

Derivación de Casos de Prueba a Partir de Escenarios

Gladys Kaplan^{1,2}, Jorge Doorn^{2,3}, Walter Panessi¹, Claudia Ortiz¹, Eugenia Céspedes¹

¹Departamento de Ciencias Básicas, Universidad Nacional de Luján (UNLu)

²Departamento de Ingeniería e Innovación Tecnológica, Universidad Nacional de La Matanza (UNLaM)

³Escuela de Informática, Universidad Nacional del Oeste (UNO)

(gkaplan@unlam.edu.ar, jdoorn@exa.unicen.edu.ar, wpanessi@unlu.edu.ar, cortiz@unlu.edu.ar, eugeniacespedes@outlook.com)

Resumen

Como se describió en un artículo precedente [Kaplan 15], este proyecto tiene por objetivo concebir un mecanismo para generar tempranamente los casos de prueba partiendo del conocimiento obtenido en la etapa de Ingeniería de Requisitos (IR), particularmente del proceso de requisitos basado en escenarios [Leite 97] [Leite 04]. En dicha oportunidad se presentó la propuesta general del proyecto, mientras que en el presente artículo se profundiza la relación *escenarios futuros – casos de prueba* (EF-CP) y se describe un mecanismo de derivación automática. En muchos casos, la etapa de prueba es la variable de ajuste cuando los tiempos y el presupuesto de un proyecto están en crisis. Lograr que las pruebas, total o parcialmente, se realicen después de cada modificación asegura la calidad de los requisitos del software. Esto fuerza la necesidad de una creación, actualización y ejecución de los CP automática, con el objetivo de probar el software tantas veces como sea necesario. Que además permitan la mínima distracción de las etapas en desarrollo y a

un bajo costo es indispensable para que no se transformen en un elemento de perturbación. Cabe destacar que, al ser una actividad transversal en el proceso de construcción del software, estos CP evolucionan durante el diseño ya que es cuando se conoce en qué componente de software se alojará cada requisito (software allocation). Este nuevo conocimiento permite generar CP de caja blanca con un alto nivel de detalle.

Palabras clave: ingeniería de requisitos, casos de prueba funcionales, escenarios futuros.

Contexto

Este proyecto es una continuación de otros realizados en diferentes Universidades. Desde el año 1995 a la fecha se ha estudiado el proceso de requisitos basado en escenarios [Leite 04], utilizado en el presente artículo, donde se estableció una estrategia para realizar un proceso de requisitos a partir de una necesidad del usuario-cliente hasta obtener un documento de requisitos de software de alta calidad. En este proceso, se describen los modelos con sus

procesos de construcción. Los modelos utilizados son el Léxico Extendido del Lenguaje (LEL), los Escenarios (EA y EF), la Ficha de Información Extemporánea (FiE), el LEL de Requisitos (LEL_R) y la Especificación de Requisitos del Software (ERS).

Introducción

Como se mencionó en la sección anterior, el presente proyecto basa su estudio en el proceso de requisitos [Leite 04], el cual crea un glosario denominando LEL [Leite 90] [Hadam 08] del dominio de la aplicación. Durante la construcción del LEL y de un conjunto de escenarios actuales (EA) que representan situaciones observables, aparece la Información Extemporánea (FiE) que registra conocimiento que no puede ser representado en el modelo en construcción en ese momento. Con el LEL, los EA y las FiE se construye un conjunto de escenarios futuros [Doorn 02] que representan situaciones no observables del contexto futuro. Estos escenarios se validan y negocian hasta obtener la versión final que representan los procesos del negocio para cuando el sistema de software se encuentre en ejecución. Es de estos escenarios futuros de donde se extraen los requisitos del software.

Las pruebas construidas tempranamente en el proceso de construcción del software son del tipo funcional ya que verifican QUÉ se debe probar dejando para un momento posterior el CÓMO se debe hacer. Es así que las pruebas funcionales son del tipo caja negra,

mientras que las pruebas durante el diseño y del código son del tipo caja blanca.

Existen diversos trabajos relacionados con la generación de casos de prueba durante la etapa de IR a partir de un modelo existente. Entre los más conocidos están la generación de CP a partir de Casos de Uso [Riebisch 03] [Palacio 09] [Odalys 06] [Correa 11] [Gutiérrez 06], a partir de algoritmos metaheurísticos [Blanco 9] utilizando el lenguaje de transformación QVT, a partir de diagramas de secuencia extendidos [Lamancha 09], a partir de diagramas de actividad [Mingsong 09] [Linzhang 04] [Boghdady 12] [Boghdady 11], a partir de diagramas de secuencia [Javed 07], a partir de diagramas de estado [Badri 09], a partir de UML [Abbors 10] [Correa 12] [Wendland 11] [Salem 11], a partir de especificación de Requisitos [Hasling 08] entre otros.

