

Especificación de Requisitos: de lo Informal a lo Formal

Aristides Dasso, Ana Funes
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950
San Luis, Argentina
{arisdas, afunes}@unsl.edu.ar

CONTEXTO

Este trabajo de investigación se encuentra enmarcado dentro del Proyecto de Incentivos código 22/F822: “Ingeniería de Software: Conceptos, Métodos y Herramientas en un contexto de Ingeniería de Software en Evolución” de la Universidad Nacional de San Luis, Argentina, en una de las líneas: del mismo: “Métodos Formales y Prototipos Evolutivos”, Dentro del contexto de desarrollo de herramientas, esta investigación tiene como objetivo el concretar una especificación formal a partir de una especificación realizada en lenguaje natural.

RESUMEN

Esta investigación se enmarca dentro del campo de la Ingeniería de Software, particularmente en lo que hace a la aplicación de metodologías formales y semi-formales de especificación de requisitos. A partir de una especificación de requisitos realizada en un lenguaje natural pasamos a una especificación en un lenguaje formal, por medio de reglas de transformación, inicialmente heurísticas, para en el futuro transformarlas en una aplicación de soporte al diseño de sistemas de software.

Palabras clave: Ingeniería del Software. Métodos Formales. Especificación. Requisitos. Lenguaje Natural.

1. INTRODUCCION

En las etapas iniciales del desarrollo de un sistema de software muchas veces nos encontramos con especificaciones informales, escritas en un lenguaje natural. Esto crea una brecha grande cuando queremos pasar de esta especificación informal a otra más formal.

Aquí presentamos una técnica para crear una especificación formal a partir de una descripción informal escrita en un lenguaje natural.

Hay muchas técnicas para extraer información útil de documentos escritos en un lenguaje natural que nos permiten construir documentos más precisos y formales, los cuales expresan los requerimientos del sistema de manera menos ambigua.

Algunas de estas técnicas incluyen la creación de los casos de uso del sistema. Un caso de uso es una secuencia de acciones que proveen una cierta funcionalidad a un actor del sistema. Un caso de uso describe una manera en la cual un actor del mundo real interactúa con el sistema. Cada una de estas formas de interacción es un evento al cual el sistema deberá dar una respuesta. Podemos entonces establecer una correspondencia entre los casos de uso del sistema y una lista de eventos del sistema.

Las listas son más simples que los casos de uso y muchas veces son uno de los primeros pasos en la construcción de un sistema.

Al mismo tiempo, cada caso de uso del sistema puede ser descrito tanto en una manera informal como formal.

Por otro lado, sabemos de los beneficios de contar con especificaciones formales en las primeras etapas de desarrollo del sistema.

Nuestro trabajo consiste en desarrollar una técnica para obtener a partir de un conjunto de casos de uso, bajo la forma de lista de eventos, a una especificación formal inicial escrita en RSL (the RAISE Specification Language) [RLG92, RLM95], consistente en signaturas de funciones de más alto nivel del sistema y tipos expresados como sorts. Cabe destacar que la misma técnica puede ser aplicada para cada caso de uso, para de esta manera derivar no sólo las signaturas sino también el cuerpo de las funciones de más alto nivel del sistema.

Inicialmente las reglas son heurísticas, pero las mismas se incorporarán a una herramienta que permita el diseño asistido de desarrollo de sistemas.

Como entrada se tienen casos de uso bajo la forma de una lista de eventos. Estos eventos son procesados empleando una herramienta de análisis del lenguaje natural para reproducir los mismos en un formato estructurado para el cual es posible dar reglas de transformación para su traducción a signaturas de funciones y tipos en RSL.

De esta manera incorporamos, en las fases iniciales del desarrollo de un sistema, dos métodos para la especificación de un sistema: primero empleando una técnica no rigurosa como son los casos de uso y las listas de eventos, para luego pasar de estas a una especificación formal empleando un método formal como es RAISE.

Para procesar el lenguaje natural proponemos el empleo de FreeLing [TALP], que es una suite abierta de analizadores de lenguaje natural. FreeLing usa EAGLES [EAGLS] tags para su análisis morfológico y también diccionarios de WordNet [WrdNet].

2. LINEAS DE INVESTIGACION y DESARROLLO

El uso de Métodos Formales en el desarrollo de sistemas de software garantiza la precisión así como la no ambigüedad, particularmente en las especificaciones del sistema. Sin embargo, uno de los principales problemas de los métodos formales es que sus especificaciones son difíciles de entender para la mayoría de los usuarios de un sistema.

