

Analizador de Completitud de Requisitos Escritos en Español Restringido

Javier Navarro, Ricardo Orozco y Aldriu Fredy Jaramillo

UNIVERSIDAD DE ANTIOQUIA. Departamento de Ingeniería de Sistemas
zeraus80@hotmail.com; nulox@hotmail.com; aljara@udea.edu.co

Carlos Mario Zapata
UNIVERSIDAD NACIONAL DE COLOMBIA.
Facultad de Minas. Escuela de Sistemas
cmzapata@unalmed.edu.co

Recibido para revisión Marzo 2005, aceptado Mayo 2005, versión final recibida Mayo 2005

Resumen: En este artículo se propone una herramienta para el análisis de completitud de requisitos escritos en Español restringido. La herramienta implementa un método para mejorar la completitud de los requisitos utilizando un enfoque lingüístico, basado en los casos semánticos asociados al sentido de un verbo. Se presenta también el proceso seguido en la evaluación de la efectividad de la herramienta al analizar la completitud de varios casos de estudio utilizando la precisión como indicador cuantitativo de la efectividad. Los resultados preliminares obtenidos confirman las hipótesis y muestran que la herramienta servirá de ayuda en la escritura de especificaciones de requisitos completas y el mejoramiento de la calidad del software que se desarrolla a partir de ellas.

Palabras Clave: Calidad de requisitos, especificación de requisitos, completitud de requisitos.

Abstract: In this paper, we propose a tool for Requirements completeness analysis over requirements written in restricted Spanish. This tool materialize a method for requirements completeness based on a linguistic approach, more specifically based on semantic cases associated with verb senses. We also present the process for effectiveness of the tool, in completeness on several case studies, using precision as a quantitative indicator of effectiveness. We obtain some preliminary results confirming the hypothesis, and we show that this tool will be helpful for writing complete requirements specifications and for improving software quality based on them.

Keywords: Requirements quality, requirements specification, requirements completeness.

1 INTRODUCCIÓN

En la Ingeniería del Software, el analista de sistemas realiza un proceso de ingeniería de requisitos en el cual, en la etapa de especificación, captura y escribe en lenguaje natural los requisitos de un sistema. Una vez capturados los requisitos, el analista abstrae, a partir de lo escrito, los elementos estructurales del sistema con sus relaciones, y elabora modelos conceptuales que describen la arquitectura y comportamiento del mismo. Para asegurar la calidad del software, los analistas de sistemas necesitan especificar requisitos con calidad.

Entre los principales atributos de calidad de una especificación de requisitos se encuentran la consistencia, completitud, corrección e inambigüedad. De manera particular, la completitud se refiere a la no existencia de información sin declarar o definir, como objetos o entidades, en una especificación de requisitos [Boehm

(1984)]. La escritura de requisitos que no cumplen con la propiedad de completitud tiene como consecuencia que se omitan, al momento de la abstracción y diseño, elementos estructurales fundamentales en el funcionamiento de un sistema, afectando la calidad del software que se desarrolla a partir de los mismos, hecho que se ve reflejado en el nivel de concordancia del software desarrollado con las necesidades del cliente.

Con el ánimo de evaluar la completitud de especificaciones de requisitos escritas en español restringido, se construyó un analizador de completitud de requisitos. La herramienta se compone de un analizador léxico-morfológico, para etiquetar las palabras de una frase según su categoría sintáctica, un analizador sintáctico basado en una gramática de libre contexto que representa un subconjunto del Español, y un método de apareamiento de los casos semánticos de verbos y sus

tantivos para determinar la completitud de las especificaciones de requisitos.

Este artículo está estructurado de la siguiente manera: en la Sección 2 se describen estudios relacionados con el dominio del problema, en la Sección 3 se presenta el método de mejoramiento de completitud que se utilizó y en la Sección 4 la herramienta que se implementó aplicada a un caso de estudio. En la Sección 5 se presenta el proceso de evaluación de la efectividad de la herramienta con los resultados sobre los casos de estudio. En la Sección 6 se discuten los resultados de la evaluación y en la Sección 7 se presentan las conclusiones y trabajos futuros.

2 TRABAJOS RELACIONADOS

En la calidad de requisitos se encuentran varios estudios relacionados con la completitud como una propiedad de la calidad; algunos de estos trabajos tienen propuestas de patrones lingüísticos y otros poseen herramientas que de forma automática analizan la calidad de las especificaciones de requisitos en su contenido.

