ab

Filtrado de Especificaciones de Software Escritas en Lenguaje Natural

Núria Castell y Àngels Hernández
Departament de Llenguatges i Sistemes Informàtics
Secció d'Intel.ligència Artificial
Universitat Politècnica de Catalunya
Campus Nord - Edificio C5. Barcelona 08071. España.
email: castell,ahernandez@lsi.upc.es
Teléfono 4017015

Abstract. La fase de especificación es una de las más importantes y menos controladas en el proceso de desarrollo de software. Hemos concebido SAREL [1] (Sistema de Ayuda para la Redacción de Especificaciones de Software escritas en Lenguage Natural) como una herramienta que mejore la fase de especificación. SAREL es una continuación del programa de investigación y desarrollo llamado LESD (Ingeniería Linguïstica para el Diseño de Software). El propósito de SAREL [2] es ayudar a los ingenieros en la creación de especificaciones de software escritas en Lenguaje Natural. Está dividido en tres módulos: el primero controla los requisitos de acuerdo con las normas de redacción, el segundo obtiene una representación conceptual utilizando la Base de Conocimiento, y el tercero lleva a cabo una serie de análisis teniendo en cuenta las siguientes propiedades de calidad: consistencia, completitud, trazabilidad, verificabilidad y modificabilidad. Una vez el requisito ha sido etiquetado como correcto, su representación conceptual es añadida a la Base de Requisitos.

1 Introducción

El proceso de desarrollo de software comienza generalmente con la fase de especificación. En esta etapa es muy importante controlar la calidad de la especificación para poder detectar posibles errores tan pronto como sea posible. La corrección de errores en las fases de desarrollo y implementación implica dedicar más tiempo y esfuerzo que en la fase de especificación. Ésta es la razón por la cual muchos diseñadores tratan cada vez más de identificar los posibles errores en las fases más tempranas del desarrollo de software.

Durante la fase inicial, las especificaciones de software para sistemas complejos dan lugar a un conjunto de documentos voluminosos que la mayoría de ocasiones están escritos en lenguaje natural. La redacción de la documentación está

[1] La versión original de este artículo, en inglés, ha sido publicado por Springer-Verlag en la colección Lecture Notes on AI [11].

[2] Este trabajo ha sido parcialmente subvencionado por CICYT (TIC93-420) y por CIRIT (GRQ1995-566)

guiada por normas de redacción que definen las restricciones lingüísticas necesarias que deben satisfacer las especificaciones. Estas normas son de dos tipos: aquéllas relacionadas con el uso del lenguaje natural en general (por ejemplo [3] y [4]); y aquéllas que están basadas en restricciones terminológicas relacionadas con un dominio en particular (por ejemplo, las normas ESA - Agencia Europea del Espacio -). Ambas restringen el uso del lenguaje natural a través de un conjunto de reglas que limitan varias irregularidades (polisemia, paráfrasis, ambigüedad, vaguedad..). A pesar de que las normas definen lingüísticamente restricciones precisas, frecuentemente no son seguidas y ello dificulta la deteccción en fases posteriores de las consecuencias posibles de tales infracciones. Además de las restricciones lingüísticas, las normas también incluyen restricciones de Ingeniería del Software relacionadas con los factores de calidad de la especificación, tales como la consistencia, completitud, trazabilidad, modificabilidad y verificabilidad.

El propósito del sistema que estamos desarrollando (SAREL) es ayudar al ingeniero en la redacción de especificaciones en lenguaje natural teniendo en cuenta las normas de redacción y los factores de calidad de software.

Otros sistemas que tratan los problemas asociados con la fase de especificación son RA [15] [16], el sistema de desarrollo FRORL [19] y OICSI [13] [17]. El Requirements Apprentice (RA) ayuda al analista humano en la creación y modificación de requisitos de software. El sistema de desarrollo FRORL facilita la especificación, análisis y desarrollo de un sistema de software. El OICSI es un sistema prototipo que ayuda al analista en el proceso de adquisición, elicitación y validación de sentencias. Aunque los objetivos de los sistemas anteriores varían ligeramente de los objetivos del sistema SAREL, forman un buen marco de referencia para nuestro trabajo de investigación. Un estudio comparativo se encuentra en [12].

