

Traceability en la etapa de elicitación de requerimientos

Leandro Antonelli (LIFIA Facultad de Informática – UNLP), lanto@sol.info.unlp.edu.ar

Alejandro Oliveros (Departamento de Ingeniería – UNLM), oliveros@fibertel.com.ar

Introducción

El desarrollo de un sistema de software requiere entender el macrosistema en el que será utilizado. Para ello los ingenieros de requerimientos utilizan métodos para comprender y modelar los elementos del macrosistema. **Client-oriented baseline** (Leite 1995) contribuye a la solución de este problema: utilizando el lenguaje del cliente, lo que mejora la comunicación y validación de las descripciones elicidadas. (Leite 1997b)

El primer componente del client-oriented baseline, es el **Léxico Extendido del Lenguaje (LEL)**. El lexico extendido del lenguaje es una representación de los símbolos del lenguaje. Cada símbolo, palabra o frase, es descrito mediante *nociones e impactos*. La *noción* es una descripción al estilo del diccionario tradicional. El *impacto* identifica la relación con los demás símbolos. Noción e impacto se deben describir maximizando el uso de símbolos pertenecientes al LEL (Leite 1993). Este modelo se centra en la idea (principio de circularidad) de que una descripción circular de los términos del lenguaje mejora la comprensión. El LEL es principalmente utilizado para elicitar el vocabulario de la aplicación directamente de los actores del universo de discurso de la aplicación (Leite 1993).

El conjunto de **Escenarios** es otro elemento del baseline y se describen gracias a los siguientes conceptos (Leite 1997):

Un escenario es una descripción de situaciones en el ambiente. Un escenario describe situaciones enfatizando en la descripción del comportamiento. Los escenarios utilizan lenguaje natural como su representación básica. Los escenarios son naturalmente relacionados al LEL, existiendo heurísticas que permiten derivarlos del LEL. La estructura de un escenario esta compuesta por *objetivo, contexto, recursos, actores y episodios*. *Objetivo, contexto, recursos y actores* son oraciones declarativas. *Episodios* es un conjunto de oraciones de acuerdo a un lenguaje muy simple que posibilita la descripción de comportamiento.

La **visión hipertextual** es otro elemento del baseline. Cada símbolo en el LEL y cada escenario es naturalmente ligado a otros Escenarios y símbolos del LEL.

Finalmente, el último componente es la **visión de configuración**. El LEL y los escenarios están sujetos a un control de versión.

Con este modelo se establecieron un conjunto de **heurísticas** que permitieron derivar **tarjetas CRC** (Wilkinson 1995) a partir del LEL y escenarios (Leonardi 1997). Estas tarjetas CRC (CRC de análisis) identifican entidades que son relevantes en el dominio de la aplicación (Wilkinson 1995). Las CRC se componen de *responsabilidades y colaboraciones*, *responsabilidades* son las actividades que deben llevar a cabo y las *colaboraciones* son los servicios que piden o prestan a otras entidades. Las CRC constituyen el primer paso dentro del análisis de clases.

BaselineMentor: la aplicación

Una aplicación, llamada Baseline Mentor (Antonelli 1999), se construyó para brindar soporte durante la fase de ingeniería de requerimientos usando la aproximación de client-oriented requirements baseline. Un client oriented requirements baseline para una dominio de aplicación dado, se denomina proyecto dentro de la herramienta. BaselineMentor puede administrar varios proyectos simultáneamente. La visión de configuración es capturada a través de versiones, en las cuales se dividen los proyectos. Cada versión tiene tres conjuntos. Un conjunto de símbolos del LEL, un conjunto de escenarios y un conjunto de tarjetas CRC. Las entradas del LEL y los escenarios son ingresados manualmente a BaselineMentor, en cambio las tarjetas CRC son obtenidas automáticamente. Dentro de cada versión, las tarjetas CRC son derivadas de las entradas del LEL y los

escenarios pertenecientes a la misma versión. El proceso de derivación es un algoritmo simplificado de la estrategia de derivación del modelo lógico orientado a objetos descrito en (Leite 1997).

