

Reutilización de Requisitos Organizados como una Familia de Diagramas

Oscar López¹, Miguel A. Laguna², and Francisco J. García³

¹ Technological Institute of Costa Rica

olopez@infor.uva.es

² University of Valladolid, Spain

mlaguna@infor.uva.es

³ University of Salamanca, Spain

fgarcia@usal.es

Resumen La determinación de requisitos es un factor crítico en el proceso de desarrollo del software porque es una actividad consumidora de recursos y propensa a errores. El disponer de elementos reutilizables de requisitos, calificados y organizados en un repositorio, contribuye al mejor aprovechamiento de los recursos del desarrollo y en la disminución de errores. No obstante, la diversidad de representaciones y diferentes niveles de descripción de requisitos dificulta su reutilización. En este artículo se propone una aproximación para reutilizar diagramas semiformales dentro de un modelo de familia de requisitos. El modelo de familia consiste en un conjunto de diagramas de requisitos y un modelo de objetivos del dominio. El artículo sugiere cómo obtener una nueva especificación de requisitos con el apoyo de un prototipo de herramienta de reutilización de requisitos.

Palabras clave: Ingeniería de requisitos, reutilización de requisitos.

1. Introducción

La reutilización de requisitos es una área prometedora para la ingeniería de requisitos [22]. Diferentes autores [2,3,4,9,17] recomiendan el abordar la reutilización lo más temprano posible en el ciclo de vida del software. La reutilización de productos de la fase de los requisitos plantea una serie de ventajas, entre ellas el estimular la reutilización a lo largo del resto del ciclo de vida y permitir un mejor aprovechamiento del esfuerzo de desarrollo [5,10,21]. En el contexto de la producción empresarial del software, la reutilización de requisitos permite completar la especificación de nuevas aplicaciones para así mejorar la calidad y productividad del proceso de ingeniería de requisitos, véase [6] y [12].

La reutilización responde al principio de aprovechar esfuerzos previos y exitosos para completar un nuevo desarrollo. Las investigaciones en reutilización han mostrado que se requiere un enfoque particular, propio y adaptado a la naturaleza del software, para disponer de la capacidad de reutilizar artefactos de desarrollos previos. Este enfoque se basa en la *selección, especialización e integración* de elementos del software que hayan sido intencionalmente *diseñados, desarrollados y documentados* para servir

como materia prima para nuevos desarrollos [7,11]. El espectro de elementos reutilizables del software abarca diferentes productos del ciclo de vida dentro de los diferentes niveles de abstracción, incluyendo productos obtenidos en la fase de requisitos [9].

Incluir la fase de requisitos contribuye en la eficacia del proceso general de reutilización. La eficacia en la reutilización pasa por un elevado nivel de abstracción, lo que da lugar a la reutilización orientada al dominio [16]. Elevar el nivel de abstracción no consiste en reutilizar elementos informales y conceptuales sino en reutilizar elementos que implementan los conceptos del dominio. Según Leite [13], para implementar los conceptos del dominio se debe organizar taxonómicamente la información o conocimiento del dominio, lo cual implica un costo elevado y años de experiencia, además del soporte de un proceso bien definido. La construcción de taxonomías de dominio, en el contexto de la reutilización de requisitos, se ha abordado en diferentes trabajos, por ejemplo en [6,10,12,21].

En este artículo se propone una aproximación orientada al dominio para reutilización de diagramas de requisitos del software. El ámbito de estudio se restringe a los requisitos que se representan mediante seis técnicas semiformales de modelado ampliamente conocidas (escenarios, casos de uso, diagramas de actividades, flujos de datos, diagramas documentos-tareas y *workflows*). La propuesta se basa en (a) la obtención de una estructura de *organización de una familia de diagramas* de requisitos en un dominio, y (b) la generación de una *nueva especificación de requisitos a partir de la familia* de diagramas. Para abordar el problema hemos propuesto un metamodelo de requisitos y un modelo de proceso para organizar y aprovechar los diagramas de requisitos existentes en un dominio. El proceso es soportado por R^2 , un prototipo de herramienta de reutilización de requisitos. El resto del artículo se distribuye en 4 secciones adicionales. La Sección 2 presenta la forma de organizar la información de los requisitos de un dominio como una familia de diagramas. La Sección 3 propone la forma de obtener una nueva especificación de requisitos a partir de los diagramas almacenados en el repositorio. La Sección 4 presenta experiencias iniciales en reutilización de requisitos. Las conclusiones y trabajo futuro, en la Sección 5, cierran el artículo.

