyecto actual. Cada uno de los requisitos seleccionados irá apareciendo en una tercera interfaz (Fig. 4) en la que se muestran sus propiedades y desde donde se insertarán en el proyecto actual.

La forma de utilizar la herramienta sería la siguiente: en primer lugar se abre en RequisitePro el proyecto denominado SIREN, que contiene el repositorio de requisitos reutilizables. Más concretamente, contiene una carpeta por cada uno de los catálogos definidos (actualmente Seguridad y LOPD). Posteriormente, accediendo al menú SIRENA, se abre la ventana de búsqueda que facilitará la selección de requisitos. Al final tendremos los requisitos reutilizados disponibles en el proyecto designado por el usuario, que se trata como un proyecto ordinario en RequisitePro.

4.2.1. Interfaz para las Búsquedas de Requisitos a Reutilizar

A través de esta interfaz podemos realizar búsquedas de requisitos atendiendo al contenido del texto del requisito (por ejemplo todos los requisitos que contienen la palabra "firewall", Fig. 3) o bien al valor de alguno de sus atributos.

En primer lugar debemos seleccionar el tipo de requisito que queremos buscar. Hasta el momento los tipos definidos son: *SRSS*, *SRS�*, *SyRSS* y *SyRS�*, que se refieren a los requisitos contenidos en los documentos SRS ("*Software Requirements Specification*") y SyRS ("*System Requirements Specification*"), correspondientes a los catálogos de seguridad (S) y LOPD (L). También existe la opción de buscar entre todos los requisitos definidos en el repositorio de SIREN.

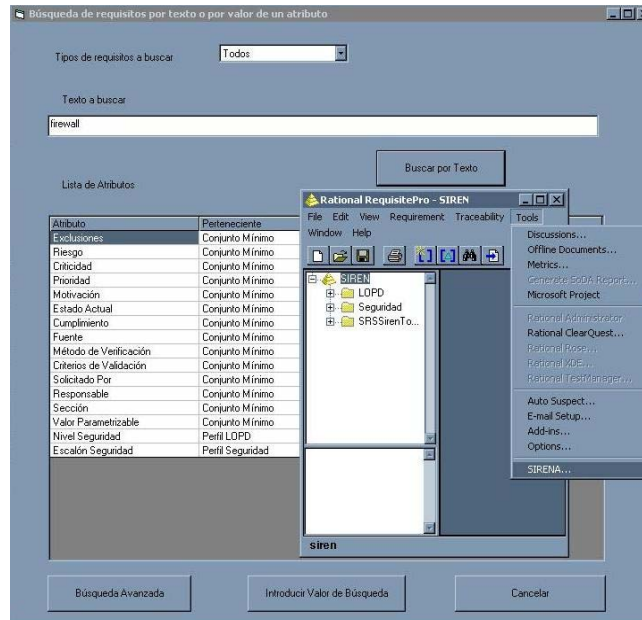


Fig. 3. Interfaz para las búsquedas

Otras posibles búsquedas pueden ser según la sección del documento en la que se encuentran los requisitos (atributo *sección*) o, para perfiles específicos como la LOPD, podemos hacer búsquedas de aquellos requisitos que cumplen con uno de los niveles de seguridad que marca la ley (*básico, medio o alto*).

4.2.2. Interfaz del Resultado de la Búsqueda

Una vez introducido el criterio de búsqueda se nos mostrará una nueva interfaz con el resultado de la misma. Para cada requisito perteneciente al resultado aparecerá: el texto del requisito, su identificador único (“*key*”) en RequisitePro y el tipo (ej. SRSS). Desde esta interfaz podemos seleccionar aquel que queramos reutilizar y, a continuación pasamos a la *Interfaz para la Reutilización del Requisito*.

4.2.3. Interfaz para la Reutilización del Requisito

Una vez que hemos seleccionado el requisito a reutilizar se nos mostrará una interfaz como la de la Fig. 4, donde se pueden ver todos los atributos del requisito, su texto, identificador único, el tipo, las relaciones de trazabilidad inclusivas y exclusivas, y los requisitos hijos.

A la hora de reutilizar el requisito hay que tener en cuenta que se ha procurado que el proceso sea lo más flexible posible, de manera que el usuario pueda modificar cualquier valor que considere oportuno para adaptarlo al nuevo proyecto, desde el texto del requisito al tipo (que por defecto es el mismo). Como el tipo del requisito influye en el conjunto de atributos, si al reutilizar el requisito se mantiene el tipo, entonces el requisito reutilizado tendrá los mismos atributos que el origen, si son distintos sólo se copiarán los atributos del conjunto mínimo.

Fig. 4. Interfaz para la reutilización del requisito.