Rasgos generales de la herramienta:

- Cuenta con funciones cortar, copiar y pegar para agilizar la edición
- Hay algunas expresiones con formato fijo, dentro de episodios, para las cuales se escribe un template, a través del menú pop up, dejando blancos para completar.
- Como la construcción de LEL y escenarios, se rige por el principio de circularidad, se obtiene un hipertexto, el cual puede ser navegado. Los links se muestran de distinta forma. Un link hacia LEL es rojo-cursiva, hacia escenario es azul-negrita y hacia CRC es gris.
- Durante la edición de LEL o escenarios, si se escribe una palabra que ya esta dentro del conjunto de LEL, se convierte en link automáticamente. Por el contrario, si la palabra no es un símbolo del LEL, pero debería serlo, del menú contextual se puede elegir una opción para convertirlo en link y crear la entrada del LEL correspondiente.
- Después de cambiar un LEL o escenario, la herramienta chequea la consistencia de los links, los crea de ser necesario. Si cambia la identificación, se modifican los anchors correspondiente. Si se elimina algún elemento, se deshacen los links.
- Durante la navegación, cuando un link es visitado, se torna mas claro
- El contexto navegacional (Schwabe 1995) es un recurso para hacer más sencillo el proceso de navegación. Los episodios dentro de un escenario, se pueden describir como escenarios en sí. Pues, si un escenario tiene varios episodios y el usuario desea explorar cada uno, él deberá hacer click en el primer episodio, leerlo, presionar el botón de atrás, presionar el segundo episodio y así seguir. Sin embargo, la aplicación reconoce estos contextos navegacionales, y cuando un link que pertenece a un contexto navegacional es explorado, se muestra la lista con todos los links pertenecientes al mismo contexto navegacional, con el fin de recorrerlos en forma sencilla pulsando solamente el botón siguiente.

Traceability

El propósito de la Traceability es describir y rastrear la vida de los requerimientos, en ambas direcciones, hacia la implementación (forward) y hacia los orígenes (backward) (Gotel 1994). Traceability se puede dividir en dos partes:

- Traceability pre-RS (Requirements Specification) concierne a todos los aspectos de la vida de los requerimientos anteriores a su inclusión en la RS.
- Traceability post-RS concierne a todos los aspectos de la vida de los requerimientos que resultan de su inclusión en la RS.

Durante el proceso de desarrollo de soft, las distintas personas implicadas (stakeholders), necesitan traceability por diversas razones (Wieringa 1995):

Gerente de proyectos: Estimar el impacto de los cambios en los requerimientos; descubrir tempranamente conflictos entre los requerimientos; tener control acerca de que requerimientos están satisfechos y cuales faltan implementar aún.

Cliente: Saber que función del sistema implementa que requerimiento, y conocer como testarlo para verificar su funcionamiento

Diseñador: Verificar que un diseño satisface los requerimientos, estimar el impacto de los cambios de requerimientos sobre el diseño, entender las razones de porque ciertos diseños fueron aceptados y otros rechazados aún cuando el diseño se produjo hace mucho tiempo por otro diseñador no presente (estas razones pueden relacionar decisiones de diseño a requerimientos no funcionales), reutilizar componentes en otros proyectos, porque los supuestos bajo los cuales el componente trabajará están registradas.

Responsable del mantenimiento: Estimar impacto de cambios de un requerimiento a otros requerimientos (descubrir conflictos, dependencias), estimar el impacto de cambios de los requerimientos en la implementación

En la Ingeniería de Requerimientos (RE) ha crecido el interés de la investigación de Traceability de Requerimientos (RT), por las razones antes enumeradas. A pesar de muchos avances, RT sigue siendo un área en el que quedan muchos problemas por resolver, los cuales se atribuyen a un inadecuado traceability pre-RS. Profesionales del área requieren técnicas para registrar y rastrear información relacionada a la producción y revisión de RS.

Traceability Pre-RS y BaselineMentor

Las heurísticas que se han diseñado por sobre client-oriented baseline y se han implementado en BaselineMentor, consisten en la propagación de cambios sobre el modelo en forma automática. Es decir, frente a la modificación de algún LEL o escenario, propagar los cambios a los demás elementos (LEL, escenario o CRC). Estas heurísticas son Pre-RS, aunque comienzan en el LEL por lo que no capturan otros elementos del Universo del Discurso.