En la sección 2, describimos nuestro sistema (SAREL) de forma general. En las secciones 3, 4 y 5 se describen cada uno de los módulos de SAREL. El artículo termina con una discursión de los objetivos de nuestro trabajo de investigación futuro.

2 SAREL: Un Sistema de Ayuda a la Redacción de Especificaciones de Software en Lenguaje Natural

El sistema SAREL es una continuación del programa de investigación y desarrollo denominado LESD (Ingeniería Lingüística para el Diseño de Software). Este proyecto fue fomentado por el centro ARAMIIHS de Toulouse (Francia) y fue llevado a cabo por investigadores españoles y franceses. El propósito de LESD [5], [6], [18] era desarrollar herramientas computacionales que (1) permitieran una interpretación conceptual de las especificaciones aeroespaciales de software preliminares o funcionales escritas en inglés; (2) permitieran la evaluación de los factores de calidad mediante algoritmos de razonamiento aplicados a la representación conceptual; y (3) ayudaran a los ingenieros en el manejo de la documentación.

En este contexto, el objetivo principal de SAREL es ayudar al ingeniero en la creación de especificaciones de software escritas en lenguaje natural. Para ser exactos las especificaciones están escritas en inglés dado que es el idioma más común en el dominio aeroespacial (el dominio de LESD). Los ejemplos que se presentan en este artículo han sido extraídos de documentos de un sistema aeroespacial. Una versión preliminar de nuestro sistema fue descrita en [12]. El proceso de ayuda, dividido en varios pasos, valida cada uno de los requisitos introducidos por el ingeniero teniendo en cuenta las normas de redacción (por ejemplo [3], [4]) y los factores de calidad [10]. Este proceso construye de manera incremental una representación conceptual de la especificación. Los controles que aparecen en la figura 1 pueden ser agrupados en tres módulos: el Módulo de Refinamiento de Estilo, el Módulo de Refinamiento Conceptual y el Módulo de Control de Calidad de Software. Una vez que el requisito ha sido validado y es correcto, su representación conceptual es añadida a la Base de Requisitos.

3 Módulo de Refinamiento de Estilo

Este módulo controla los requisitos de acuerdo con las normas de redacción y está compuesto por cuatro fases: análisis léxico, análisis sintáctico-semántico, control de ambigüedad y control de simplicidad. Este módulo primero controla el léxico utilizado en el requisito, después analiza su coherencia y finalmente valida la forma.

3.1 Análisis Léxico

Este análisis lleva a cabo el control del léxico utilizado en la especificación. Dado un requisito, el análisis léxico verifica que las palabras utilizadas pertenezcan al léxico del dominio de aplicación. Para ello, SAREL utiliza tres léxicos diferentes:

- El léxico extendido incluye todas las palabras relacionadas con el dominio de aplicación.
- El léxico permitido es un extracto del anterior. Cada palabra del léxico permitido representa un conjunto de sinónimos del extendido.
- El léxico general contiene las palabras que se utilizan generalmente en inglés.

En primer lugar el analizador asegura que todas las palabras que forman parte del requisito pertenezcan a uno de los tres léxicos anteriores. Después de ello todas las palabras que pertenecen al léxico extendido serán sustituídas por los sinónimos correspondientes al léxico permitido. Esta división del léxico ofrece al ingeniero cierta flexibilidad en la redacción, dado que puede utilizar un léxico extendido en lugar del léxico permitido que es más restrictivo.

Actualmente estamos adaptando una base de datos léxica on-line (WordNet) [14] al dominio aeroespacial para la construcción de los léxicos anteriores. En cualquier caso el proceso de validación léxica es independiente del dominio de aplicación.

Fig. 1. Módulos de SAREL

3.2 Análisis Sintáctico-Semántico

En esta fase, se utiliza el analizador Alvey [7]. Este analizador está basado en la lógica de Montague y obtiene como resultado una representación semántica en forma de árbol. Este entorno ha sido modificado y adaptado al dominio aeroespacial y los resultados obtenidos son satisfactorios.

Aunque las reglas semánticas están asociadas con las sintácticas el GDE (Entorno de Desarrollo de Gramáticas) no es capaz de capturar el significado completo de un requisito y es por ello que a continuación debe ser realizado un análisis conceptual.

