

Plan de Trabajo Final
Carrera: Ingeniería de Sistemas
Facultad de Ciencias Exactas – UNICEN

Tema: “Recuperación de Trazabilidad entre Documentos de Diseño y Requerimientos mediante Técnicas Semi-Automáticas”.

Alumno/s: Attanasio Ruiz German y Gonzalez Rodrigo Damian

Director: Mg. Alejandro Rago

Codirector: Dra. Claudia Marcos

1. Introducción

La definición de los requerimientos de un sistema es una actividad crítica en el desarrollo de software. El análisis de los mismos sirve fundamentalmente para comprender las necesidades de los stakeholders y, de esta manera, poder construir un sistema adecuado [16]. Los analistas tienen la responsabilidad de organizar y documentar dichos requerimientos de manera clara, lo que facilita su análisis, comunicación e implementación. La elicitación de requerimientos incompleta o de mala calidad puede tener consecuencias negativas en las etapas posteriores del desarrollo [11].

Los documentos de requerimientos (en conjunto con otros artefactos) son utilizados en las etapas de análisis y diseño para generar la arquitectura de software del sistema. La arquitectura de software puede definirse como: “La estructura o estructuras del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos”. El documento de la arquitectura es el vehículo principal para la comunicación entre los diferentes stakeholders [6] y presenta diferentes perspectivas y vistas del diseño de un sistema. Sin una arquitectura adecuada que satisfaga la funcionalidad requerida y al mismo tiempo cumpla con los atributos de calidad deseados, el proyecto será propenso al fracaso [16].

Si bien las etapas de captura de requerimientos y creación de la arquitectura están estrechamente relacionadas, las actividades asociadas con dichas etapas son llevadas a cabo por separado y con una interacción limitada [7]. En la práctica, el proceso de transición de requerimientos a arquitectura está (generalmente) basado en la experiencia, intuición, comunicación y conocimiento del dominio de arquitectos y diseñadores [4]. Evaluar la calidad de la transición es difícil debido a que la recuperación de la trazabilidad es compleja y muchas veces inexistente. Una traza

entre requerimientos y componentes de diseño hace explícita la información de transición de un requerimiento (desde que el mismo fue definido) hasta que éste fue satisfecho por la arquitectura (mediante componentes y tácticas arquitectónicas), y viceversa.

2. Motivación

Establecer la trazabilidad entre los requerimientos y los componentes arquitectónicos que los satisfacen, como así también mantener estas asociaciones actualizadas a medida que el sistema evoluciona, es importante por las siguientes razones:

- Ayuda a los stakeholders a ver el impacto de los requerimientos sobre la arquitectura.
- Facilita las tareas de análisis ante cambios en los requerimientos, simplificando la incorporación de nueva funcionalidad y el estudio de su impacto en la arquitectura.
- Asegura que la arquitectura contemple todos los requerimientos del sistema.
- Asegura que la documentación de requerimientos esté completa y que esta sea consistente con las decisiones tomadas en el diseño de la arquitectura.

En la industria, la ausencia de estos registros de trazabilidad entre artefactos se atribuye principalmente a que realizar los análisis pertinentes para obtenerlos es un proceso largo, costoso y propenso a errores. Adicionalmente, es necesario la persona que realiza el análisis tenga que conocer detalladamente los requerimientos y la arquitectura del sistema. Aunque es posible automatizar esta tarea, dado que existen técnicas y estudios para la recuperación de información de documentos de requerimientos y diseño, no hemos encontrado evidencias de trabajos que se enfoquen en la recuperación de trazabilidad entre dichos documentos.

3. Objetivos

El objetivo del presente trabajo es diseñar e implementar una herramienta que ayude al arquitecto a recuperar la trazabilidad entre documentos de requerimientos y documentos de arquitectura. Nuestra hipótesis es que, tanto en los documentos de requerimientos como en la documentación de la arquitectura existen indicios que permitirían inferir las trazas de forma automática. La herramienta analizará estos artefactos mediante técnicas de procesamiento de texto y proveerá soporte para que el arquitecto pueda obtener información acerca del impacto que tuvo un requerimiento sobre la arquitectura o cuáles de los requerimientos no fueron tenidos en cuenta en el diseño de la arquitectura, entre otras. Específicamente, el proceso de búsqueda de trazas provisto por herramienta funcionará en tres grandes fases (las dos primeras independientes entre sí).

