

HACIA UN MODELO DE TRAZAS DESACOPLADO DE LOS MODELOS DE REQUISITOS

Graciela Hadad¹, Andrea Vera¹, Jorge Doorn^{1,2}, Gladys Kaplan¹

¹Departamento de Ingeniería e Investigaciones Tecnológicas, Universidad Nacional de La Matanza

²Depto. de Computación y Sistemas, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Pcia. de Buenos Aires
gracielahadad@gmail.com, andreafabianavera@gmail.com, jdoorn@exa.unicen.edu.ar, gladyskaplan@gmail.com

CONTEXTO

La presente propuesta se corresponde con la línea de investigación “Gestión de Requisitos” en el proyecto de investigación “Consolidación de Requisitos de Software” del grupo de investigación en “Ingeniería de Requisitos” de la UNLaM.

RESUMEN

Como menciona Leite [Leite 97]: “el cambio es una propiedad intrínseca al software”, y por lo tanto merece una consideración especial en el proceso de desarrollo: la administración de dichos cambios. Desde la perspectiva de la ingeniería de requisitos, esta actividad se denomina: la gestión de requisitos. Ella involucra dos sub-actividades centrales: el control del versionado y la trazabilidad de los mismos. En este proyecto se aspira a crear un modelo de trazas que dé apoyo y contexto a los requisitos pero sin que perturbe a los modelos de requisitos, y especificar un proceso de administración de trazas que involucre facilidad de creación y navegación.

Palabras clave: *trazabilidad, requisitos, gestión de requisitos*

1. INTRODUCCIÓN

Los requisitos evolucionan como consecuencia de cambios que el Universo de Discurso (UdeD) impone al software. Numerosos estudios realizados en las décadas de los 80 y los 90 del siglo pasado han mostrado que el 50% o más de los requisitos van a cambiar antes que el sistema de software se ponga en operación [Kotonya 98]. Si bien estas estadísticas reflejan la situación de la industria del software en una época en la que poco se había avanzado en la Ingeniería de Requisitos, la casi totalidad de los investigadores en el área siguen atestiguando que los cambios en los requisitos de un sistema de software es una realidad vigente en la actualidad. Es por esta razón que se debe poner énfasis en especificaciones de requisitos que sean capaces de evolucionar a lo largo del proyecto de desarrollo y que permitan el seguimiento de los mismos desde cualquier punto del proceso de producción del software hasta sus orígenes.

La gestión de requisitos es la actividad que administra los cambios en los requisitos. Estos cambios son de dos tipos: cambios en requisitos existentes o aparición de nuevos requisitos. Los factores que provocan estos cambios son básicamente la evolución del UdeD, incluyendo cambios en las expectativas de los clientes y usuarios y el descubrimiento de defectos en la comprensión o definición de los requisitos en

subsiguientes etapas del proyecto.

Para que la gestión sea una actividad viable y exitosa, es indispensable la administración de las dependencias entre los requisitos y la administración de las vinculaciones entre el documento de requisitos y otros documentos, modelos y componentes del software [Kotonya 98] [Davis 99]. Pues un cambio o incorporación de un único requisito puede alterar un conjunto de requisitos en cualquiera de los Modelos de Procesos de Software, y dependiendo de la etapa en que se encuentre el proyecto es necesario la actualización de la documentación y otros artefactos del software, lo cual debe identificarse y evaluarse previo a su efectiva realización. Esto está intrínsecamente relacionado con un concepto clave: la trazabilidad de los requisitos. La trazabilidad de los requisitos se refiere a la habilidad de definir, capturar y seguir las pistas dejadas por los requisitos sobre otros elementos del ambiente de desarrollo del software y las pistas dejadas por dichos elementos sobre los requisitos [Pinheiro 96]. Por otro lado, esta actividad puede ser considerada desde la visión del ciclo de vida del software como parte de una actividad de mayor alcance: la Gestión de Configuración [Crnkovic 99].