3.3 Control de la Ambigüedad

Es posible obtener más de una representación semántica a partir de un requisito. Esta situación aparece cuando el requisito contiene algún tipo de ambigüedad. El objetivo entonces es identificar la representación que se corresponde con la idea del ingeniero. Por ejemplo el requisito "*The AEROS will monitor the computer on-board and the status of the space-vehicle*" puede dar lugar a dos interpretaciones posibles:

a) "*The AEROS will monitor the computer on-board the space-vehicle and the AEROS will monitor the status of the space vehicle*"

b) "*The AEROS will monitor the computer on-board and the AEROS will monitor the status of the space vehicle*"

El controlador de la ambigüedad aplica una serie de reglas al conjunto de representaciones semánticas para calificarlas. Un ejemplo de regla podría ser aquella que afirma que la preposición *of* está siempre relacionada con el último grupo nominal. Este controlador coopera con el ingeniero a la hora de seleccionar la representación semántica correcta.

3.4 Control de la Simplicidad

Una de las propiedades expresadas en las normas de redacción es la simplicidad. A partir de la representación semántica del requisito este controlador detecta si la estructura de la frase es simple o compuesta. A modo de ejemplo presentamos el siguiente requisito que no cumple dicha propiedad.

Req.: "*The AEROS will control the computer on-board the space-vehicle and the AEROS will control the automatic systems of the flight configuration*"

Si el requisito está compuesto por dos requisitos simples, el proceso de validación puede seguir adelante procesando los dos requisitos simples de manera secuencial.

4 Módulo de Refinamiento Conceptual

Este módulo valida el requisito en relación con la Base de Requisitos (RB). En primer lugar obtiene un representación conceptual utilizando la Base de Conocimiento (KB). Tanto la RB como la KB utilizan un formalismo basado en frames [18]. A partir de la representación conceptual este módulo detecta posible información duplicada. Actualmente ya se ha definido la Base de Conocimiento, y la Base de Requisitos se incrementa manualmente cuando un requisito ya ha sido analizado. El proceso de integración automática de requisitos está todavía en fase de desarrollo.

4.1 Análisis Conceptual

Este análisis identifica en la Base de Conocimientos aquellas entidades involucradas en la representación semántica y construye la representación conceptual del requisito. La figura 2, de [18], muestra la representación conceptual de los dos requisitos siguientes:

Req-1: "*The AEROS will monitor the systems of the space-vehicle*"

Req-2: "*The AEROS will control the automatic systems of the space-vehicle*"

4.2 Control de Duplicidad

La función del controlador de duplicidad es verificar que el requisito introducido por el ingeniero contiene nueva información. Para ello, compara la representación conceptual del requisito con la Base de Requisitos para descubrir posibles duplicidades. Los ejemplos siguientes muestran un conjunto de requisitos equivalentes, para los cuales el sistema obtiene la misma representación conceptual.

Req-1:"*The AEROS will monitor the computer on-board the space-vehicle*"

Req-2:"*The computer on-board the space-vehicle will be monitored by the AEROS*"

Req-3:"*In the space-vehicle, the computer on-board will be monitored by the AEROS*"

Req-4:"*The computer on-board the space-vehicle will be monitored*"

El controlador de duplicidad comenzará buscando en la Base de Requisitos todos aquéllos que contengan la actividad que aparece en el nuevo requisito. Todos ellos serán examinados para descubrir si tanto el agente como el objeto de la actividad coinciden con los del nuevo requisito. Cuando el controlador detecta una duplicidad, ofrecerá al ingeniero la posiblidad de refinar el nuevo requisito.

Fig. 2. Representación Conceptual

5 Módulo de Control de Calidad de Software

Este módulo lleva a cabo una serie de análisis opcionales que validan de manera global la Base de Requisitos incrementada con el nuevo requisito. El objetivo es ofrecer información acerca de los factores de calidad de software que han sido considerados más relevantes en LESD (*completitud, trazabilidad, consistencia, verificabilidad y modificabilidad*).

Una vez el requisito ha sido validado y la información acerca de la calidad de software ha sido presentada al ingeniero, él decide si la representación conceptual del requisito debe ser añadida o no a la Base de Requisitos. Este proceso construye de manera incremental la representación conceptual de la especificación escrita en lenguaje natural.