El propósito de la primer fase es localizar porciones de funcionalidad que tengan influencia en la arquitectura de software. Particularmente, se pretende identificar crosscutting concerns en documentos de requerimientos. Un concern se define como "cualquier cuestión de interés en un sistema de software"[19], y cuando algún concern no puede ser fácilmente encapsulado es denominado crosscutting concerns (CCC) [20]. Los CCCs son un buen punto de partida para analizar la trazabilidad ya que están (fuertemente) relacionados con los atributos de calidad de un sistema [16]. Para llevar a cabo la detección de crosscutting concerns se explorarán técnicas de Procesamiento de Lenguaje Natural (NLP) [17, 18, 3] y Aprendizaje de Maquina (ML) [10]. Adicionalmente, en esta fase no sólo tratará de identificar los CCCs sino también relacionarlos con conceptos propios del diseño arquitectónico.

La segunda fase tiene como propósito analizar los documentos de diseño en búsqueda de indicios acerca de las decisiones de diseño ya tomadas para satisfacer los atributos de calidad y asimismo implementar la funcionalidad del sistema. Particularmente, estamos interesados en identificar conceptos tales como estilos arquitectónicos, patrones de diseño, atributos de calidad y tácticas arquitectónicas. En cuanto a la detección de dichos conceptos, se estudiarán técnicas de NLP y ML (de forma similar a la primer etapa), pero teniendo en cuenta la diferencia de información y estructura que presentan los documentos de diseño y focalizando sobre los conceptos específicos listados arriba. A tal efecto, se está explorando la utilización de diferentes taxonomías y clasificaciones de conceptos de arquitectura las cuales permitirán diferenciar los conceptos recuperados, y asimismo facilitarán su asociación con los CCCs identificados previamente.

La tercera y última fase de esta propuesta tiene el objetivo recuperar la trazas entre los CCCs y los conceptos arquitectónicos extraídos en las fases anteriores. Para llevar a cabo esta tarea, se analizará la aplicación de técnicas automáticas tales como clasificadores bayesianos (BC) [2] y Latent Semantic Indexing (LSI), entre otras. Por un lado, BC es una técnica fundamentada en el teorema de Bayes la cual permite encontrar relaciones entre diferentes términos del texto. Por otro lado, Latent Semantic Indexing (LSI) [8] es una técnica de NLP basada en el principio de que las palabras que se utilizan en el mismo contexto tienden a tener significados similares, y sirve para analizar la relación entre un conjunto de documentos y los términos que ellos contienen. Ambas técnicas tienen un gran potencial para llevar a cabo la recuperación automática de trazas.

4. Trabajos Relacionados

Los trabajos relacionados se pueden organizar en dos grupos. El primer grupo está formado por enfoques y herramientas abocadas a analizar documentos de requerimientos, y cuyos objetivos van desde la detección y clasificación de

requerimientos no funcionales [5], hasta la identificación de CCCs. Tales trabajos proponen la utilización de técnicas de NLP [17, 3, 1, 18] y/o ML [4] para cumplir sus objetivos. El segundo grupo abarca trabajos relacionados a la identificación de conceptos en los documentos de diseño para luego determinar su trazabilidad tanto con código fuente [9, 14] como con casos de pruebas [13]. Es necesario un estudio más detallado para considerar trabajos que relacionen especificaciones de requerimientos y documentos de diseño.