2. La Familia (de Diagramas) de Requisitos

Las líneas de productos permiten incrementar la productividad, reducir costos de desarrollo de cada producto, obtener mejores estimaciones referentes al proceso de desarrollo y mejorar la calidad de los productos [2,3]. El desarrollo de líneas de productos involucra el establecimiento de una familia de requisitos que refleja los resultados del análisis del dominio para identificar *comunalidades*, *variabilidades* y dependencias entre los requisitos. Diferentes aproximaciones de líneas de productos, como FODA [8], PuLSE [1], y ODM [20], se fundamentan en un análisis previo del dominio. Sin embargo, en muchos dominios se ha recopilado información de requisitos mediante métodos de ingeniería de requisitos, que se enfocan generalmente hacia un producto en particular y no desde una perspectiva institucional de línea de productos. Para abordar la reutilización sistemática de esa información generada en el proceso de ingeniería de requisitos se requiere el soporte para organizar familias (de diagramas) de requisitos.

2.1. El Metamodelo de Requisitos

La base teórica para la organización de requisitos como una familia, y para la generación de nuevas especificaciones a partir de esa familia, es un metamodelo que permite describir cada elemento de la información de requisitos como una instancia de alguna clase de requisitos. En la Figura 1 se muestra parte de nuestro metamodelo de requisitos cuyas entidades centrales son *Objetivo del Dominio*, *Modelo de Representación* (de requisitos) y *Unidad de Modelado*, más detalles pueden consultarse en [14].

Un Objetivo del Dominio constituye un requisito fundamental que debe satisfacerse a través del desarrollo de una clase de aplicaciones (actuales o futuras). No se restringe el tipo de información -funcional o no funcional, por ejemplo- susceptible de ser presentada como Objetivo del Dominio, la única restricción es que sea una intención, finalidad o meta, suficientemente significativa en el dominio, que debe satisfacerse mediante el desarrollo de aplicaciones software. Cada Objetivo del Dominio tiene una serie de atributos: identificador, complejidad, costo, tecnología requerida, prioridad, cualificación, e información adicional. Los Objetivos del Dominio están enlazados mediante la Relación Modelo cuyos roles permiten representar asociaciones padre-hijo. Esta relación permite establecer una estructura jerárquica en forma de retículo en la cual cada Objetivo del Dominio puede tener cero o muchos hijos y cero o muchos padres.

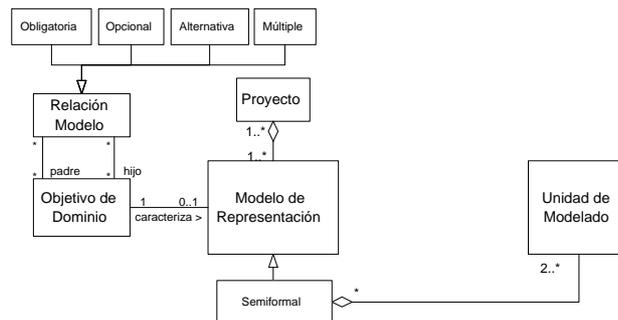


Figura1. Parte del Metamodelo de Requisitos.