Teniendo en cuenta que estos análisis son opcionales, el ingeniero podría decidir no utilizarlos (es una decisión razonable cuando se están añadiendo los primeros requisitos). En cualquier caso estos módulos pueden ser utilizados en el futuro para controlar globalmente la calidad de la especificación de software en ese momento. Los análisis relacionados con los factores de calidad ya estudiados son descritos en los siguientes párrafos.

5.1 Análisis de la Trazabilidad

El objetivo aquí es proveer al ingeniero de información acerca de los enlaces de trazabilidad del requisito. Para ello, SAREL activa un conjunto de algoritmos existentes [18] que controlan la trazabilidad en el sentido que muestran las relaciones entre el requisito introducido y un subconjunto de requisitos de la RB (o bien más específicos o bien más generales). A partir de esta información el ingeniero puede observar las relaciones existentes entre los requisitos introducidos hasta ese punto. A partir del siguiente requisito:

Req-1: "*The AEROS will receive the data of the space vehicle*"

el sistema muestra, entre otros, el siguiente requisito más general:

Req-2: "*The system will monitor the data of the space vehicle*"

El segundo requisito es más general que el primero porque "AEROS" es una instancia de "system" y "receive" es parte de "monitor".

La falta de relaciones refleja que el requisito está aislado respecto al resto de requisitos.

5.2 Análisis de Completitud

En esta etapa SAREL activa una serie de mecanismos de razonamiento que están siendo desarrollados ([1] [2]) teniendo en cuenta varios aspectos de la completitud. Para analizar la completitud, es necesario tener disponible una jerarquía general de accciones-subacciones que puedan ser asociadas a cualquier conjunto de especificaciones relacionadas con un sistema aeroespacial. Un ejemplo de esta jerarquía está presente en la actividad *monitor*, que comprende tres subactividades *receive*, *analyse* y *display*. Dado el siguiente requisito:

Req.: "*During the launch phase, the AEROS will analyse and display the status of the space vehicle*"

teniedo en cuenta la jerarqía descrita anteriormente, el sistema analiza las relaciones entre requisitos en la Base de Requisitos. Si no existe ningún requisito que contenga la actividad *monitor*, informará al ingeniero. De la misma manera, si el sistema observa que hay un requisito que contiene la actividad *monitor* pero no existen requisitos que contengan las subactividades de la jerarquía, informará al ingeniero que la especificación actual podría ser incompleta.

5.3 Análisis de Modificabilidad

La complejidad y consistencia de futuras modificaciones en la especificación de software depende del nivel de propagación de la modificación sobre el resto de requisitos. El concepto de modificabilidad en LESD ha sido formalizado [8] [9] [1] de acuerdo con los niveles de interconexión entre los requisitos de la especificación. El rango de valores posibles es [0..1], pero el rango de valores aceptables es un subintérvalo de éste. En caso de obtener una medida que no pertenezca a este segundo rango, el ingeniero debería analizar un posible problema de interconexión excesiva o insuficiente.

6 Conclusiones y Trabajo Futuro

Nuestro trabajo de investigación está centrado en el control de la calidad en especificaciones de software escritas en lenguaje natural. Dado que la fase de especificación es realizada por un analista humano, hemos diseñado un sistema para redactar especificaciones de software en lenguaje natural. La calidad de la especificación es estudiada desde dos puntos de vista: las normas de redacción y los factores de calidad de software.

El proceso de ayuda está dividido en tres módulos: el Módulo de Refinamiento de Estilo, el Módulo de Refinamiento Conceptual y el Módulo de Control de Calidad de Software. Una vez un requisito ha sido validado por estos módulos, es añadido a la Base de Requisitos. Durante este proceso obtenemos de manera incremental una representación conceptual de la especificación.

Para desarrollar exhaustivamente nuestro sistema, el trabajo futuro estará focalizado en a) diseñar mecanismos automáticos que permitan la obtención de una Representación Conceptual a partir de una Representación Semántica, b) desarrollar mecanismos semiautomáticos que construyan la Base de Conocimiento del dominio a partir de documentos, y c) definir el proceso de validación de la Base de Conocimiento obtenida. Además, se continuará con el estudio de los factores de calidad seleccionados.