Las pruebas son una etapa del proceso de construcción del software que determina, junto a las actividades de validación, la calidad del nuevo sistema de software a construir. Paradójicamente tienden a minimizarse cuando los tiempos o los costos de un proyecto entran en crisis debido a una mala estimación, por una pobre administración del proyecto o por la presencia de más cambios que los esperados. Este último caso, el de muchos cambios en los requisitos, aumenta considerablemente el costo del producto en construcción debido a la necesidad de actualizar constantemente los CP y como consecuencia, repetir todas las pruebas

para asegurar la consistencia de todo el software. Para evitar que las pruebas sean una actividad regulable según las necesidades del proyecto y no del producto, es que se hace indispensable automatizar la generación de los CP encontrando algún mecanismo de prueba lo más automático posible.

A lo largo del proyecto se espera probar la hipótesis de trabajo subyacente que considera que la automatización de la generación de los CP permite sostener en el tiempo la calidad del producto a construir.

Líneas de Investigación, Desarrollo e Innovación

Para alcanzar el objetivo final del proyecto se determinaron diferentes hitos. En una primera etapa se analizaron los CP funcionales y su relación con los modelos de requisitos. En la segunda etapa del proyecto se construyó una plantilla para registrar los casos de prueba. En la etapa que se presenta en este artículo se generó un proceso de derivación desde los EF a la plantilla de CP. El proceso para realizarlos es que por cada EF se abstraen los objetivos funcionales a validar y se generan los CP necesarios. La construcción de escenarios propone que cada EF debe satisfacer un objetivo y sus subobjetivos. Dado que un CP puede satisfacer un subobjetivo que esté incluido en varios escenarios y como un objetivo de un escenario puede requerir varios CP para probarlo, la relación entre EF y CP es de M:N.

Luego, se espera realizar la comprobación

de los casos de prueba con los casos de aceptación para analizar la posibilidad de automatizar el proceso.

Como línea complementaria de investigación se analiza el impacto de las jerarquías conceptuales de los léxicos representados en los glosarios de requisitos, específicamente en el LEL, y se realizó una inspección para asegurar su inclusión en el glosario.

Resultados y Objetivos

Como se mencionó [Kaplan 15] se cuenta con aproximadamente 150 casos de estudio para probar el proceso. Se han tomado estos casos para realizar las derivaciones y refinar el proceso.

Los casos de prueba se utilizan para verificar los requisitos de software. Es justamente desde la ERS, desde donde se pueden reconocer las trazas hacia los EF que le dieron origen, con el objetivo de identificar el conocimiento contextual de los requisitos del software y generar así los CP, tomando solo dicha información. Los casos realizados han demostrado que la información del EF es suficiente para generar un CP. Esto nos permitió definir la plantilla de CP donde se deriva casi automáticamente la información del EF. Se analizó que los casos de aceptación son un buen mecanismo para validar el CP semi automáticamente.

Se espera analizar la posibilidad de que los CP no existan, sino que en el momento de su recuperación se construyan automáticamente. Esto haría que estén siempre actualizados los CP ante cambios en los requisitos. Se

estudiará la posibilidad de automatizar la generación de los CP evitando la intervención humana y beneficiando así que las pruebas puedan estar siempre actualizadas.

Formación de Recursos Humanos

Se planifica la finalización de la carrera de grado Licenciatura en Sistemas de Información de la alumna Eugenia Cespedes (30 materias aprobadas) y del alumno Julián Massolo (25 materias aprobadas) en UNLu. También la presentación de la tesis de Maestría en Ingeniería de Software de Claudia Ortiz y Walter Panessi en UNLP y la finalización de la tesis doctoral de Gladys Kaplan y de David Petrocelli en UNLP.