Por otro lado estos usuarios pueden fácilmente entender una especificación desarrollada en un lenguaje natural, aún cuando la misma sea bajo la forma de casos de uso o lista de eventos.

Desarrollar una herramienta de apoyo al diseño de software, especialmente para las primeras etapas del mismo, y especialmente que facilite el pasaje de manera automática a partir de los casos de uso del sistema a una primera especificación formal es uno de los principales objetivos del trabajo.

Este trabajo se lleva a cabo dentro de la línea de “Métodos Formales y Prototipos Evolutivos” del proyecto de incentivos de la Universidad Nacional de San Luis, código 22/F822: “Ingeniería de Software: Conceptos, Métodos y Herramientas en un contexto de Ingeniería de Software en Evolución”

3. RESULTADOS OBTENIDOS/ESPERADOS

Debido a la complejidad y la ambigüedad inherente de los lenguajes naturales, así como su riqueza, la transformación propuesta es, por ahora, más heurística que automática.

Sin embargo, además de continuar trabajando sobre las reglas heurísticas de transformación, esto es sólo una primera parte del trabajo, ya que, sobre la base de estas reglas, se pretende realizar el desarrollo de una herramienta que lleve a cabo la transformación si bien de manera no completamente automática, de forma semi-automática con algún tipo de asistencia por parte del usuario en aquellos casos que resulten complejos de resolver ya que para el diseñador de sistemas de software, quien se encuentra comprometido con el

sistema que está siendo analizado, determinar, por ejemplo, dominio y co-dominio de las funciones no debería ser un problema siendo que él conoce la semántica asociada, especialmente si cuenta con una herramienta que le muestre las distintas alternativas.

Cabe destacar que sobre la base de este primer esbozo de requerimientos, el desarrollador cuenta con un primer modelo formal de referencia que le permitirá continuar desarrollando el sistema ahora ya sobre una base formal.

En síntesis, se espera: en una primera etapa desarrollar reglas heurísticas de transformación, las que contribuirán en la segunda etapa al desarrollo de una herramienta que inicialmente sirva como apoyo a los diseñadores del sistema en la producción de las primeras especificaciones formales.

4. FORMACION DE RECURSOS HUMANOS

Dado que esta línea de investigación se encuentra en una etapa inicial, creemos que una profundización en el descubrimiento y elaboración de las reglas de transformación es la base para la realización de una tesis de maestría.

Se espera, asimismo, desarrollar una herramienta que permita la transformación propuesta de forma automática y que este desarrollo sea realizado por un equipo de alumnos como parte de su trabajo de grado.

5. BIBLIOGRAFIA

[COC98] Alistair Cockburn, Basic Use Case Template.

http://alistair.cockburn.us/index.php/Resources_for_writing_use_cases#Use_Case_Templates. Retrieved January 2008.

[CRaG08] CRaG Systems, Use Case Document.dot.

<http://www.cragssystems.com/SFRWUC/docs/UseCaseDocument.dot>. Retrieved January 2008.

[DFa198] A. Dasso, A. Funes et al., "An Application of Formal Methods: Specifying a

Credit Card System", proceedings of IASTED 98, Conference on Software Engineering, Las Vegas, USA, 1998.

[DFa199] A. Dasso, A. Funes, C. George, G. Montejano, D. Riesco, R. Uzal; "A Technique to Induce Formal Specification at the Early Stages of System Development", Proceedings of the 3rd Annual International Conference on Software Engineering and Applications, IASTED, Scottsdale, Arizona, 6 al 8 de Octubre de 1999. IASTED/ACTA Press, Anaheim-Calgary-Zürich. ISBN: 0-88986-273-7. pp 344-347.

[EAGLS]<http://www.ilc.cnr.it/EAGLES96/home.html>

[OMG05] Object Management Group, Software Process Engineering Metamodel Specification. January 2005.

[prim03] primaryview.org, Scenarios, Sequence Diagrams, and Narrative Text. February 2003.

<http://www.primaryview.org/UML/Scenarios.html>

[Rea08] ReadySET Pro, Use Case Tutorial. <http://readyssetpro.com/whitepapers/usecasetutorial.html>. Retrieved January 2008.

[RLG92] The RAISE Language Group, *The RAISE Specification Language*, Prentice Hall, 1992.

[RLM95] The RAISE Method Group, *The RAISE Development Method*, Prentice Hall, 1995.

[TALP] <http://garraf.epsevg.upc.es/freeling/>

[WrdNet] <http://wordnet.princeton.edu/>