El primer estudio consta de un modelo de patrones lingüísticos que se define a partir del lenguaje que es utilizado implícitamente por los ingenieros de requisitos en la abstracción de los elementos estructurales de un sistema perteneciente al dominio de las bases de datos. Esos patrones lingüísticos se basan en una clasificación de verbos y en los tipos de relaciones o casos semánticos que hay entre un grupo de palabras y un verbo. Un ejemplo de la aplicación del modelo puede ser encontrado en Rolland (1992).

En el segundo estudio Olmishi (1991) implementa un lenguaje de requisitos para el japonés que se basa en un modelo compuesto de tres tipos de marcos que restringen el vocabulario y el contexto de las especificaciones de requisitos. De esos tipos de marcos, el de casos ayuda al ingeniero de requisitos a detectar casos semánticos omitidos de un concepto y el uso ilegal de datos. El modelo de marcos se restringe al dominio de sistemas de archivo, lo que representa una desventaja ya que sólo asocia los verbos a conceptos del anterior dominio, característica que puede dificultarle al analista de requisitos la relación de un verbo con un concepto. Un ejemplo del uso del modelo de marcos se encuentra en Field (1999).

Los estudios de Wilson (2000), Fabbrini (2001) y Lami (2004), implementan una herramienta para evaluar las especificaciones de requisitos automáticamente utilizando indicadores de calidad que han sido definidos para algunos aspectos de calidad, como la completitud, consistencia, ambigüedad, entre otros, que son deseables en las especificaciones de requisitos. La herramienta realiza el análisis de las especificaciones de requisitos por medio del conteo de palabras, para establecer de acuerdo

con el indicador de calidad que esté evaluando, la debilidad o fortaleza de un requisito. La desventaja de la herramienta es que no ayuda a definir un requisito completo al momento de su escritura, y en la detección del atributo de calidad completitud sólo tiene en cuenta la estructura sintáctica de las especificaciones de requisitos, obviando su composición semántica.

Ben Achour (1999) propone una guía basada en patrones y estructuras lingüísticas que usan reglas sintácticas y semánticas, que guían al ingeniero de requisitos, y al usuario común, en la escritura de escenarios de especificaciones de requisitos de un sistema determinado. Las reglas semánticas se basan en los casos semánticos relacionados a un verbo, y su aplicación ayuda a mejorar la completitud y consistencia de los requisitos especificados en un escenario. El inconveniente de la guía es que el analista de requisitos debe conocerla en su totalidad ya que no se encuentra automatizada. De otro lado, en el estado del arte no se han encontrado, a la fecha, propuestas para el análisis de completitud de requisitos escritos en Español.

A continuación se presenta el método utilizado para la implementación del analizador de completitud.

3 MÉTODO PARA MEJORAR LA COMPLETITUD DE REQUISITOS UTILIZANDO UN ENFOQUE LINGÜÍSTICO

En esta Sección se presenta un método para determinar la completitud en una especificación de requisitos. Para la implementación del método se deben realizar inicialmente dos procesos, el análisis léxico-morfológico y el análisis sintáctico de los requisitos.

3.1 Análisis Léxico-Morfológico

Este proceso utiliza como recurso lingüístico un diccionario con la información de la raíz y categoría sintáctica de las palabras del Español. El proceso verifica para cada palabra de un requisito, su presencia en el diccionario y su categoría sintáctica (artículo, verbo, preposición, sustantivo, adjetivo, adverbio). La información de los casos semánticos relacionados a un verbo, y de los casos semánticos desempeñados por un sustantivo, están contenidos en dos lexicones, el de verbos y el de sustantivos, los cuales son recursos lingüísticos fundamentales para la implementación del analizador de completitud. El producto de este proceso son las categorías sintácticas de cada una de las palabras de un requisito. Este producto es requerido por el proceso que se describe a continuación.

3.2 Análisis Sintáctico

En este proceso se verifica que la sentencia cumpla con las reglas de la gramática que define un subconjunto del

Español (al cual se ha denominado Español restringido). En el análisis sintáctico se reciben las categorías sintácticas resultantes del análisis léxico-morfológico, para comprobar la validez de éstas con la gramática y producir un árbol sintáctico, el cual será utilizado en procesos posteriores. Luego de esto, se procede a realizar el análisis de completitud que se describe en los siguientes pasos.

