

Reverse Engineering in Software Requirements in web applications using a LEL *

Angela Verónica Granizo Rodríguez¹[0000-0001-8117-275X], Leandro
Antonelli²[0000-0003-1388-0337], Sergio Firmenich^{2,3}[1234-5678-9012], and Diego
Firmenich⁴[0000-0002-7212-4454]

¹ Escuela Superior Politécnica de Chimborazo. Panamericana Sur km 1 y 1/2 .
Riobamba. Ecuador

agranizo@esPOCH.edu.ec
sergio.firmenich@lifia.info.unlp.edu.ar

² LIFIA, Facultad de Informática, UNLP. La Plata. Calle 50 y 120, S/N. Buenos
Aires. Argentina.

lanto@lifia.info.unlp.edu.ar

³ CONICET. Godoy Cruz 2290. Ciudad Autónoma de Buenos Aires. Argentina.

⁴ Departamento de Informática, Facultad de Ingeniería, Universidad Nacional de la
Patagonia San Juan Bosco. Argentina.

diego.firmenich@gmail.com

Abstract. This document explores and describes the state of the art of Reverse Engineering in the specification of Software Requirements for web applications using an LEL (Language Extended Lexicon). A review of the scientific literature on this topic was carried out with the aim of investigating how the application of Reverse Engineering in the specification of software requirements would improve their quality, with the support of the creation of a tool for web augmentation, that is, a web extension that allows creating a glossary of terms or LEL of web applications, this with the purpose of knowing their operation for maintenance and improvement purposes.

Keywords. Software Requiriments, Reverse Engineering, Language Extended Lexicon.

* Universidad Nacional de La Plata

1 Introducción/Motivación

La mayoría de los problemas que se dan en la entrega de los sistemas tienen que ver con conceptos erróneos de requisitos en la captura de requerimientos, lo que provoca falencias en su calidad. Por ello, es indispensable un proceso de ingeniería de requisitos (RE) para cumplir con los objetivos de tiempo, costo y calidad. Cuando las aplicaciones web ya fueron construidas o en sistemas ya existentes o heredados, ya se

plasmaron los requerimientos y si estos tenían errores, para lograr conocer su funcionamiento para mantenimiento y mejoramiento se puede recurrir a Ingeniería Inversa; extraer los requerimientos de software. En este caso aparece la necesidad de contar con un LEL, que facilite el conocimiento del dominio de la aplicación. De acuerdo a los antecedentes descritos anteriormente, se puede plantear la siguiente pregunta de investigación ¿cómo se puede mejorar la calidad de los requerimientos de software en aplicaciones web aplicando Ingeniería Inversa y utilizando un LEL?

El presente trabajo describe el estado del arte y la propuesta de un enfoque que utilice Ingeniería Inversa en la especificación de requerimientos de software de aplicaciones web utilizando un LEL (Léxico Extendido del Lenguaje). Por lo tanto, vamos a crear herramientas de aumentación web como una extensión web que permita extraer contenidos del DOM que sirvan como referencia para la especificación de requerimientos de software.

2 Estado del arte

Las Empresas que poseen sistemas informáticos obsoletos se enfrentan a una encrucijada; continuar utilizando los sistemas heredados, realizar sus cambios requeridos y sus costos se incrementarán inevitablemente, por otro lado, reemplazar sus sistemas heredados con sistemas nuevos, esto también tendrá un costo y puede suceder que los nuevos sistemas no provean apoyo efectivo como los sistemas heredados. Lo mismo ocurre con las aplicaciones web que son sistemas, pero funcionan en la world wide web y son el motivo de investigación. Se puede referir varias razones del costo de mantenimiento de sistemas heredados o aplicaciones web con cierta antigüedad, por ejemplo: diferentes equipos han implementado distintas partes de la aplicación web, es decir existirán estilos de programación no consistentes. Todo o parte de la aplicación web se implementó usando un lenguaje de programación que luego ya es obsoleto, con dificultad se puede encontrar personal con conocimientos de los mismos o consultoría externa costosa, para dar mantenimiento. La documentación de la aplicación web no es apropiada y no está actualizada, la única existente es el código fuente o únicamente se cuenta con la versión ejecutable.

De acuerdo a este preámbulo, la ingeniería inversa de software es la forma de recuperar los requisitos o las especificaciones de diseño de un sistema, su objetivo es extraer muchos tipos de información del software existente y usarla para la renovación, mejor comprensión y ayuda en el mantenimiento de los sistemas heredados [1]. La ingeniería inversa es el proceso de analizar un sistema para identificar sus componentes y sus interrelaciones y crear representaciones del mismo en otra forma o en un nivel más alto de abstracción [2]. Las técnicas de ingeniería inversa existentes se centran en extraer la información de las funciones centrales y de las reglas negocio del software heredado, que se llama primary concerns [3].

Por otro lado, un proceso de ingeniería de requisitos es crucial para cumplir con los objetivos de tiempo, costo y calidad. Es necesario recuperar los requisitos del resultado de ingeniería inversa del sistema heredado y al integrar este resultado en la fase de requisitos del ciclo de vida del software, es posible tener una mejor obtención de requisitos [1].