Las relaciones padre-hijo entre Objetivos del Dominio pueden ser de cuatro tipos:

- **Obligatoria:** Establece que las instancias de objetivo hijo están unidas fuertemente a las instancias padre de tal forma que la separación de ellos conlleva a la pérdida del sentido de cada uno en el dominio. Por ejemplo,
 - Objetivo Padre: Proporcionar servicios de internet
 - Objetivo Hijo: Soporte de conexión física a red pública de datos
- **Opcional:** Esta relación enlaza débilmente a las instancias de objetivo hijo con el padre, de modo que esa unión se puede separar sin que ello produzca la pérdida del sentido de cada elemento participante. Por ejemplo,
 - Objetivo Padre: Proporcionar servicios de internet
 - Objetivo Hijo: Proporcionar protección mediante cortafuegos

- Alternativa: Relación que indica que el enlace cobra sentido al escoger uno y sólo uno de los elementos hijos. Por ejemplo,
 - Objetivo Padre: Soporte de interfaz a grupos de discapacidad
 - Objetivos Hijos: Interfaz para sordos - Interfaz para ciegos - Interfaz para problemas de movilidad de extremidades
- Múltiple: Relación que indica que se debe escoger al menos una, pero puede escogerse varias, de las instancias de objetivos hijos. Por ejemplo,
 - Objetivo Padre: Proporcionar servicios de internet
 - Objetivos Hijos: Soportar correo electrónico - Soportar acceso a web - Soportar servicio chat

En general, los requisitos actúan como un medio de comunicación, negociación y especificación de las necesidades de los *stakeholders*. Por esta razón se utilizan diferentes Modelos de Representación (de requisitos), que pueden ser de tipo Formal, Semiformal e Informal, para documentar los requisitos. El metamodelo en su estado actual permite describir el tipo Semiformal, y queda abierta la posibilidad de incluir la descripción de los otros tipos de modelos en el trabajo futuro. El tipo Semiformal es una agregación de Unidades de Modelado y se asocian a los Objetivos del Dominio mediante la relación *caracteriza a*, ver Figura 1. Un Objetivo del Dominio puede ser caracterizado, a lo sumo, por un Modelo de Representación. Existen diferentes niveles de descripción en Objetivos de dominio, desde muy abstractos a menos abstractos. Un Objetivo de Dominio que no se asocia a un Modelo de Representación constituye un objetivo muy abstracto que debe ser desglosado mediante objetivos hijos. Esto concuerda con el trabajo de Kuusela [12] que organiza los requisitos de líneas de productos en una jerarquía de definición formada por objetivos de diseño y decisiones de diseño.

Cada Proyecto de requisitos, dentro de la familia, está formado por la agregación de Modelos de Representación. De este modo, cuando se construye un proyecto específico de requisitos que pertenece a la familia, cada diagrama de requisitos se modela como parte de un Proyecto, y asociado al Objetivo de Dominio correspondiente. La multiplicidad de la agregación de los Modelos de Representación en la entidad Proyecto refleja la dimensión de reutilización de diagramas de requisitos. Un Modelo de Representación forma parte de al menos un Proyecto, y puede ser reutilizado en varios proyectos.

2.2. Diagramas que se pueden instanciar del Metamodelo

Los trabajos de reutilización de requisitos que abordan aspectos de taxonomías de dominio [6,10,12,21] afrontan la especificación de requisitos reutilizables mediante texto natural y lenguajes formales. A diferencia de estos, nosotros utilizamos diagramas semiformales existentes en un dominio. Hemos restringido el ámbito de estudio a los requisitos que se representan mediante las técnicas casos de uso, escenarios, diagramas de actividades, flujos de datos, diagramas documentos-tareas y *workflows*. Estos diagramas permiten el modelado de requisitos que se especifican desde la perspectiva de los clientes, los llamados Requisitos-C [19]. Esta restricción del ámbito de estudio obedece a la necesidad de elevar el nivel de abstracción [16] para iniciar la reutilización de los requisitos. Otras técnicas y niveles de descripción de requisitos serán incluidos dentro de una aproximación de reutilización de requisitos como parte de nuestro trabajo futuro.

