

# Control de Calidad de Especificaciones de Software Escritas en Lenguaje Natural

Jordi Àlvarez, Núria Castell, Neus Català y Àngels Hernández  
grupo ER&IA

Universitat Politècnica de Catalunya  
Departament de Llenguatges i Sistemes Informàtics  
Campus Nord, Mòdul C5, Barcelona 08034  
castell@lsi.upc.es

## Resumen

El problema básico que abordamos en el grupo ER&IA (*Enginyeria de Requisits i Intel·ligència Artificial*) es el control de la calidad de las especificaciones preliminares de software escritas en lenguaje natural. Las propiedades que tenemos en cuenta para este control son la trazabilidad, la consistencia, la completitud, la verificabilidad y la modificabilidad. Los objetivos concretos en nuestro grupo de trabajo son la definición computacional de las propiedades en estudio, el desarrollo del entorno informático necesario para su tratamiento y el diseño e implementación de la interacción sistema/ingeniero para la realización de las tareas de análisis de las especificaciones.

## 1 Antecedentes

El trabajo de investigación en el que estamos involucrados trata del control de las especificaciones de software escritas en lenguaje natural. La mayoría de errores detectados en las fases de codificación y mantenimiento provienen de una mala especificación [6, 14]. Además, el coste de corregir un error de especificación se incrementa desmesuradamente cuando se detecta en las fases de codificación o mantenimiento [20].

Éstos fueron los principales motivos por los que surgió el proyecto LESD (Linguistic Engineering for Software Development) [8]. Este proyecto nació en el centro ARAMIIHS de Toulouse (Francia), centro creado mediante un acuerdo entre el CNRS i la empresa MATRA MARCONI SPACE. En el proyecto participaban investigadores del IRIT (Institut de Recherche en Informatique de Toulouse, de la Universidad de Ciencias Paul Sabatier, de la universidad de letras Le Mirail, de la empresa MATRA i, más recientemente a través de una acción integrada hispano-francesa, nuestro grupo de la Universidad Politècnica de Catalunya (Departamento de Lenguajes y Sistemas Informáticos - Sección de Inteligencia Artificial).

La finalidad de LESD era el desarrollo de herramientas informáticas que permitan crear una interpretación conceptual de especificaciones de requisitos en inglés, evaluar ciertos

criterios de calidad (algoritmos de razonamiento sobre la representación conceptual) y ayudar en la explotación de estos documentos.

A más largo término, el objetivo de LESD era el desarrollo de un sistema de ayuda a la redacción que permita detectar los problemas ligados al uso de una terminología no identificada y de formas sintácticas complejas o ambiguas. El dominio en el que trabajaba LESD era el software aeroespacial.

LESD explora en profundidad el lenguaje, por lo que integra componentes léxicos, sintácticos y semánticos. El conocimiento del dominio ha sido introducido previamente, pues es absolutamente imprescindible para la construcción de la interpretación conceptual y para el razonamiento. Para tener una visión más amplia de LESD podemos remitirnos a [13, 7, 23].

## 2 Proyecto TIC93-0420

El trabajo de LESD fue continuado por nuestro grupo en el proyecto español CICYT TIC93-0420. El problema básico que abordamos era el control de la calidad de las especificaciones. Las propiedades que tenemos en cuenta para este control son la trazabilidad, la consistencia, la completitud, la verificabilidad y la modificabilidad.

En el marco de LESD sólo se había estudiado la trazabilidad de las especificaciones [7, 23]. Los objetivos concretos de nuestro proyecto TIC son la definición computacional de las 4 propiedades restantes (sección 2.2), el desarrollo del entorno informático necesario para su tratamiento y el diseño e implementación de la interacción sistema/ingeniero para la realización de las tareas de análisis de las especificaciones (sección 2.3).

### 2.1 Participantes

En la actualidad, el grupo ER&IA está constituido por 4 miembros:

- Núria Castell es Profesora Titular de Universidad y trabaja en el área de procesamiento del lenguaje natural (especificaciones de software, extracción de información a partir de documentos).
- Jordi Alvarez es Becario del programa FI de la Generalitat de Catalunya y está trabajando sobre mecanismos de razonamiento que, a partir de la conceptualización del conocimiento contenido en especificaciones de requisitos, nos permitan controlar la calidad de éstas.
- Neus Català es Profesora Asociada y trabaja en la construcción automática de diccionarios a partir de textos técnicos.
- Angels Hernández es Profesora Titular de Escuela Universitaria y está trabajando en el diseño y desarrollo del sistema SAREL (Sistema de Ayuda a la Redacción de Especificaciones en Lenguaje Natural)
- También formaron parte del grupo Toni Tuells, que inició el estudio de la completitud; y Olga Slavkova, principalmente interesada en métricas y que llevó a cabo el estudio sobre la modificabilidad [11, 5].