3.3 Determinación del Sentido del Verbo

Realizados los anteriores procesos sobre un requisito, se procede a determinar el sentido de su verbo, el cual es único. Por ejemplo, el verbo adquirir puede tener, entre otros, el sentido de aprender (si lo que se adquiere es información) o comprar (si lo que se adquiere es un bien). En el lexicón de verbos de Dorr (2001) se encuentran los posibles sentidos que puede tomar un verbo y los casos semánticos asociados a cada uno de estos sentidos. Entre los principales casos semánticos contenidos en el lexicón de verbos tenemos: agente, tema, ubicación, objetivo, instrumento, entre otros. Para establecer el sentido del verbo, se le pide al usuario que elija de las opciones ofrecidas por el lexicón, aquella que corresponda al sentido de utilización del verbo en el requisito. Una vez determinado el sentido del verbo se procede a determinar los casos semánticos ausentes en el requisito. De aquí en adelante, con el fin de abreviar, haremos referencia a los casos semánticos como "casos".

3.4 Determinación de Casos Ausentes

Considerando que cada requisito puede estar constituido de casos obligatorios y opcionales, se ha dividido esta tarea en dos fases dando prioridad al apareamiento de casos obligatorios (ya que sin ellos el requisito carece de sentido). Una vez estén cubiertos los casos obligatorios se procede al apareamiento de los casos opcionales, los cuales se presentan puesto que brindan información adicional del sistema.

3.4.1 Determinación de casos obligatorios ausentes

Para cada frase nominal de un requisito se realizan los siguientes pasos:

- Se establece el nombre principal en la frase nominal.
- Con ayuda del lexicón se determinan los casos en los que puede participar el nombre.
- Se realiza la intersección o apareamiento de los casos del nombre y los casos obligatorios del verbo.
- Si la intersección da como resultado el conjunto vacío, el sistema registra qué información falta para informar al usuario al finalizar el análisis. De lo contrario, siempre y cuando el caso no haya sido

apareado con otra frase nominal, se almacena el caso que cumple la frase nominal en el requisito (agente, ubicación, etc.). Considerando que un caso normalmente ocurre una sola vez en cada requisito, esta información facilita el apareamiento de los casos correspondientes a las otras frases nominales de la misma.

- Si existen casos obligatorios ausentes: Se informa al usuario de la ausencia del caso para que pueda reescribir el requisito.

Una vez apareados todos los casos obligatorios, puede llevarse a cabo el análisis de casos opcionales ausentes cuyo proceso es análogo al anterior y se realiza teniendo en cuenta los casos apareados en esta etapa.

Se ha presentado el método en el cual se basa el analizador de completitud implementado. En la siguiente sección se presenta la interfaz gráfica de la herramienta y se muestra su funcionamiento en un requisito de un caso de estudio.

4 ANALIZADOR DE COMPLETITUD

4.1 Descripción del Analizador

Para implementar el analizador de completitud de requisitos escritos en Español restringido, se construyó una herramienta siguiendo el método de desarrollo en cascada y utilizando el lenguaje de programación orientado a objetos Java en la etapa de desarrollo. Para la implementación de las reglas de la gramática que restringen el Español se utilizó la herramienta Java Compiler. El analizador de completitud (ver Figura 1) utiliza varios recursos lingüísticos en su funcionamiento: un diccionario con la información sintáctica de las palabras, un lexicón de verbos con la información de los casos relacionados al sentido de un verbo y un lexicón de sustantivos con los casos que puede desempeñar cada sustantivo.

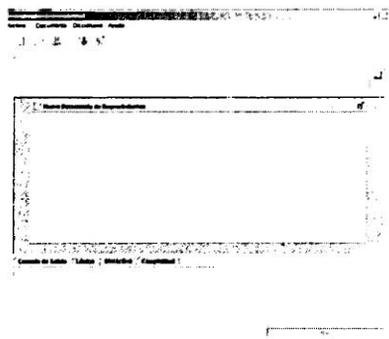


Figura 1: Interfaz gráfica del analizador de completitud

4.2 Caso de Estudio

En esta sección se muestra cómo se hace el análisis de completitud de una oración perteneciente a un caso de estudio tomado de Dong, Kalaivani, Far y Eberlein (2003). La oración que se va a analizar es "El cliente tiene una cuenta". Esta oración es sometida a los procesos especificados en la sección anterior con los siguientes resultados:

En el análisis léxico-morfológico la oración se etiqueta de acuerdo con la categoría sintáctica de las palabras que la componen (ver Tabla 1). Con el ánimo de hacer énfasis en el análisis de casos se omiten los detalles relacionados con la gramática que restringe el español y el análisis sintáctico.