3. Proceso de Reutilización de Requisitos

Para alcanzar los fines de la reutilización sistemática de los requisitos se requiere el análisis de los documentos de requisitos y su organización en un repositorio de artefactos reutilizables [5]. Nosotros afrontamos estas acciones a través de un proceso, basado en la Familia de Diagramas, ver Figura 2. Este proceso consta de tres subprocesos: (1) Organización de la Familia de Diagramas, (2) Especificación de un Nuevo Producto, y (3) Ingeniería de Requisitos. La salida del primer subproceso es la familia de diagramas, que es utilizada en el segundo subproceso para obtener la especificación de un nuevo proyecto de requisitos. Si existe la necesidad de expandir la familia de diagramas entonces se envían los nuevos objetivos al tercer subproceso. La Figura muestra que el subproceso 2 retroalimenta al Modelo de Familia, es decir que los nuevos productos se incorporan a la familia de diagramas para garantizar que esta familia mantiene todos los requisitos de todos los productos miembros.

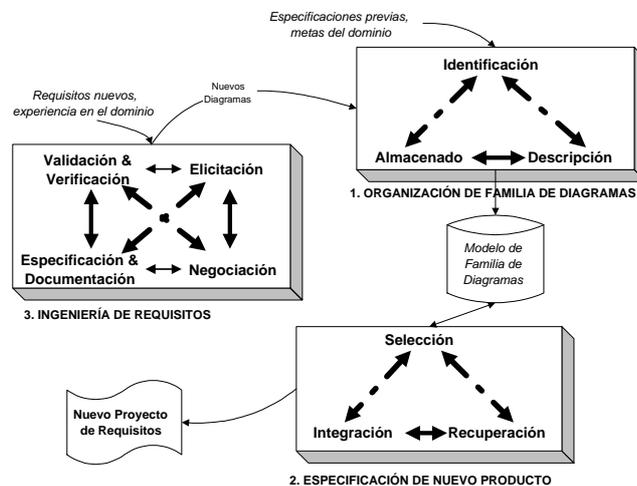


Figura2. Proceso de Reutilización de Diagramas de Requisitos.

3.1. Organización de la Familia de Diagramas

La actividad central del proceso de reutilizar requisitos es la organización de la familia de diagramas. Las entradas para esta actividad son los documentos de especificaciones previas de requisitos, además de conocimiento del dominio. El propósito de esta actividad es obtener una partición de los requisitos mediante las acciones de *identificación*, *descripción*, y *almacenamiento* de la familia de diagramas. Esta organización de requisitos se basa en los puntos de vista de los *stakeholders*, como guía para obtener la familia, lo cual también ha sido utilizado en el trabajo de Mannion [10]. Para la obtención de la familia se procede de la siguiente manera:

- *Identificación* de los *stakeholders* del dominio y establecimiento de un diccionario de dominio. A partir de esta identificación se obtienen y confrontan las metas de cada uno de los *stakeholders* y se establece un retículo de Objetivos del Dominio, donde estos objetivos están enlazados mediante la Relación Modelo. Esta acción requiere de experiencia y conocimiento del dominio, para guiar la toma de decisiones en el establecimiento del retículo. Estas decisiones incluyen el tratamiento de metas traslapadas, metas en conflicto y dependencias entre las metas de los *stakeholders* a través de la Relación Modelo.
- *Descripción* de los Objetivos de Dominio mediante diagramas de requisitos existentes en el dominio. Esta etapa pretende aprovechar los diagramas de requisitos existentes en las aplicaciones del dominio, sin embargo se puede requerir la reescritura de algunos diagramas mediante un proceso iterativo y que se realiza con apoyo del entorno que se presenta en la Sección 4.1.
- *Almacenamiento* de la familia de diagramas en un repositorio que brinda el soporte operativo para *crear, gestionar y utilizar* los diagramas reutilizables. Esta etapa se fundamenta en la propuesta de McClure [15], que establece que el repositorio es *una herramienta para la definición, almacenamiento, acceso y gestión de la información que describe a una empresa y a sus sistemas software, durante cada una de las diferentes fases del ciclo de vida del software*. El Modelo de la Familia de Diagramas contiene información de las fases iniciales del proceso del software.