Así también el Léxico Extendido del Lenguaje (LEL) o glosario de términos en la elicitación de requerimientos se ha propagado entre desarrolladores e investigadores en la Ingeniería de Requisitos. El glosario es un artefacto de especificación de requisitos de software utilizado para describir los elementos centrales en el contexto de la aplicación mediante la descripción de su vocabulario. El LEL apoya la construcción de un glosario y clasifica los símbolos en Sujeto, Objeto, Verbo y Estado, estos a su vez se describen con los atributos de noción e impacto [4].

A continuación se describe una revisión de la literatura de varios autores; en primer lugar Wang et al. (2006), presentaron un enfoque de Ingeniería Inversa para recuperar el diagrama de clase UML del código fuente de Java [5]. Fahmi et al. (2007), expusieron un modelo revisado del proceso de reingeniería tradicional y también describieron la racionalidad del modelo propuesto. Han discutido sobre la ingeniería inversa de software, la ingeniería de requisitos, sus prácticas y actividades básicas. Mencionan que, si los requisitos del sistema heredado se conocen y se consideran al hacer la nueva especificación de requisitos, seguramente se puede mejorar la fase de requisitos. Argumentan la necesidad de obtener requisitos de los resultados de la ingeniería inversa [1]. Su et al. (2008), propusieron un framework de Ingeniería Inversa de software orientado a aspectos para la solución de comprensión y evolución de problemas de propiedades transversales (crosscutting properties) en sistemas heredados. El objetivo de este enfoque fue recuperar preocupaciones transversales del sistema a nivel de requisitos [3]. Aman et al. (2013), presentan un framework de ingeniería inversa de XML a UML para la generación de especificación de requisitos de software [6].

Todos los trabajos relacionados son muy relevantes y como se puede notar el uso de la Ingeniería Inversa de ha hecho presente de distintas formas, pero nadie se enfoca en el uso del LEL, si bien hacen Ingeniería Inversa en requerimientos de software, no se apoyan en el uso del LEL que es mi enfoque. Como propuesta, se plantea incluir el LEL en la especificación de requerimientos de aplicaciones web aplicando Ingeniería Inversa para conocer su funcionamiento con fines de mantenimiento y mejoramiento.

3 Planteamiento del problema

Como problema podemos describir, que la complejidad en la extracción de requerimientos desde sistemas heredados hace que los mismos tengan falencias en su calidad, puesto que no se cuenta con un documento de requerimientos y peor aún documentación con un léxico extendido del lenguaje (LEL), es decir que este enfoque permitirá mejorar la calidad de los requerimientos de software. Para alcanzar este objetivo se aplicará Ingeniería Inversa a aplicaciones web para conocer su funcionamiento con fines de mantenimiento y mejoramiento de las mismas. La contribución, es que no existe un enfoque que utilice Ingeniería Inversa para especificar requerimientos utilizando un LEL; por tanto, vamos a crear herramientas de aumentación web como una extensión web que permita extraer contenidos del DOM que sirvan como referencia para la especificación de requerimientos de software.

4 Metodología y enfoque de investigación

El enfoque de esta investigación es cuantitativo. Dentro del diseño de la investigación utilizaremos el método experimental, pues vamos a seleccionar dos grupos uno de control y otro experimental para probar la herramienta desarrollada. El tipo de investigación, es correlacional con dos variables involucradas; la independiente que será el enfoque de extracción de requerimientos de aplicaciones web y la variable dependiente la calidad de los requerimientos. La idea es verificar si la correlación entre estas variables es significativa.

5 Plan de evaluación

Dentro del enfoque propuesto, se está desarrollando la herramienta de aumentación web, es decir la extensión web de una aplicación ya construida, misma que permite crear el LEL. También se está trabajando en una revisión sistemática acerca de Ingeniería Inversa en Requerimientos de Software. Posteriormente se realizará un experimento con analistas de desarrollo a los cuales se dividirá en dos grupos, con y sin el enfoque y veremos los resultados.

6 Etapa doctoral

Early Stage -ES-

References

1. S. A. Fahmi and H.-J. Choi, "Software Reverse Engineering to Requirements," 2007, Accessed: Mar. 05, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/4420580/>.
2. E. J. Chikofsky and J. H. Cross, "Reverse Engineering and Design Recovery: A Taxonomy," IEEE Softw., vol. 7, no. 1, pp. 13–17, 1990, doi: 10.1109/52.43044.
3. Y. Su, X.-W. Zhou, and M.-Q. Zhang, "Approach on Aspect-Oriented Software Reverse Engineering at Requirements Level," in 2008 International Conference on Computer Science and Software Engineering, 2008, vol. 2, Accessed: Mar. 06, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/4722062/>.
4. M. Urbieta, L. Antonelli, G. Rossi, and J. C. S. do Prado Leite, "The impact of using a domain language for an agile requirements management," Inf. Softw. Technol., vol. 127, p. 106375, Nov. 2020, doi: 10.1016/J.INFSOF.2020.106375.
5. X. Wang and X. Yuan, "Towards an AST-Based Approach to Reverse Engineering," 2006, Accessed: Mar. 06, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/4054609/>.

6. H. Aman and R. Ibrahim, "Reverse Engineering: From Xml to Uml for generation of software requirement specification," 2013, Accessed: Mar. 16, 2022. [Online]. Available:<https://ieeexplore.ieee.org/document/6637575/>.