Tabla 1: Etiquetamiento del requisito El cliente tiene una cuenta

PALABRA	CATEGORIA SINTÁCTICA	ETIQUETA
El	Determinante	Det
cliente	Sustantivo	Sust
tiene	Verbo	Verb
una	Determinante	Det
cuenta	Sustantivo	Sust

En el análisis sintáctico, con las etiquetas resultantes del análisis léxico morfológico se verifica que la estructura sintáctica de la sentencia que describe la oración cumpla con las reglas de la gramática. El resultado de este proceso es el árbol sintáctico de la Figura 2.

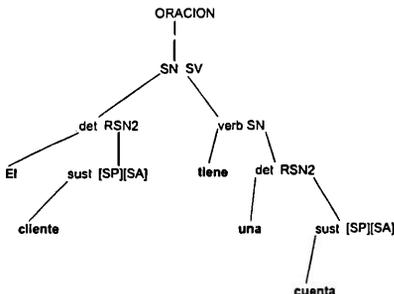


Figura 2: Árbol sintáctico del requisito El cliente tiene una cuenta

Realizado el anterior proceso, se procede a determinar el sentido del verbo. Del lexicon de verbos se extrae la información del verbo 'tener' (ver Tabla 2). Luego de esto, se presenta al usuario un menú con los sentidos del verbo tener (Get, hold, possess, have) y se solicita

que elija el sentido que mejor se ajuste al significado del requisito. Con lo cual, para este ejemplo, se obtiene el sentido possess el cual no presenta casos opcionales y cuyos casos obligatorios son: th y loc. La información del verbo tener luego de establecer sus sentidos se presenta en la Tabla 3.

Tabla 2: Información del verbo Tener extraída de Dorr (2001)

Sentido	Casos Obligatorios	Casos Opcionales
Get	agnt, th	src, ben
Get	agnt, ben, th	Src
Hold	agnt, th	Instr
Hold	agnt, th	Loc
Hold	exp, perc	
Hold	prop(que)	
Possess	th, loc	
Have	th, loc	
Hold	th, predicado posesional	
Hold	agnt, th, loc	

Tabla 3: Verbo 'Tener' luego de establecer su sentido

Verbo	Casos Obligatorios	Casos Opcionales
Tener	thm, loc	

Una vez establecido el sentido del verbo y la información de sus casos relacionados, se obtiene la información de los casos que puede desempeñar cada frase nominal del requisito (véase Tabla 4). Con la anterior información se procede a la determinación de los casos obligatorios ausentes mediante el apareamiento de los casos del verbo tener con los casos de las frases nominales.

Tabla 4: Información de las frases nominales extraída del lexicon de sustantivos

Frase Nominal	Casos Posibles*
Cliente	agnt, exp, loc, perc
Cuenta	th, exp, perc

Verbo: Tener

Casos obligatorios del verbo: th, loc

Frase Nominal: Cliente

Posibles casos cliente: agnt, exp, loc, perc

Resultado del apareamiento: Cliente = loc

El apareamiento de casos da como resultado que Cliente desempeña el caso loc en el requisito. Por lo tanto, el caso loc no puede presentarse de nuevo en la misma.

Verbo: Tener

Casos obligatorios del verbo: th, loc

Frase Nominal: Cuenta

Posibles casos cliente: th, exp, perc

Resultado del apareamiento: Cuenta = th

El verbo tener presenta los casos obligatorios *th*, *loc*, y no tiene casos opcionales, por lo tanto puede omitirse la evaluación de casos opcionales y se concluye que el requisito no presenta casos ausentes, lo que significa que está completo.

5 EVALUACIÓN DE LA EFECTIVIDAD DEL ANALIZADOR DE COMPLETITUD

Para la evaluación del analizador de completitud se siguieron los principios básicos de medición de Roche (1994), los cuales son caracterizados en las siguientes 5 actividades:

- Formulación.
- Colección.
- Análisis.
- Interpretación.
- Realimentación.