El Modelo de la Familia de Diagramas incorpora tanto puntos comunes como puntos de variación mediante las relaciones entre los Objetivos del Dominio. Las relaciones de tipo *Opcional, Alternativa y Múltiple* modelan los puntos de variabilidad. La relación de tipo *Obligatoria* modela los puntos comunes.

3.2. Especificación de un Nuevo Producto

La Especificación de un Nuevo Producto a partir de la familia consiste en tomar decisiones acerca de las *comunalidades y variabilidades* de requisitos. Las *variabilidades y comunalidades* se establecen en el Modelo de la Familia de Diagramas. Cada Proyecto que pertenece a la familia de productos posee un conjunto de requisitos definido. Por tanto, al especificar un nuevo producto se debe escoger entre las posibilidades que ofrece el Modelo de Familia de Diagramas. Para especificar un nuevo producto se procede a recorrer el retículo de objetivos para obtener los elementos reutilizables que se ajusten a las nuevas necesidades. Esta fase consiste en las siguientes acciones:

- *Selección* de los Objetivos del Dominio aplicables al nuevo producto. Se recorre el retículo preguntando al usuario por la necesidad de incluir el objetivo actual en el nuevo producto. Los objetivos seleccionados se agregan al nuevo Proyecto y se incluyen en el documento de especificación del Nuevo Proyecto de Requisitos.
- *Recuperación* de los diagramas que correspondan a los Objetivos del Dominio seleccionados. Este paso involucra la interacción con el motor gestor de la base de datos donde se implementa el repositorio.
- *Integración* del nuevo Proyecto de Requisitos en el Modelo de Familia de Diagramas. El repositorio debe ser retroalimentado mediante la integración del nuevo

Proyecto. En este punto es importante contar con medios para garantizar la consistencia lógica de la especificación del nuevo Proyecto, como puede ser el análisis apoyado por aplicaciones de redes de Petri.

3.3. Ingeniería de Requisitos

La especificación de nuevos productos puede implicar la incorporación de nuevos requisitos en la familia de diagramas. En este caso, los nuevos requisitos son especificados mediante el proceso de ingeniería de requisitos que incluye las actividades de *elicitación, negociación, especificación & documentación, y verificación & validación* [18]. La salida de este proceso es un conjunto de *Nuevos Diagramas* que pasan a formar parte de la familia. Como estos diagramas se especifican una vez que se ha construido el Modelo de Familia de Diagramas en el dominio, su especificación se realiza con un compromiso de reutilización y con el apoyo del entorno R^2 .

4. Experiencias de Reutilización de Requisitos

El proceso propuesto de reutilización de requisitos es el eje de dos acciones conjuntas: el desarrollo de un entorno de reutilización de requisitos y la conducción de un caso de estudio.

4.1. Un Entorno para Reutilización de Requisitos (R^2)

R^2 es una herramienta de soporte para las actividades del proceso de reutilización de requisitos. Este entorno se desarrolla en el lenguaje JAVA y proporciona una interfaz gráfica para acceder a la funcionalidad para editar diagramas de requisitos, gestionar el repositorio, e intercambiar datos vía XML con una aplicación de redes de Petri coloreadas. El metamodelo, presentado en [14], se ha implementado en una base de datos relacional soportada por ORACLE.

Para soportar la edición de diagramas de requisitos, los diferentes Proyectos dentro del dominio pueden ser inspeccionados mediante una vista de proyecto, según se muestra en la Figura 3. Los identificadores de los proyectos aparecen en el lado izquierdo de la pantalla mientras que la notación gráfica del diagrama actual aparece en el lado derecho. Los proyectos y los diagramas pueden expandirse y contraerse utilizando los signos de más y menos en el lado izquierdo de su identificador.

El editor de requisitos cumple dos funciones muy importantes dentro del entorno de reutilización de requisitos. En primer lugar, el editor permite re-estructurar los diagramas según sea necesario al momento de crear el Modelo de Familia de Diagramas del dominio. En segundo lugar, el editor permite dar mantenimiento a los diagramas de requisitos dentro de la familia. Al soportar diversas técnicas de modelado, el editor ofrece facilidades para la comunicación con un mayor número de *stakeholders* en el dominio.