Es importante cubrir todas las alternativas de modificaciones posibles, para este propósito se ha hecho un análisis exhaustivo y sistemático de éstas alternativas. En primer lugar se elaboró la lista de todos los elementos que conforman client-oriented baseline (eje vertical del cuadro siguiente). Luego se enumeró todas las posibles acciones sobre estos elementos (eje horizontal del cuadro siguiente).

		agregar		agregar		Modificar		eliminar		eliminar	
		texto	link	texto	link	Texto	link	texto	link	texto	link
LEL	Sinonimos										
	Nocion										
	Impactos										
Escenario	Titulo										
	Objetivo										
	Contexto										
	Actores										
	Recursos										
	Episodios										
Tarjeta CRC	Nombre										
	Responsabilidades										
	Colaboraciones										

En último termino se analizó en particular cada una de las intersecciones que quedan determinadas en el cuadro, describiendo las acciones a llevar a cabo para propagar los cambios.

Como ejemplo tómesese el caso de agregar un link en actores de escenarios. Si se agrega un link tiene que ser un link a una entrada del LEL, ya que no hay otra alternativa. Este hecho es condición suficiente como para crear una CRC (de acuerdo a las heurísticas de derivación), así que, si la CRC no existe aún, se la debe crear a partir del LEL correspondiente. La creación de la CRC consiste en darle un nombre y responsabilidades, que se obtienen de la entrada del LEL, y definir las colaboraciones. Para este propósito, se deben revisar todas los escenarios, buscando en los episodios si aparece la CRC recién creada y en colaboración con que otras.

Futuras investigaciones

Hasta ahora la herramienta ha sido probada capturando la información de los LEL y Escenarios construidos manualmente, los resultados preliminares resultan satisfactorios. Otra línea de experimentación se desarrolla con el uso en trabajos de estudiantes de posgrado. El modelo client-oriented baseline genera las CRC, una próxima línea de trabajo es establecer la relación con otros enfoques de obtención de requerimientos. En primer lugar se enfocará la relación con el UML (Fowler

1997): ¿se pueden derivar Use Case del client-oriented baseline o éste podrá reemplazar a los Use Case en el UML?. En términos generales se trata de investigar la relación que puede establecerse entre el UML y el client-oriented baseline. Un tema de estudio posterior es la obtención de una especificación de requerimientos. Tanto LEL como escenarios describen el entorno del sistema, pero no determinan los requerimientos de él.

Referencias

- Antonelli, L., Oliveros, A., Rossi, G.: Baseline Mentor, An application that derives CRC Cards from Lexicon and scenario. XXVIII JAIO. WER'99. Argentina. 1999.
- Fowler, M, Scott, K.: UML Distilled. Applying the Standard Object Modeling Language, Addison-Wesley, 1997
- Gotel, O. C. Z., Finkelstein, A. C. W.: An Analysis of the Requirements Traceability Problem. IEEE International Conference on Requirements Engineering. Los Alamitos. California. IEEE Computer Society Press, April 1994, pp. 94-101. (1994)
- Leite, J.C.: Eliciting Requirements Using a Natural Language Based Approach: The Case of the Meeting Scheduler Problem. Supplied by author (1993)
- Leite, J.C., Leonardi, C., Rossi, G.: Deriving Object Oriented Specification from External Scenarios. Supplied by author (1997)
- Leite, J.C., Oliveira, A.: A Client Oriented Requirements Baseline. Proceedings of RE 95'. Second IEEE International Symposium on Requirements Engineer. UK. IEEE Computer Society Press (1995) 108-115
- Leite, J.C., Rossi, G., et al.: Enhancing a Requirements Baseline with Scenarios. Proceedings of RE 97'. IEEE Third International Requirements Engineering Symposium, IEEE Computer Society Press (1997b) 44-53
- Leonardi, C., Maiorana, V., Balaguer, F.: Una estrategia de Análisis Orientada a Objetos basada en escenarios. Actas de II Jornadas de Ingeniería del software. JIS97. Dpto. de Informática. Universidad del País Vasco. San Sebastián. pp. 87-100. (1997)
- Schwabe, D., Rossi, G.: An Object Oriented Hypermedia Design Model. Communication of ACM Vol 38 Nro 8 (1995) 45-46
- Wieringa, R.: An Introduction to Requirements Traceability. Reporte Técnico. Faculty of Mathematics and Computer Science. Vrije Universiteit. Esprit Project 2RARE. November 1. (1995)
- Wilkinson, N. M.: Using CRC Cards: An Informal Approach to Object-Oriented. AT&T Bell Laboratories. (1995)