## 2.2 Estudio de los Factores de Calidad

En cuanto a los factores de calidad, se han estudiado la completitud y la modificabilidad [5]. La primera se ha estudiado desde un punto de vista simbólico [3, 2] mientras que el estudio de la modificabilidad [11] se ha hecho a partir del modelo presentado en [22], basado en métricas.

### 2.2.1 Modificabilidad

En cuanto a la modificabilidad, el estudio ataca dos objetivos principales:

- Analizar el nivel de complejidad de una modificación respecto al conjunto de requisitos como un todo y respecto a cada requisito individual.
- Seleccionar la lista de requisitos que pueden estar afectados por una modificación determinada.

La automatización del análisis de la modificabilidad nos permite por un lado controlar el nivel de interacción entre los requisitos (si es demasiado alto, puede ser costoso hacer una modificación; mientras que si es demasiado bajo puede ser indicativo de que la especificación es incompleta); y por otro lado nos permite, dada una modificación, acceder a la lista de requisitos que deberemos revisar para llevarla a cabo.

La base del concepto de modificabilidad ha sido definido en función del nivel de interconexión entre los requisitos de la especificación. La interconexión entre diferentes requisitos viene definida por el uso común de entidades del dominio en esos requisitos.

Para calcular esta interconexión se hace uso de lo que llamamos *tabla de interconexión* y se definen dos métricas: *nivel de interconexión* y *nivel de interconexión individual* que se basan en el modelo de diferencia de entropías de Emden [25], adaptado para calcular la complejidad de un programa por Robillard [21].

### 2.2.2 Completitud

La completitud es uno de los factores más importantes y a la vez difíciles de evaluar en una especificación. En el estudio realizado se ha puesto de manifiesto la necesidad de tener un amplio conocimiento del dominio de aplicación para poder evaluar la completitud [3].

El trabajo sobre este factor de calidad empezó con un estudio más bien teórico [24] sobre las distintas vertientes del concepto y su posible aplicación al campo de la ingeniería de requisitos.

De esto pasamos a un análisis de la completitud mediante el uso de metaconocimiento, bien general o bien sobre el dominio en el cual se mueven nuestras especificaciones de requisitos [3, 2, 5]. Este metaconocimiento nos permite, a partir del conocimiento o información presente en la especificación, deducir otra información que necesariamente debería también estar presente en la especificación<sup>1</sup>.

El problema principal que presenta esta aproximación es el de adquirir el metaconocimiento que nos permita realizar todo esto. El hecho de que se trate de especificaciones

---

<sup>1</sup>Evidentemente este mecanismo no nos va a proporcionar la información exacta que debería estar presente en la especificación sino una abstracción o generalización de ésta (de otro modo, podríamos deducir una especificación completa a partir de una incompleta).

preliminares y que estén escritas en lenguaje natural hace que esta tarea sea realmente complicada.

Esto nos llevó a una nueva aproximación, que se puede ver como una evolución de la anterior, en que intentamos adquirir un modelo de especificación de forma automática [4]. Para ello aplicamos técnicas de aprendizaje automático, lógica terminológica y redes semánticas.

La idea principal que hay detrás de la adquisición automática de un modelo, es que una especificación consiste en una serie de descripciones de entidades (aquellas que intervienen en el sistema especificado). Estas descripciones de entidades se pueden agrupar en tipos que se pueden estructurar de forma jerárquica (igual que las entidades a las que describen [19]) y, además, imponen unas ciertas restricciones sobre sus instancias.

Lo que el sistema de aprendizaje pretende realizar es aprender estos tipos (y también sus restricciones) a partir de un conjunto de especificaciones de requisitos que son correctas. Posteriormente, una vez ya tenemos nuestra base de conocimiento con la información sobre los tipos, podemos utilizarla para validar nuevas especificaciones. La validación se hace efectiva combinando lógica terminológica [18] (que nos permite identificar fácilmente a una descripción con un tipo), y la representación probabilística introducida en COBWEB [15], que nos permite determinar qué características o información debe tener una descripción asociada a un tipo.