La especificación de un nuevo producto es soportado en R^2 a través de una vista de Objetivos del Dominio que se muestra en la Figura 4. La inspección de la vista de objetivos permite seleccionar el diagrama que describe al objetivo actual para añadirlo

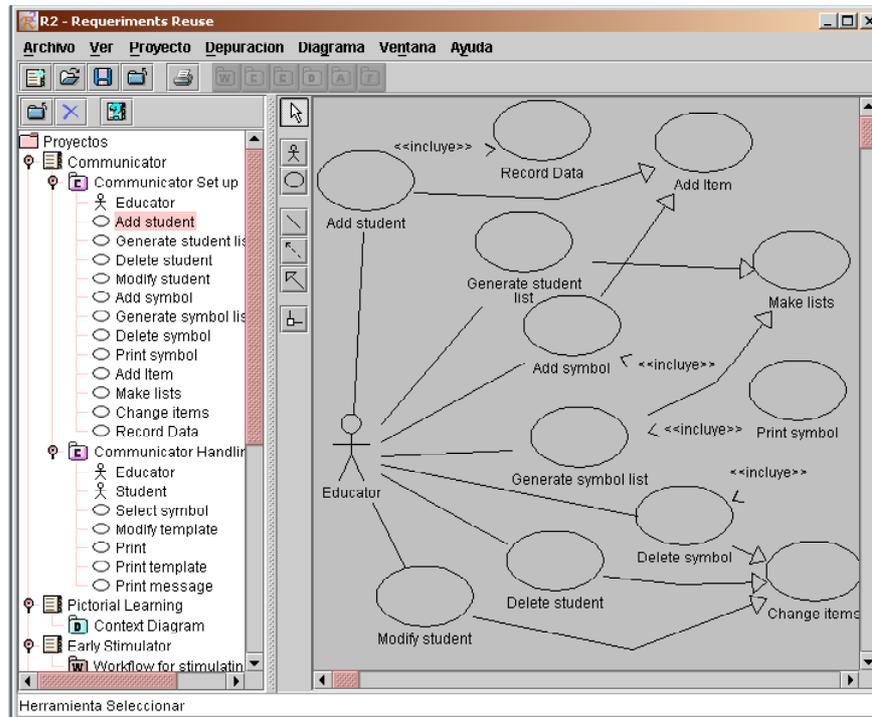


Figura3. Vista de proyecto en el Entorno R^2 .

al proyecto en construcción. Los identificadores de los objetivos aparecen en el lado izquierdo de la pantalla mientras que la información del objetivo actual se despliega en el lado derecho. Los objetivos pueden expandirse y contraerse utilizando los signos de más y menos en el lado izquierdo.

La exportación de datos permite que los proyectos de requisitos sean llevados a una aplicación de redes de Petri coloreadas, esto con el propósito de garantizar la consistencia lógica de la especificación mediante la técnica de simulación. Esta funcionalidad es asistida por el entorno R^2 .

4.2. Caso de Estudio: Aspectos preliminares

En la actualidad se está conduciendo un estudio en el dominio de software para discapacitados. En este dominio se han desarrollado 10 diferentes aplicaciones sin una perspectiva de línea de productos, razón por la cual se cuenta con diferentes diagramas de requisitos que comparten similitudes y *variabilidades*. Los diagramas semiformales se han organizado en una familia de requisitos con apoyo de un experto del dominio mediante múltiples reuniones de trabajo. Ha sido necesario re-escribir algunos de los diagramas, así como crear nuevos diagramas a sugerencia del experto del dominio.