Por otro lado, si un usuario intenta añadir un requisito que tiene relaciones de *traza inclusiva* con otro, y éste no ha sido reutilizado todavía, se avisará al usuario para que, o bien no considere la relación de trazabilidad (Botón “*No considerar traza*”) o bien, inserte previamente este nuevo requisito (Botón “*Añadir Traza*”). En este segundo caso, se presentarán los detalles del requisito con el que tiene la relación de traza en una interfaz igual que la descrita en la Fig. 4. Si este último a su vez tuviera otras relaciones de trazabilidad sucedería lo mismo y así sucesivamente hasta que hayamos recorrido todos los requisitos con los que están relacionados.

Por otra parte, desde esta interfaz de reutilización se mantienen también las *relaciones de trazabilidad exclusiva*, de tal manera que si se intenta reutilizar un requisito que presenta una relación de exclusividad con otro ya reutilizado se muestra la incidencia al usuario.

En cuanto a las *relaciones hijo* existen dos opciones: reutilizar el requisito considerando a sus hijos (botón “*Reutilizar requisito con hijos*”) o sin considerarlos (botón “*Reutilizar requisito sin hijos*”). En el primer caso se añadirá el requisito padre y después se mostrarán uno a uno cada uno de los requisitos hijos (Fig. 4) para ir reutilizándolos. Además podemos no considerar alguna de las relaciones hijo (botón “*No considerar hijo*”) y de esta forma sólo se nos mostrarán los hijos que sean interesantes para el proyecto actual.

En el caso de que el *requisito sea parametrizable* (como en la Fig. 4), tenemos la posibilidad de instanciarlo antes de añadirlo al proyecto actual. Si ese es el caso, pulsando sobre el botón “*Instanciar Requisito*” se mostrarían la lista de las partes parametrizables del requisito con el tipo asociado (Fig. 5). Los tipos habrán sido declarados previamente al introducir los requisitos en los catálogos y tendrán unos valores predefinidos entre los que podrá elegir el usuario.

Por último, el valor del atributo *fuentes* del requisito que estamos reutilizando será el identificador del requisito origen del cual ha sido reutilizado, lo cual permitirá a cualquier usuario en un determinado momento ver cual era el requisito de procedencia

y si se modificó algún valor a la hora de la reutilización. Además permitirá controlar que no se reutilice dos veces el mismo requisito en un mismo proyecto.

Una vez reutilizado el requisito, este se incluirá en el proyecto designado por el usuario, desde donde éste podrá incluirlo en el documento que estime oportuno, modificarlo, eliminarlo o bien dejarlo en la base de datos hasta próximas revisiones.

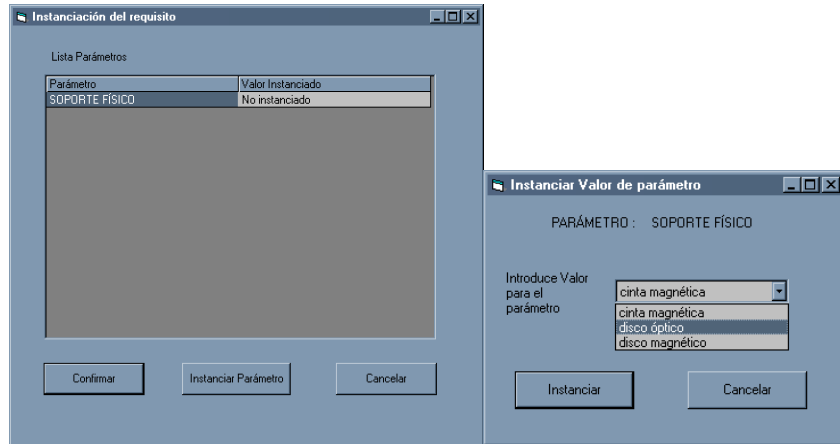


Fig. 5. Interfaz para la instanciación de los requisitos.

5. Conclusiones y Vías Futuras

En este trabajo hemos presentado una herramienta, *SirenTool*, para dar soporte automatizado a SIREN, un método para la reutilización de requisitos que estamos definiendo en nuestro grupo de investigación. Con esta nueva herramienta, SIREN puede ser adoptado en entornos académicos o industriales, permitiéndonos valorar y mejorar la definición del método, y (esperamos que) produciendo beneficios al permitir la reutilización de requisitos procedentes de proyectos anteriores.

La comparativa de algunas de las herramientas CARE comerciales ha revelado la limitación de éstas en cuanto al soporte de la reutilización y a cuestiones específicas del proceso SIREN. Ante esta situación se presentó la disyuntiva de realizar una herramienta *ad-hoc* o extender una de las existentes. Se decidió extender la funcionalidad de RequisitePro, lo que ha supuesto un gran ahorro de esfuerzo. Cabe decir que el trabajo realizado para RequisitePro se podría haber realizado para cualquiera de las otras herramientas que también disponen de una interfaz de extensibilidad (Caliber-RM, DOORS).