Para identificar los requisitos afectados por los cambios, así como los documentos, modelos y componentes involucrados, se debe utilizar la información de trazabilidad. Es fundamental contar con alguna herramienta de soporte a la gestión pues generalmente se manejan grandes cantidades de información en distintos formatos, y es necesario conservar el rastro de cada cambio solicitado, en evaluación e implementado.

Otro problema que se presenta en la gestión de requisitos es el control del versionado [Sawyer 01] [CMMI 06], es decir, el mantenimiento de la historia de los cambios en los requisitos. El sistema de versionado [Conradi 98] es un soporte de servicios indispensable para garantizar una adecuada trazabilidad de los mismos, da una idea de la evolución de los requisitos, registrando todos los cambios realizados con su justificación.

La trazabilidad de los requisitos es una función de la gestión de requisitos, que se encarga de mantener vínculos entre requisitos dependientes, entre requisitos, diseño y código, y entre requisitos y fuentes de información que los originaron. Según [IEEE Std 830-1998], un requisito es rastreable si su origen es claro y si

facilita su referencia en la documentación de futuros desarrollos o mejoras. Kotonya & Sommerville ofrecen una definición más precisa [Kotonya 98]: un requisito es rastreable si se puede determinar quién lo sugirió, qué requisitos están relacionados con él y cómo se relaciona con otra información tal como: diseño del sistema, implementaciones y documentación del usuario. Muchos investigadores han estudiado el tema y realizado propuestas sobre la trazabilidad de los requisitos [Gotel 94] [Wieringa 95] [Palmer 96] [Jarke 98] [Pinheiro 04].

Se han propuesto distintas formas de trazabilidad de requisitos. Tanto Wieringa [Wieringa 95] como [IEEE Std 830-1998], mencionan dos tipos: Backward traceability (habilidad de rastreo de un requisito a sus fuentes) y Forward traceability (habilidad de rastreo de un requisito hacia los componentes de diseño y código). Mientras que Davis en [Davis 93] menciona cuatro tipos pues agrega el rastreo inverso en cada caso, es decir de una fuente a un requisito y de un componente a un requisito; CMMI [CMMI 06] la denomina rastreabilidad bidireccional. Por otro lado, Pinheiro en [Pinheiro 04] introduce otra clasificación: Inter-requirements traceability (habilidad de rastreo entre requisitos dependientes) y Extra-requirements traceability (habilidad de rastreo entre requisitos y otros artefactos). Como se observa, Wieringa y Davis hacen referencia a este último tipo de trazabilidad. Para realizar un cambio en un requisito es necesario poder rastrear tanto interna como externamente el requisito, con el fin de evaluar adecuadamente tanto los requisitos dependientes como los componentes que deberán adaptarse y, en ciertos casos, es importante analizar el origen del requisito.

La recolección y mantenimiento de la información de rastreo es de muy alto costo. Por lo tanto, se deben tener políticas que indiquen qué tipo de rastreos se realizarán y cómo se mantendrá dicha información. Como indica Kotonya & Sommerville [Kotonya 98], la información de rastreo que más habitualmente se mantiene en la práctica es la que corresponde a Inter-requirements traceability y a Forward Traceability desde el documento de requisitos al diseño.

A continuación se extrae la definición dada por Pinheiro & Goguen [Pinheiro 96] por su claridad y amplitud: “La trazabilidad de los requisitos se refiere a la habilidad de definir, capturar y seguir las pistas dejadas por los requisitos sobre otros elementos del ambiente de desarrollo del software y las pistas dejadas por dichos elementos sobre los requisitos”.

La trazabilidad no sólo se utiliza para administrar los cambios en los requisitos, sino que también es de fundamental ayuda para la verificación y validación de los requisitos y para el control del proceso de desarrollo [Palmer 96] [Davis 99], pues facilita detectar conflictos utilizando los vínculos establecidos entre los elementos rastreables, posibilita asegurar que decisiones tomadas

avanzado el desarrollo sean consistentes con decisiones tempranas, y permite verificar que todos los requisitos han sido implementados en el software, entre otros usos. A pesar de los múltiples propósitos que cubre la trazabilidad de los requisitos, muchas veces sólo es aplicada parcialmente. Esto es debido, por un lado, a su alto costo de producción y mantenimiento (gran diversidad de entidades rastreables) y, por otro lado, a la imperiosa necesidad de contar con herramientas automatizadas que permitan implementar adecuadamente la trazabilidad.