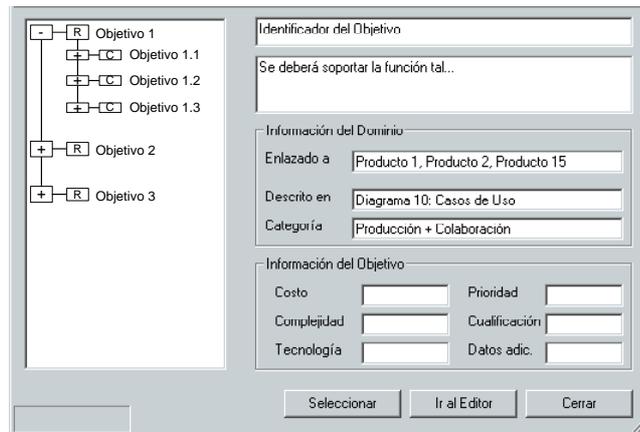


Figura4. La vista de Objetivos de Dominio en el Entorno R^2 .

La construcción del Modelo de Familia de Diagramas se ha abordado mediante un proceso subjetivo que demanda aspectos de conocimiento y experiencia en el dominio. En muchos casos ha sido necesario alcanzar soluciones de compromiso respecto a la forma de relacionar los requisitos en la familia. Aunque se han publicado trabajos similares, en cuanto a la organización de requisitos en familias [6,10,12,21], estos no brindan detalles de su proceso de construcción. Todo esto pone en evidencia que se requiere un período de estabilización de la familia de requisitos. Además, se corrobora lo afirmado por Leite [13] respecto a la necesidad de realizar experimentos en el proceso de construcción de dominios.

El contar con un entorno prototipo ha sido un factor relevante para facilitar la creación de la familia de requisitos. Para gestionar la cantidad de información de requisitos que existe en dominios es fundamental la tecnología de repositorios. Para integrar las diversas técnicas de modelado de requisitos ha sido esencial el aporte de la funcionalidad de edición e inspección de los diagramas de requisitos.

5. Conclusiones

En este artículo se ha presentado un proceso para la reutilización de productos iniciales del ciclo de vida del software. La contribución del artículo es una propuesta para la organización de diagramas semiformales de requisitos dentro de un retículo construido con los objetivos del dominio. A partir de los diagramas organizados se pueden obtener especificaciones de nuevos productos dentro del dominio. Un aspecto positivo de la propuesta es el soporte dado por un prototipo de entorno de reutilización de requisitos (R^2). Tanto el proceso como el entorno se hallan en fase validación en el dominio de aplicaciones de software para discapacitados, en la Universidad de Valladolid.

El elemento crítico de la propuesta es la etapa de construcción de la familia de diagramas semiformales de requisitos. Esta etapa es guiada por valoraciones subjetivas en la organización de los requisitos, consume gran parte del tiempo y requiere gran

experiencia en el dominio, lo cual coincide con lo apuntado por Leite [13], Sutcliffe y Maiden [21], y Mannion [10]. Sin embargo, la necesidad de alternativas para agilizar el proceso de los requisitos nos estimula para seguir en la investigación con familias de requisitos en dominios.

A partir del trabajo desarrollado, se abren dos interesantes líneas de acción. Primero, añadir detalles de funcionalidad al entorno R^2 para un mejor soporte a las labores de reutilización de requisitos. R^2 será enriquecido mediante la opción de «arrastrar y soltar» de modo que facilite la edición de la vista de proyecto y de objetivos de dominio. Además se desarrollará un módulo que permita a R^2 operar como asistente de reutilización recorriendo automáticamente el retículo de diagramas y así permitir al usuario realizar la selección de elementos reutilizables según sus necesidades. Segundo, completar la experimentación con la familia de requisitos en el dominio de software para discapacitados. Además, se planea incluir otras representaciones de requisitos como los diagramas de clases, diagramas de secuencias y diagramas de colaboración.

Agradecimientos

Este trabajo es en parte financiado por el proyecto DOLMEN (CICYT-TIC2000-1673-C06-05). Oscar López agradece a la Agencia Española de Cooperación Internacional (AECI) y al Ministerio de Ciencia y Tecnología de Costa Rica.