Bibliografía

1. Alvarez J., Castell N., Slavkova O. "Combining Knowledge and Metrics to Control Software Quality Factors" In "Achieving Quality in Software" pp 201-212 Chapman & Hall (1996)
2. Alvarez J., Castell N. "An Approach to the Control of Completeness Based on MetaKnowledge", Technical report, LSI-94-50-R Dept. of LSI, Universitat Politècnica de Catalunya, 1994.
3. ANSI/IEEE Std 729-1983. *IEEE Guide to Software Requirements Specifications* 1983.
4. Association Europèene des Constructeurs de Matériel Aéronautique. *AECMA Simplified English, A Guide for the preparation of aircraft maintenance documentation in the international aerospace maintenance language*, December 1989.

5. Borillo M., Borillo A., Castell N., Latour D., Toussaint Y., Verdejo M.F. "Applying Linguistic Engineering to Software Engineering: The traceability problem". In *Proceedings of the European Conference on Artificial Intelligence (ECAI92)*, pages 593-595, Viena, Austria, August 1992.
6. Borillo M., Toussaint Y., and Borillo A. "A. Motivations du project LESD". In *Conference on Linguistic Engineering'91*, Versailles, France, January 1991.
7. Briscoe T., Grover C., Boguraev B., Carroll J. "The ALVEY Natural Language Tools Project Grammar: A Large Computational Grammar". Technical report, ALVEY Documents, Cambridge Univ., Computer Laboratory, UK, 1987.
8. Castell N., Slavkova O. "The Modifiability Factor in the LESD Project: Definition and Practical Results", Technical report, LSI-95-7-R Dept. of LSI, Universitat Politècnica de Catalunya, 1993.
9. Castell N., Slavkova O. "Metrics for Quality Factors in the LESD Project". In 5th European Software Engineering Conference (ESEC'95), Sitges, Spain, 1995. Lecture Notes in Computer Science n. 989, Springer Verlag pp 423-437.
10. Castell N., Slavkova O., Toussaint Y. and Tuells A. "Quality Control of Software Specifications written in Natural Language". In *Proceedings of the Seventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE'94)*, Austin, Texas, USA, 1994.
11. Castell N., Hernández A. "Filtering Software Specifications Written in Natural Language". In 7th Portuguese Conference on Artificial Intelligence, (EPIA'95) Madeira, Portugal, 1995. Lecture Notes on AI num. 990.
12. Hernández, A. "SAREL: An assistance system for writing software specifications in natural language". In *Proceedings of the IBERAMIA'94*, ISBN 980-6168-16-X Caracas, Venezuela, 1994.
13. Jarke M., Bubenko J., Rolland C., Sutcliffe A. and Vassiliou J. "Theory Underlying Requirement Engineering: An Overview of NATURE at Genesis". In *Proceedings of the IEEE International Symposium on Requirements Engineering (RE'93)*, San Diego, California, USA, 1993.
14. Miller, G.A. "Wordnet: A Dictionary Browser" in *Information in Data, Proceedings of the First Conference of the UW Centre for the New Oxford Dictionary*, Waterloo, Canada: University of Waterloo. 1985.
15. Reubenstein H.B. and Waters R.C. "The Requirements Apprentice: Automated Assistance for Requirements Acquisition". *IEEE Transactions on Software Engineering*, 17:226-240, 1991.
16. Rich C. and Waters R.C. *The Programmer's Apprentice.* Reading, MA: Addison-Wesley, and Baltimore, MD. ACM Press, 1990.
17. Rolland C., Proix C. *A Natural Language Approach for Requirements Engineering*". Conceptual Modeling, Databases and CASE: An Integrated View of Information Systems Development, P. Loucopoulos, R. Zicari (eds.) WILEY, 1992.
18. Toussaint Y. *Méthodes Informatiques et Linguistiques pour l'Aide a la Spécification de Logiciel.* PhD thesis, Universidad Paul Sabatier, Toulouse, 1992.
19. Tsai J.P., Weigert T. and Jang H.C. "A Hybrid Knowledge Representation as a Basis of Requirement Specification and Specification Analysis". *IEEE Transactions on Software Engineering*, 18:1076-1100, 1992.