Nuestra herramienta aporta a la reutilización de requisitos la gestión de requisitos parametrizados, de las relaciones de traza inclusiva y exclusiva, y añade facilidades de búsqueda más avanzadas a las ya soportadas por las herramientas comerciales. Estas funciones se podrían conseguir mediante cualquiera de las tres herramientas comerciales evaluadas, pero con *SirenTool* se automatizan y de esta manera se consigue una ganancia de productividad.

En cuanto a las vías futuras queremos decir que el prototipo de la herramienta se encuentra en continua evolución. Actualmente se trabaja en la mejora de sus

funcionalidades, ampliando los criterios de búsqueda para la reutilización de requisitos, definiendo nuevos catálogos de requisitos, y estudiando la integración con otras herramientas, como Rational Rose, para la definición de los requisitos funcionales.

6. Referencias

- 1- Alexander, Ian. Requirements Engineering Tool Vendors and Freeware Suppliers. <http://easyweb.easynet.co.uk/~iany/other/vendors.htm>
- 2- CaliberRM, Borland. <http://www.borland.com/caliber/index.html>.
- 3- DOORS, Quality Systems & Software. <http://www.telelogic.com/doors>.
- 4- Glass, R.L. Software Runaways: Monumental Disasters. Prentice Hall, 1998.
- 5- Glass, R.L. Software Engineering: Facts and Fallacies. Addison-Wesley, 2002.
- 6- INCOSE (the International Council On Systems Engineering). <http://www.incose.org>.
- 7- IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications, en Volumen 4: Resource and Technique Standards, The Institute of Electrical and Electronics Engineers, Inc. IEEE Software Engineering Standards Collection.
- 8- IEEE Std 1233-1998. Guide for Developing System Requirements Specifications, en Volume 1: Customer and Terminology Standards. 1999, The Institute of Electrical and Electronics Engineers, Inc. IEEE Software Engineering Standards Collection.
- 9- Kotonya, G., Sommerville, I., Requirements Engineering. Processes and Techniques, John Wiley & Sons, 1998.
- 10- REM, Amador Durán. <http://klendathu.lsi.us.es/REM>.
- 11- Requisite Pro, Rational Software. <http://www.rational.com>.
- 12- Robertson, S., Robertson, J., Mastering the Requirement Process. Addison-Wesley, 1999.
- 13- Smith, J.M. Troubled IT Projects Prevention and Turnaround. IEEE, 2001.
- 14- Sommerville I. Ingeniería del Software, 6ª edición, Addison Wesley. Año 2002.
- 15- The Atlantic Systems Guild inc. Requirements Engineering Tools. <http://www.systemsguild.com/GuildSite/Robs/retools.html>
- 16- Toval, A., García Báidez, F., Technical Report TR LSI 2-01. Departamento de Informática y Sistemas. Universidad de Murcia. Reutilización de Requisitos de Seguridad compatibles con MAGERIT. 2000.
- 17- Toval, A., Nicolás, J., Moros, B., Baidez, F. Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach. Requirements Engineering Journal (2002) 6:205-219.
- 18- Toval, A., Nicolás, J., Moros, B., Lasheras, J. Análisis y Prototipado de una Herramienta CARE de Soporte al Método SIREN. Proyecto Informático. Facultad de Informática. Departamento de Informática y Sistemas. Universidad de Murcia. Marzo 2003.
- 19- Toval, A., Nicolás, J., Moros, B. Requisitos Reutilizables de Seguridad en Sistemas de Información y Bases de Datos, Seguridad en Bases de Datos, pp:269-300, Edit. Dintel 2001.
- 20- Toval, A., Nicolás, J., Moros, B. SIRENrm, Un Proceso de Ingeniería de Requisitos Basado en Reutilización. En "Applying Requirements Engineering". Amador Durán y Miguel Toro (eds). Catedral Publicaciones. 2002.
- 21- Toval, A., Olmos, A., Piattini, M.. Legal Requirements Reuse: A Critical Success Factor for Requirements Quality and Personal Data Protection. Proceedings of the IEEE Joint International Conference on Requirements Engineering (ICRE'02 and RE'02), pp: 9-13. Septiembre 2002.
- 22- Toval, A., Olmos, A., Technical Report TR LSI 3-01. Departamento de Informática y Sistemas. Universidad de Murcia. Reutilización de Requisitos de Protección de Datos con SIRENrm. 2000.