Basados en el factor de calidad de fiabilidad propuesto por MacCall, Richards y Walters (1977), el cual hace referencia al punto que se espera que un programa lleve a cabo su función con la exactitud pretendida, se definieron dos indicadores de precisión: Precisión casos presentes y Precisión casos ausentes. El cálculo de estos indicadores se realiza utilizando precisión microaveraging [Lewis (1992), la cual consiste en determinar un sólo valor de precisión medio para los casos presentes y para los casos ausentes de una especificación de requisitos. Los dos indicadores se definieron de la siguiente forma:

- Precisión casos presentes: Mide el porcentaje de concordancia de los casos presentes detectados por el Analizador de Completitud contra los casos presentes que son identificados por el analista de sistemas.

Casos concordantes presentes / Total de casos presentes

- Precisión casos ausentes: Mide el porcentaje de concordancia de los casos ausentes detectados por el Analizador de Completitud contra los casos ausentes que son identificados por el analista de sistemas.

Casos concordantes ausentes / Total casos ausentes

Por otra parte, se plantearon las siguientes hipótesis para determinar si el analizador de completitud es efectivo en sus resultados:

Ho₁: El analizador de completitud es preciso al identificar casos presentes en un requisito con respecto al analista de requisitos.

Ho₂: El analizador de completitud es preciso al identificar casos ausentes en un requisito con respecto al analista de requisitos.

La importancia de la evaluación radicó en determinar la efectividad del analizador de completitud en la identificación de casos presentes y ausentes en una especificación de requisitos. Las anteriores hipótesis nulas son rechazadas si los resultados difieren de sus declaraciones. Por consiguiente se especificaron las siguientes hipótesis alternativas:

H1₁: El analizador de completitud es impreciso al identificar casos presentes en un requisito con respecto al analista de requisitos.

H1₂: El analizador de completitud es impreciso al identificar casos ausentes en un requisito con respecto al analista de requisitos.

En la evaluación de la efectividad del analizador de completitud, se seleccionaron varios casos de estudio propuestos por algunos autores.

Cada caso de estudio fue dividido en requisitos, los cuales se formalizaron de acuerdo con las reglas de la gramática. Para cada requisito, se identificó el sentido del verbo que lo componía junto con sus casos obligatorios y opcionales, de acuerdo con el lexicón de verbos utilizado. También, cada requisito fue analizado previamente por un analista de sistemas, quien identificó los casos presentes y ausentes de acuerdo con su experiencia. Los casos de cada verbo, y los identificados por el analista de sistemas, se compararon con los resultados arrojados por el analizador de completitud sobre cada requisito, con el objetivo de establecer el porcentaje de precisión en la implementación del método utilizado.

En la Tabla 5 se presentan los resultados que arrojó el análisis de completitud sobre los casos de estudio seleccionados.

En la Tabla, las columnas son las siguientes:

RPC: Representa la cantidad de casos presentes concordantes entre el analista y el analizador de completitud.

TRP: Es el total de casos presentes detectados por el analista y el analizador.

PRP: Hace referencia al porcentaje de concordancia de casos presentes entre el analista y el analizador sobre el total de requisitos analizados.

RAC: Es la cantidad de casos ausentes concordantes entre el analista y el analizador en el análisis de completitud de un requisito.

Tabla 5: Resultado del análisis de completitud sobre los casos de estudio

CASO DE ESTUDIO	RPC	TRP	PRP(%)	RAC	TRA(%)	PRA	TR
[Harmain, 2000]	23	32	71.87	2	17	11.76	8
[Toory, 1980]	21	31	67.14	0	4	00.00	13
[Buchholz, 1991]	2	3	66.66	1	1	100.0	1
[Kristen, 1991]	5	6	83.33	0	1	00.00	2
[Cyre, 1995]	14	18	77.78	2	7	28.57	7
[Mayr, 2004]	12	15	80.00	0	6	00.00	7
TOTAL	77	105	73.33	5	36	13.88	38

TRA: Representa el total de casos ausentes detectados por el analista y el analizador.

PRA: Se refiere al porcentaje de concordancia de casos ausentes entre el analista y el analizador sobre el total de requisitos analizados.

TR: Corresponde al total de requisitos analizados.