### **2.2.3 Otros factores de calidad**

La creación de una jerarquía de tipos de descripciones como la que creamos nos permite también tratar, además de la completitud, otro factor de calidad: la consistencia. De la misma forma que podemos evaluar qué información es necesaria, también podemos evaluar qué información es inconsistente en una descripción.

En un futuro próximo investigaremos si otros factores de calidad se pueden abordar también desde esta perspectiva; pero todo parece indicar que así es.

## **2.3 SAREL: Un Sistema de Ayuda a la Redacción de Especificaciones de Software en Lenguaje Natural**

En paralelo al proceso de definición computacional de los factores de calidad, estamos desarrollando un sistema que ayude al ingeniero de software en la fase de creación de especificaciones preliminares. [16, 10].

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 [17, 1]) y los factores de calidad [13]. Este proceso construye de manera incremental una representación conceptual de la especificación. Los controles están 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. Sobre esta Base el ingeniero podrá activar los mecanismos de comprobación de los factores de calidad.

### **2.3.1 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 am-

bigüedad y control de simplicidad.

**Análisis Léxico** Dado un requisito, el análisis léxico verifica que las palabras utilizadas pertenezcan al léxico del dominio de aplicación. Para ello se utilizan tres léxicos diferentes: el extendido (incluye todas las palabras relacionadas con el dominio de aplicación), el permitido (es un extracto del anterior) y el general (contiene las palabras que se utilizan generalmente en inglés).

**Análisis Sintáctico-Semántico** En esta fase, se utiliza el analizador Alvey [9] y se obtiene como resultado una representación semántica del requisito en forma de árbol.

**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 entre las posibles representaciones semánticas aquella que se corresponde con la idea del ingeniero.

**Control de la Simplicidad** A partir de la representación semántica del requisito éste controlador detecta si la estructura de la frase es simple o compuesta. Si el requisito está compuesto por varios requisitos simples, el proceso de validación puede seguir adelante procesando todos los requisitos simples de manera secuencial.

### 2.3.2 Módulo de Refinamiento Conceptual

Este módulo valida el requisito en relación con la Base de Requisitos (RB). En primer lugar obtiene una representación conceptual utilizando la Base de Conocimientos (KB) y a partir de esta representación detecta posible información duplicada.

**Análisis Conceptual** Este análisis identifica en la Base de Conocimientos aquellas entidades involucradas en la representación semántica y construye la representación conceptual del requisito.

**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.

### 2.3.3 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 (*completitud, trazabilidad, consistencia, verificabilidad y modificabilidad*).

Teniendo en cuenta que estos análisis son opcionales, el ingeniero podría decidir no utilizarlos. En cualquier caso estos módulos pueden ser utilizados en el futuro para controlar globalmente la calidad de la especificación de software en ese momento.

**Análisis de la Trazabilidad** SAREL activa un conjunto de algoritmos existentes [23] que controlan la trazabilidad en el sentido que muestran las relaciones entre el requisito introducido y un subconjunto de requerimientos de la RB (o bien más específicos o bien más generales).

**Análisis de Completitud** En esta etapa SAREL activa una serie de mecanismos de razonamiento que están siendo desarrollados ([5, 3]) teniendo en cuenta varios aspectos de la completitud. Para analizar la completitud, es necesario tener disponible una jerarquía general de acciones-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*.

**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 [12, 11, 5] 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 subintervalo 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.

### 3 Trabajo en Curso y Líneas Futuras

Por lo que se refiere a los mecanismos de razonamiento para evaluar los factores de calidad, en un futuro inmediato continuaremos trabajando con nuestro modelo de representación de conocimiento y aprendizaje, haciendo más expresivo al primero y profundizando en el segundo, pues se encuentra en un estado preliminar. Una vez hecho esto, abordaremos el resto de factores de calidad desde una perspectiva común a partir de estas herramientas.

En cuanto al sistema SAREL, el trabajo en curso está centrado 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 Conocimientos del dominio a partir de documentos, y c) definir el proceso de validación de la Base de Conocimientos obtenida. Además, se continuará con el estudio de los factores de calidad seleccionados.

## Referencias

- [1] AECMA. *AECMA Simplified English, A Guide for the Preparation of Aircraft Maintenance Documentation in the International Aerospace Maintenance Language*, 1989.
- [2] J. Alvarez. An approach to the control of completeness in LESD. In *Proceedings of Doctoral Consortium of 2nd International Symposium on Requirements Engineering*, Mar. 1995.
- [3] J. Alvarez and N. Castell. An approach to the control of completeness based on metaknowledge. Technical Report LSI-94-50-R, Universitat Politècnica de Catalunya, 1994.