2. PERSPECTIVA DEL PROYECTO

En este proyecto se estudiarán mejores formas de proveer a la rastreabilidad de los requisitos de software, es decir, qué aspecto del negocio generó una determinada funcionalidad del software y cómo está resuelta en el software una determinada demanda del cliente o usuario.

Cuando se está en presencia de dos modelos sucesivos en el proceso de desarrollo de software, cualquiera sean estos, es natural pensar que las trazas de un modelo a otro pueden representarse por medio de una matriz de trazas o un mecanismo similar. Este tipo de soluciones ignora que la naturaleza de la traza es tal que hay información, muchas veces muy relevante, relacionada con cada vínculo entre modelos. Por otra parte, esa visión de dos modelos sucesivos es en casi todos los casos falsa ya que para la construcción de un modelo o producto en el proceso de desarrollo de software se debe recurrir a información de diversas fuentes, lo que hace necesario pensar en cubos y eventualmente hipercubos de trazas. Esto es realmente inviable en casos prácticos. El recurso al que se suele apelar consiste en utilizar vínculos explícitos para representar las trazas. Estos modelos basados en vínculos carecen de la expresividad necesaria para representar la complejidad de las trazas en el proceso de requisitos y peor aún contaminan los modelos en los que los vínculos están empotrados de tal manera que aquellos llegan a ser prácticamente ilegibles. Debe notarse que habitualmente los modelos ya tienen vínculos de naturaleza semántica por lo que se debería proceder además a diferenciar los vínculos por su propósito.

Visto con la perspectiva de las bases de datos se puede notar que la presencia de modelos relacionados por vínculos es, en alguna medida, la reintroducción de las ideas de las bases de datos jerárquicas y en red con punteros empotrados al dominio de la Ingeniería de Requisitos en particular y a la Ingeniería de Software en general.

El modelo de trazas que se aspira a diseñar dará apoyo a un proceso de requisitos específico [Leite 04] que se basa en modelos en lenguaje natural: Léxico Extendido del Lenguaje (glosario del universo de discurso), Escenarios Actuales (situaciones observables en el proceso del negocio), Escenarios Futuros (situaciones

esperadas en el negocio con la incorporación del nuevo software) y Especificación de Requisitos (descripciones explícitas de cada requisito del software).

Los escenarios futuros sirven de ancla para la pre y post trazabilidad, permitiendo el rastreo de los requisitos hacia sus orígenes (vinculando los escenarios futuros con los escenarios actuales y con el léxico extendido del lenguaje) y hacia el diseño y el código. Son un medio que facilita la gestión de los cambios en los requisitos a lo largo del ciclo de vida del software.

2.1. OBJETIVOS PROPUESTOS

Los objetivos propuestos para este proyecto son:

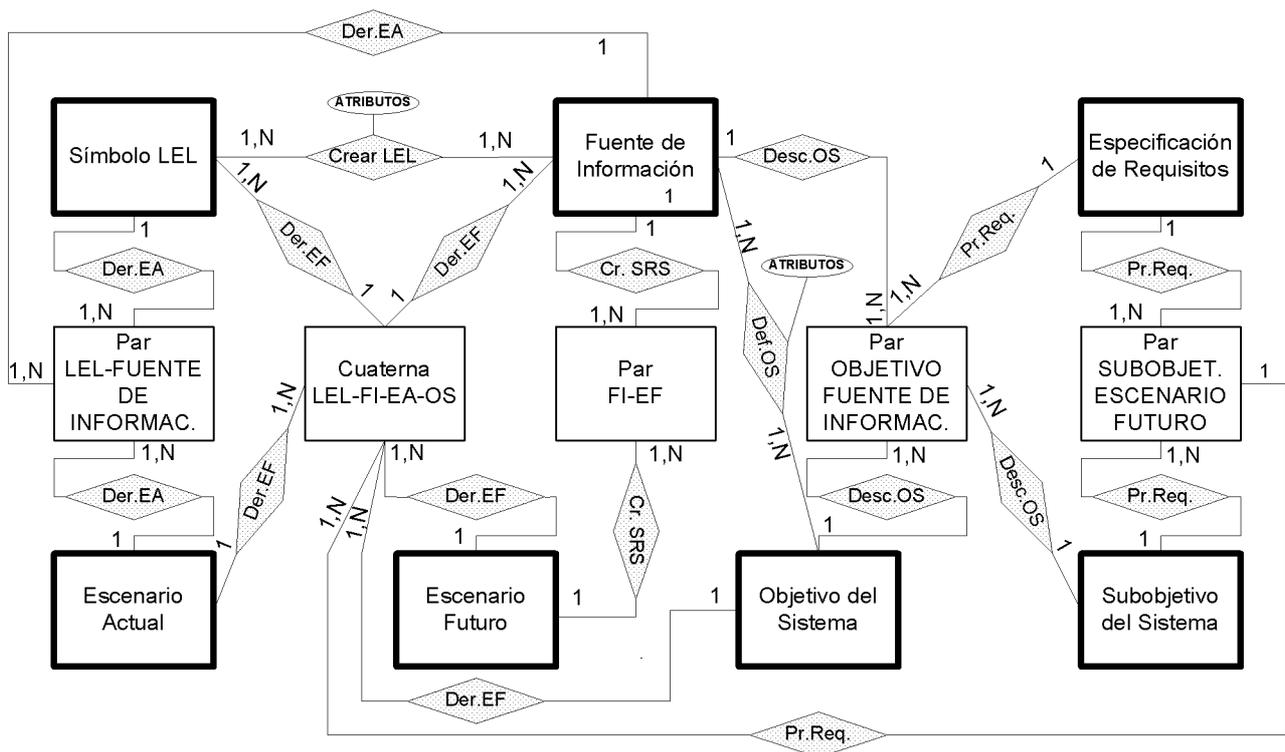
- Proveer un modelo de trazas que no interfiera con los modelos de requisitos.
- Definir un mecanismo de rastreo que brinde facilidad de navegación.
- Proveer un modelo de versionado.

Siendo los objetivos específicos los siguientes:

- Definir el modelo de trazas a ser utilizado (determinar qué componentes del modelo de trazas pueden ser capturados automáticamente).
- Definir el proceso de captura de trazas.
- Definir el modelo y el proceso de navegación con trazas.
- Definir el modelo de versionado para los modelos de requisitos.
- Detectar las relaciones existentes entre el modelo de trazas y el modelo de versionado.

3. ESTADO DE AVANCE

Se ha definido una primera versión del modelo de trazas, que se presenta en la Figura 1. Los componentes del proceso de requisitos se han modelado separando los datos visibles y manipulables por el usuario final de aquellos necesarios para el modelo de trazas. Estos últimos son los representados en la Figura 1 como entidades con marco grueso; no son los componentes propiamente dichos. Las entidades con marco fino corresponden a trazas de creación entre los distintos modelos de requisitos.



- (Crear LEL) - Trazas de la creación del LEL
- (Der.EA) – Trazas de la derivación de Escenarios Actuales
- (Def.OS) Trazas de la definición del Objetivo del Sistema
- (Pr.Req.) - Trazas de la priorización de los requisitos
- (Der.EF) – Trazas de la derivación de Escenarios Futuros
- (Cr.SRS) – Trazas de la creación del SRS
- (Desc.OS) - Trazas de descomposición del objetivo del sistema

Figura 1 – Primera versión del modelo de trazas

Se han definido los atributos de las entidades del tipo Trazas. Se está estudiando los atributos a asignar a la entidad Fuentes de Información, la que por su variedad

es probable que se requiera construir una jerarquía con la misma.

Se planifica estudiar en detalle los procesos de construcción del LEL, derivación de escenarios actuales, entre otros, desde el punto de vista de las trazas para proveer mecanismos automáticos o semi automáticos de registro de las mismas.