6 ANÁLISIS DE LOS RESULTADOS

Con base en los resultados de la Tabla 5, el valor del indicador de precisión de casos presentes sobre la totalidad de casos de estudios analizados (73.33%), mostró que el analizador de completitud fue preciso al identificar los casos presentes en una especificación de requisitos. La mayoría de los casos presentes que fueron identificados por el analista de requisitos fueron identificados por el analizador de completitud, lo que le dio un grado de efectividad aceptable. Considerando el valor del indicador de precisión de casos presentes, se aceptó la hipótesis nula H_{01} , y por lo tanto, el analizador de completitud fue preciso al identificar casos presentes en un requisito con respecto al analista de requisitos.

Por otra parte, el valor del indicador de precisión de casos ausentes sobre la totalidad de casos analizados (13.88%), mostró que el analizador de completitud no fue preciso al identificar los casos ausentes en la especificación de requisitos. La mayoría de los casos ausentes que fueron identificados por el analista de requisitos difirieron de los casos ausentes identificados por el analizador de completitud, lo que incidió en un bajo grado de efectividad en el análisis.

La razón de que el analizador de completitud haya diferido tanto con el analista de requisitos sobre los casos ausentes estriba en la subjetividad del analista de requisitos, quien en su análisis consideró casos ausentes que para el analizador de completitud es imposible identificar porque no están definidos en el lexicon de verbos. Por lo tanto, teniendo presente el valor del indicador de precisión de casos ausentes, se rechazó la hipótesis nula H_{02} , y en consecuencia, según la hipótesis alternativa H_{11} , el analizador de completitud no fue preciso al identificar

casos ausentes en un requisito con respecto al analista de requisitos.

7 CONCLUSIONES Y TRABAJOS FUTUROS

En este artículo se presentó una herramienta para el análisis de completitud de requisitos escritos en Español restringido. Igualmente, se ha presentado una evaluación de la efectividad de la propuesta. Los resultados obtenidos permiten retroalimentar el proceso para el propósito de mejorar la completitud de las especificaciones de requisitos y, en última instancia, para el mejoramiento de la calidad del software.

Entre los trabajos futuros, se destaca la implementación del analizador de completitud con selección automática del sentido del verbo de acuerdo con el contexto o dominio del requisito, y la definición de una gramática con capacidad de reconocer requisitos con más de un verbo. Igualmente, añadir al método de análisis de completitud la capacidad de identificar, de entre un grupo de verbos, el que ha de ser apareado para la detección de los casos semánticos.

REFERENCIAS

- Ben Achour, C. (1999), Extraction des besoins par analyse de scénarios textuels, PhD thesis, Univerité Paris.
- Boehm, B. (1984), 'Verifying and validating software requirements and design specifications'. *IEEE Software* 1.
- Dong, L., Kalavani, S., Far, B. y Eberlein, A. (2003), Automatic transition from use-cases to class model, Technical report, Department of Electrical and Computer Engineering. University of Calgary.
- Dorr, B. (2001), *Computational Lexicon*, University of Maryland.
- Fabbrini, F. (2001), An automatic quality evaluation for natural language requirements, Technical report, Istituto di Elaborazione dell'Informazione del C.N.R. Pisa, Italy.

- Fliedl, G. (1999), The NTMS parser: an efficient computational linguistic tool, Technical report, University of Klagenfurt. Carinthia, Austria.
- Lami, G. (2004), An automatic tool for the analysis of natural language requirements, Technical report, Information Science and Technology Institute "A. Faedo", Pisa, Italy.
- Lewis, D. (1992), Representation and learning in information retrieval, PhD thesis, Department of Computer and Information Science, University of Massachusetts.
- MacCall, J., Richards, P. y Walters, G. (1977), 'Factors in software quality'.
- Ohnishi, A. (1994), Customizable software requirements languages, Technical report, Department of Computer Science, Ritsumeikan University. Shiga, Japan.
- Roche, J. (1994), 'Software metrics and measurement principles', *Software Engineering Notes, ACM* 19(1), 76-85.
- Rolland, C. (1992), A natural language approach for requirements engineering, in 'Proceedings of the Fourth International Conference CAISE'92 on Advanced Information Systems Engineering'.
- Wilson, W. (2000), Automated quality analysis of natural language requirement specifications, Technical report, Software Assurance Technology Center.