- [4] J. Alvarez and N. Castell. Uso de técnicas de aprendizaje para la validación de especificaciones. *Novática*, (118):45–52, Noviembre–Diciembre 1995.
- [5] J. Alvarez, N. Castell, and O. Slavkova. Combining knowledge and metrics to control software quality factors. In *IFIP International Conference on Achieving Quality in Software (AQUIS'96)*, Florence, Jan. 1996.
- [6] V. Basili and B. Perricone. Software errors and complexity: An empirical investigation. *Communications of the ACM*, 27(1):42–52, Jan. 1984.
- [7] M. Borillo, N. Castell, D. Latour, Y. Toussaint, and F. Verdejo. Applying linguistic engineering to software engineering: The traceability problem. In *Proceedings of The 10th European Conference on Artificial Intelligence (ECAI'92)*, pages 593–595, Vienna (Austria), Aug. 1992. John Wiley & Sons.
- [8] M. Borillo, Y. Toussaint, and A. Borillo. Motivations du projet LESD. In *Linguistic Engineering Conference*, Versailles, France, Jan. 1991.
- [9] T. Briscoe, C. Grover, B. Boguraev, and J. Carroll. The ALVEY natural language tools project grammar: A large computational grammar. Technical report, Cambridge University, Computer Laboratory, 1987. ALVEY Documents.
- [10] N. Castell and A. Hernández. Filtering software specifications written in natural language. In *Proceedings of 7th Portuguese Conference on Artificial Intelligence, (EPIA '95)*, pages 447–455, Madeira, Portugal, Oct. 1995. LNAI-990, Springer Verlag.
- [11] N. Castell and O. Slavkova. Metrics for quality factors in the LESD project. In *Proceedings of 5th European Software Engineering Conference*, pages 423–437, Sitges (Spain), Sept. 1995. LNCS-989, Springer Verlag.
- [12] N. Castell and O. Slavkova. The modifiability factor in the LESD project: Definition and practical results. Technical Report LSI-95-7-R, Universitat Politècnica de Catalunya, 1995.
- [13] N. Castell, O. Slavkova, Y. Toussaint, and A. Tuells. Quality control of software specifications written in natural language. In *Proceedings of 7th International Conference on Industrial & Engineering Applications of AI & Expert Systems (IEA-AIE'94)*, pages 37–44, Austin, Texas, June 1994. Gordon and Breach Science Publishers.
- [14] A. Endres. An analysis of errors and their causes in system programs. *IEEE Transactions on Software Engineering*, 1(2):140–149, June 1975.
- [15] D. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.
- [16] A. Hernández. Sarel: An assistance system for writing software specifications in natural language. In *Proceedings of the IBERAMIA'94*, Caracas, Venezuela, 1994. ISBN 980-6168-16-X.
- [17] IEEE. *IEEE Guide to Software Requirements Specification*, IEEE-830, 1984.

- [18] R. MacGregor. Using a description classifier to enhance deductive inference. In *Proceedings of the 7th IEEE Conference on Artificial Intelligence*. IEEE CS Press, 1991.
- [19] N. Maiden, P. Mistry, and A. Sutcliffe. How people categorise requirements for reuse: a natural approach. In *Proceedings of 2nd International Symposium on Requirements Engineering*, pages 194–203, York, England, Mar. 1995. IEEE CS Press.
- [20] R. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, New York, NY, USA, third edition, 1992.
- [21] P. Robillard and G. Boloix. The interconnectivity metrics: A new metric showing how a program is organized. *The Journal of Systems and Software*, 10:29–39, 1989.
- [22] O. Slavkova. Modelo para el control de calidad en LESD basado en medición del software. Technical Report LSI-93-26-R, Universitat Politècnica de Catalunya, 1993.
- [23] Y. Toussaint. *Méthodes Informatiques et Linguistiques pour l'aide a la Spécification de Logiciel*. PhD thesis, Paul Sabatier University, Toulouse, 1992.
- [24] T. Tuells and N. Castell. The completeness problem in LESD. Technical report, Universitat Politècnica de Catalunya, 1993.
- [25] M. van Emden. Hierarchical decomposition of complexity. *Machine Intelligence*, 5:361–380, 1970.