Dentro de los trabajos del primer grupo, se destaca la herramienta REAssistant [16], la cual asiste a los analistas de requerimientos a recuperar CCCs a partir de especificaciones de casos de uso. REAssistant utiliza dos técnicas basadas en semántica para buscar concerns latentes: clustering semántico entre palabras y un lenguaje de consultas (es decir, de búsqueda) enriquecido con semántica. Esta última técnica, que involucra un lenguaje de consultas, podría llegar a utilizarse con algunas modificaciones menores para recuperar conceptos de documentos de diseño. Otra herramienta interesante es Theme/Doc [3], la cual permite detectar CCCs en documentos de requerimientos de manera visual resaltando los concerns que afectan múltiples documentos. La herramienta utiliza como entradas una lista de acciones y otra de entidades, las cuales el analista tiene que proveer a priori (luego de leer los documentos de requerimientos) para que la identificación sea efectiva. 3CI [1] se basa en las ideas de Theme/Doc y automatiza aún más el proceso de identificación de CCCs, evitando la necesidad de tener la lista de acciones y de entidades. Esta herramienta utiliza técnicas de NLP para detectar verbos y agruparlos semánticamente en una matriz que representa la distribución de verbos en los diferentes requerimientos. Para detectar CCCs, 3CI analiza esta matriz y la distribución de los verbos en los requerimientos. La herramienta EAMiner [18] también permite a los analistas identificar potenciales aspectos tempranos (CCCs) de forma asistida. Para ello, trabaja en dos etapas, primero etiquetando cada palabra con su “parte del discurso” (POS) y luego utilizando una taxonomía para categorizar cada palabra de acuerdo a su significado semántico y su relación con CCCs preestablecidos. La idea de utilizar una taxonomía para agrupar conceptos similares podría llegar a aplicarse sobre los documentos de diseño y de esa forma agrupar conceptos semánticamente similares como patrones o tácticas arquitectónicas.

Con respecto al segundo grupo de trabajos, focalizados en el análisis de documentos de diseño, podemos mencionar el trabajo de Marcos et al. [15]. Dicho trabajo presenta una herramienta que recupera vínculos desde el código fuente a los documentos de diseño. Esta herramienta utiliza NLP en ambos documentos para identificar sustantivos y Latent Semantic Indexing (LSI) para descubrir vínculos entre los documentos en base a los sustantivos detectados. De forma similar, Kuhn et al. exploró el uso de LSI para enriquecer la documentación de un sistema recuperando información del mismo a partir del código fuente [12]. Finalmente, en [2] se utilizó un clasificador bayesiano para recuperar la trazabilidad del código

fuelle a los documentos de diseño. Éste construye un modelo probabilístico que permite relacionar el código fuente con los documentos de diseño. Sin embargo, dicho modelo presenta un problema ya que necesita ser entrenado con un número considerable de documentos para poder recuperar la trazabilidad correctamente.

5. Cronograma de actividades

Se plantea realizar el trabajo final en 12 meses, las actividades y su respectiva duración aproximada están enumeradas a continuación.

Actividad	Duración Estimada
Relevamiento bibliográfico sobre técnicas de NLP, LSI y ML aplicados en artefactos textuales de software.	2 mes
Análisis de técnicas de NLP y ML para recuperar y clasificar CCC de documentos de requerimientos.	1 mes
Análisis de técnicas de NLP, ML y mecanismos de búsqueda para recuperar conceptos arquitectónicos de los documentos de diseño. Análisis de taxonomías de atributos de calidad para clasificar los conceptos recuperados de los documentos de diseño.	1 mes
Análisis y evaluación de técnicas de procesamiento de lenguaje natural y recuperación de información para descubrir trazabilidad entre documentos de requerimientos y documentos de diseño.	2 meses
Diseño de los módulos que reconozcan conceptos de arquitectura en documentos de diseño utilizando técnicas de NLP. Diseño de los módulos para recuperar la trazabilidad entre conceptos de diseño y de requerimientos.	2 meses
Implementación de los módulos de reconocimiento de conceptos en documentos de arquitectura y trazabilidad con documentos de requerimientos.	1 mes
Prueba de la herramienta sobre casos de estudio reales para evaluar su efectividad.	1 mes
Documentación y elaboración de informe final.	2 meses