Referencias

1. J. Bayer, O. Flege, P. Knauber, R. Laqua, D. Muthig, K. Schmid, T. Widen, and J.-M. De-Baud. PuLSE: A methodology to develop software product lines. In *Proceedings of the Fifth ACM SIGSOFT Symposium on Software Reusability (SSR'99)*, pages 122–131, Los Angeles, CA, USA, May 1999. ACM.
2. Jan Bosch. *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*. ACM Press. Addison-Wesley, May 2002.
3. Paul C. Clements and LindaÑ. Nothrop. *Software Product Lines: Practices and Patterns*. The SEI series in software engineering. Addison-Wesley, 2002.
4. Richard E. Creps, Mark A. Simos, and Rubén Prieto-Díaz. The STARS conceptual framework for reuse processes. In *Proceedings of STARS'92*, November 1992.
5. Jacob L. Cybulski. Patterns in software requirements reuse. Technical report, Department of Information Systems. University of Melbourne, July 1998.
6. Stuart R. Faulk. Product-line requirements specifications (PRS): an approach and case study. In *Proceedings of 5th IEEE International Symposium on Requirements Engineering*, pages 48–55, Toronto, Canada, August 2001. IEEE Computer Society.
7. Francisco José García. *Modelo de Reutilización Soportado por Estructuras Complejas de Reutilización Denominadas Mecanos*. PhD thesis, Universidad de Salamanca, Spain, 2000.
8. K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. Feature-Oriented Domain Analysis (FODA). Feasibility study. Technical Report CMU/SEI-90-TR21 (ESD-90-TR-222), Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213, November 1990.
9. E. Karlsson, editor. *Software Reuse. A Holistic Approach*. Wiley Series in Software Based Systems. John Wiley and Sons Ltd, 1995.
10. Barry Keepence, Mike Mannion, Hermann Kaindl, and Joe Wheadon. Reusing single system requirements from application family requirements. In *Proceedings of the 21st International Conference on Software Engineering*, pages 453–462. ACM Press, May 1999.

11. Charles W. Krueger. Software reuse. *ACM Computing Surveys*, 24(2):131–183, June 1992.
12. J. Kuusela and J. Savolainen. Requirements engineering for product lines. In *Proceedings of ICSE 2000, IEEE*. IEEE Computer Society, 2000.
13. J. Leite. Are domains really cost effective? In *Proceedings of the Workshop on Institutionalizing Software Reuse (WISR'99)*, January 1999.
14. O. López, M.A. Laguna, and F.J. García. Metamodeling for requirements reuse. In *Proceedings of the V Workshop em Engenharia de Requisitos WER 2002*, To appear. Valencia, España, November 2002.
15. Carma McClure. *Software Reuse Techniques: Adding Reuse to the System Development Process*. Prentice-Hall, 1997.
16. James M. Neighbors. The Draco approach to constructing software from reusable components. *IEEE Transactions on Software Engineering*, SE-10(5):564–574, September 1984.
17. James M. Neighbors. The evolution from software components to domain analysis. *International Journal of Software Engineering and Knowledge Engineering*, 2(3):325–354, October 1992.
18. K. Pohl. Requirements engineering, an overview. *Encyclopedia of Computer Science and Technology*, Vol. 36, Marcel Decker Inc., 1996.
19. H. Dieter Rombach. Software specifications: A framework. SEI Curriculum Module. Technical Report SEI-CM-11-2.1, Software Engineering Institute, Carnegie Mellon University, January 1990.
20. Mark Simos, Dick Creps, Carol Klingler, Larry Levine, and Dean Allemang. Organization domain modeling (ODM) guidebook - version 2.0. Technical Report STARS-VC-A025/001/00, Lockheed Martin Tactical Defense Systems, 9255 Wellington Road Manassas, VA 22110-4121, June 1996.
21. A. Sutcliffe and N. Maiden. The domain theory for requirements engineering. *IEEE Transactions on Software Engineering*, 24(3):174–196, March 1998.
22. Axel van Lamsweerde. Requirements engineering in the year 00: A research perspective. In *Proceedings of the 22nd. International Conference on Software Engineering*, Limerich, June 2000. ACM Press.