4. FORMACION DE RECURSOS HUMANOS

El presente proyecto es parte de la tesis doctoral "Modelado del registro de trazas en la Ingeniería de Requisitos" que está desarrollando la Ing. Andrea Vera en la UNLP.

5. REFERENCIAS

- [CMMI 06] Software Engineering Institute, "Capability Maturity Model Integration", CMMI-DEV v1.2, CMU/SEI-2006-TR-008, Carnegie Mellon University, 2006, <http://www.sei.cmu.edu/cmmi/>.
- [Conradi 98] Conradi, R., Westfechtel, B., "Version Models for Software Configuration Management", ACM Computing Surveys, Vol.30, 1998, pp.232-282.
- [Crnkovic 99] Crnkovic, I., Funk, P., Larsson, M., "Processing Requirements by Software Configuration Management", 25th Euromicro Conference (EUROMICRO'99), IEEE Computer Society, Milán, Italia, Vol.2, Septiembre 1999, pp.2260-2265.
- [Davis 99] Davis, A., Leffingwell, D., "Making Requirements Management Work For You", Crosstalk, The Journal of Defense Software Engineering, Vol.12, N°4, Abril 1999.
- [Gotel 94] Gotel, O.C.Z., Finkelstein, A.C.W., "An analysis of the requirements traceability problem", ICRE'94, First IEEE International Conference on Requirements Engineering, IEEE Computer Society Press, Colorado Springs, Abril 1994, pp.94-101.
- [IEEE Std 830-1998] IEEE Recommended Practice for Software Requirements Specifications (ANSI), IEEE, Nueva York, 1998.
- [Jarke 98] Jarke, M., "Requirements tracing", Communications of the ACM, Vol.41, N°12, Diciembre 1998, pp.32-36.
- [Kotonya 98] Kotonya, G., Sommerville, I.: Requirements Engineering: Processes and Techniques. John Wiley & Sons, 1998.
- [Leite 97] Leite, J.C.S.P., "Software Evolution, The Requirements Engineering View", keynote address en anales de XXVI JAIHO - SoST'97 Simposio en Tecnología de Software, Buenos Aires, Argentina, Agosto 1997, pp.21-23.
- [Leite 04] Leite, J.C.S.P., Doorn, J.H., Kaplan, G.N., Hadad, G.D.S., Ridao, M.N., "Defining System Context using Scenarios", en el libro "Perspectives on Software Requirements", Kluwer Academic Publishers, EEUU, ISBN: 1-4020-7625-8, capítulo 8, 2004, pp.169-199
- [Palmer 96] Palmer, J.D., "Traceability", en Software Engineering, editores M. Dorfman y R.H. Thayer, IEEE Computer Society Press, 1996, pp.266-276. Reimpreso en "Software Requirements Engineering", editores R.H. Thayer y M. Dorfman, IEEE Computer Society Press, 2° edición, Los Alamitos, CA, 1997, pp.364-374.
- [Pinheiro 04] Pinheiro, F.A.C., "Requirements Traceability", en el libro "Perspectives on Software Requirements", Kluwer Academic Publishers, Estados Unidos, ISBN: 1-4020-7625-8, capítulo 5, 2004, pp.91-113.
- [Pinheiro 96] Pinheiro, F.A.C., Goguen, J.A., "An object-oriented tool for tracing requirements", IEEE Software, Special issue of papers from ICRE'96, Vol.13, N°2, Marzo 1996, pp.52-64.
- [Sawyer 01] Sawyer, P., Kotonya, G., "Software Requirements", SWEBOOK, Guide to the Software Engineering Body of Knowledge, editores P. Bourque y R. Dupuis, IEEE Computer Society, Los Alamitos, CA, capítulo 2, 2001, pp.31-56, IEEE Trial Version 1.00, http://www.swebok.org/stoneman/trial_1_00.html.
- [Wieringa 95] Wieringa, R.J., "An introduction to requirements traceability", Reporte Técnico IR-389, Faculty of Mathematics and Computer Science, University of Vrije, Amsterdam, Septiembre 1995.