6. Bibliografía

- [1] Busyairah Syd Ali and Zarinah Mohd Kasirun. 3ci: A tool for crosscutting concern identification. In *Computational Intelligence for Modelling Control & Automation*, 2008 International Conference on, pages 351–355. IEEE, 2008.
- [2] Giuliano Antoniol, Gerardo Canfora, Andrea De Lucia, and Ettore Merlo. Recovering code to documentation links in OO systems. In *Reverse Engineering*, 1999. Proceedings. Sixth Working Conference on, pages 136–144. IEEE, 1999.
- [3] Elisa Baniassad and Siobhán Clarke. Finding aspects in requirements with theme/doc. *Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design*, page 16, 2004.
- [4] Agustin Casamayor. *Assisting Architectural Software Design with Text Mining Techniques*. PhD thesis, Universidad Nacional del Centro, Facultad de Ciencias Exactas, 2011.
- [5] Jane Cleland-Huang, Raffaella Settini, Xuchang Zou, and Peter Solc. Automated classification of non-functional requirements. *Requirements Engineering*, 12(2):103–120, 2007.
- [6] Paul Clements. *Comparing the sei’s views and beyond approach for documenting software architecture with ansi-ieee 1471-2000*. Technical report, DTIC Document, 2005.
- [7] R. Kazman D. M. Berry and R. Wieringa. *Second international software requirements to architectures workshop*. 2003.
- [8] Scott C. Deerwester, Susan T Dumais, and Richard A. Harshman. *Indexing by latent semantic analysis*. 1990.
- [9] Andrian Marcus Denys Poshyvanyk. *Using traceability links to assess and maintain the quality of software documentation*. TEFSE, 2007.
- [10] Czibula Istvan Gergely. *Use of search techniques to software development*. PhD thesis, BabesBolyai University, Faculty of Mathematics and Informatics, 2009.
- [11] Hubert F Hofmann and Franz Lehner. *Requirements engineering as a success factor in software projects*.
- [12] Adrian Kuhn, Stéphane Ducasse, and Tudor Gîrba. Semantic clustering: Identifying topics in source code. *Information and Software Technology*, 49(3):230–

243, 2007.

[13] Marco Lormans and Arie Van Deursen. Can lsi help reconstructing requirements traceability in design and test? In *Software Maintenance and Reengineering*, 2006. CSMR 2006. Proceedings of the 10th European Conference on, pages 10–pp. IEEE, 2006.

[14] Yoelle S Maarek and Frank Z. Smadja. Full text indexing based on lexical relations an application: software libraries. In *ACM SIGIR Forum*, volume 23, pages 198–206. ACM, 1989.

[15] Andrian Marcus and Jonathan I. Maletic. Recovering documentation-to-source-code traceability links using latent semantic indexing. In *Proceedings of the 25th International Conference on Software Engineering, ICSE '03*, pages 125–135, Washington, DC, USA, 2003. IEEE Computer Society.

[16] A. Rago. Tool support for identifying crosscutting concerns in use case specifications. Master thesis in software engineering, UNICEN University ISISTAN Research Institute, March 2013.

[17] Alejandro Rago, Esteban Abait, Claudia Marcos, and Andrés Diaz-Pace. Early aspect identification from use cases using nlp and wsd techniques. In *Proceedings of the 15th workshop on Early aspects*, pages 19–24. ACM, 2009.

[18] Américo Sampaio, Ruzanna Chitchyan, Awais Rashid, and Paul Rayson. Eaminer: a tool for automating aspect-oriented requirements identification. In *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering*, pages 352–355. ACM, 2005.

[19] Stanley M Sutton Jr and Isabelle Rouvellou. Advanced separation of concerns for component evolution. Citeseer.

[20] Stanley M Sutton Jr and Isabelle Rouvellou. Modeling of software concerns in cosmos. In *Proceedings of the 1st international conference on Aspect-oriented software development*, pages 127–133. ACM, 2002.

ATTANASIO RUIZ
CU: 246193

.....
Firma del Alumno

Rodrigo Gonzalez
CU 245710

.....
Firma del Alumno

Avalo la presente solicitud de evaluación,

.....
Firma del Director

.....
Firma del Co-director