Referencias

- [Abbers 10] Abbers, F., Backlund, A. ; Truscan, D., “MATERA - An Integrated Framework for Model-Based Testing”, Engineering of Computer Based Systems (ECBS), 17th IEEE International Conference 2010
- [Badri 09] Mourad Badri, Linda Badri and Maxime Bourque-Fortin, “Automated State-Based Unit Testing for Aspect Oriented Programs A Supporting Framework”, Journal of Object Technology. Vol. 8, No. 3, May/June 2009.
- [Blanco 06] Raquel Blanco, Eugenia Díaz, Javier Tuya,” Generación automática de casos de prueba mediante búsqueda dispersa ”, Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.2, No. 1, 2006.
- [Boghdady 12] P. N. Boghdady, Nagwa L. Badr, M. A. Hashim, Mohamed F. Tolba, “Test Cases Automatic Generator (TCAG): A Prototype”, First International Conference, AMLTA 2012, Cairo, Egypt, December 8-10, 2012.
- [Boghdady 11] Boghdady, P.N. Badr, N.L. ; Hashim, M.A. ; Tolba, M.F., “An enhanced test case generation technique based on activity diagrams”, Computer Engineering & Systems (ICCES), 2011
- [Correa 11] Natalia Correa, Roxana Giandini, “Casos de Prueba del Sistema Generados en el Contexto MDD/MDT”, LIFIA- Laboratorio de Investigación y Formación en Informática Avanzada, Universidad Nacional de La Plata, 40 JAIIO, ASSE 2011.
- [Correa 12] Natalia Correa, Roxana Giandini (2012). Casos de Prueba del Sistema Generados en el Contexto MDD/MDT. 41 JAIIO - ASSE 2012 - ISSN: 1850-2792 - Pág. 91-105
- [Doorn 02] Doorn J., Hadad G., Kaplan G. (2002) Comprendiendo el Universo de Discurso Futuro, WER’02 - Workshop on Requirements Engineering, Valencia, Spain.
- [Gutiérrez 06] Javier J. Gutiérrez, María J. Escalona, Manuel Mejías y Jesús Torres, Hacia una propuesta de pruebas tempranas del sistema, XV Jornadas de Ingeniería del Software y Bases de Datos. JISBD 2006.
- [Hadad 08] Hadad G.D.S., Doorn J.H., Kaplan G.N. Creating Software System Context Glossaries, In: Mehdi Khosrow-Pour (ed) Encyclopedia of Information Science and Technology. IGI Global, Information Science Reference, Hershey, PA, USA, ISBN: 978-1-60566-026-4, 2nd edn, Vol. II. 2008.
- [Hasling 08] Bill Hasling, Helmut Goetz, Klaus Beetz. Model Based Testing of

- System Requirements using UML Use Case Models. International Conference on Software Testing, Verification, and Validation. 2008
- [Javed 07] Javed, A.Z. “Automated Generation of Test Cases Using Model-Driven Architecture”, *Automation of Software Test*, 2007.
- [Kaplan 15] Gladys Kaplan, Jorge Doorn, Walter Panessi, Claudia Ortiz, Eugenia Cespedes, Julian Massolo, David Petrocelli, “Generación semi automática de casos de prueba a partir de escenarios”, WICC 2015.
- [Lamancha 09] Beatriz Pérez Lamancha, Macario Polo, “Generación automática de casos de prueba para líneas de producto de software”, *Revista Española de Innovación, Calidad e Ingeniería del Software*, Vol.5, No. 2, 2009.
- [Leite 97] Leite, J.C.S.P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., Oliveros, A., “Enhancing a Requirements Baseline with Scenarios”, *Requirements Engineering Journal*, Vol.2, N° 4, 1997.
- [Leite 90] Leite J.C.S.P., Franco, A.P.M., (1990) “O Uso de Hipertexto na Elicitação de Linguagens da Aplicação”, *Anais de IV Simpósio Brasileiro de Engenharia de Software*, SBC.
- [Leite 04] Leite, J.C.S.P., Doorn, J.H., “Perspectives on Software Requirements: An introduction” en el libro “Perspectives on Software Requirements”, Kluwer Academic Publishers, EEUU, ISBN: 1-4020-7625-8, Capítulo 1, 2004.
- [Linzhang 04] Wang Linzhang, Yuan Jiesong ; Yu Xiaofeng ; Hu Jun ; Li Xuandong; Zheng Guoliang, “Generating test cases from UML activity diagram based on Gray-box method”, *Software Engineering Conference*, 2004. 11th Asia-Pacific
- [Mingsong 09] Chen Mingsong, Qiu Xiaokang, Li Xuandong, “Automatic Test Case Generation for UML Activity Diagrams”, *Journal of Object Technology*. Vol. 8, No. 3, May/June 2009.
- [Odalys 06] Jordán Enriquez, Odalys; Vázquez Ruiz, Orelvis, “GENERACIÓN DE CASOS DE PRUEBA A PARTIR DE CASOS DE USO EN LAS PRUEBAS DE SOFTWARE”, *Ingeniería Industrial*, vol. XXVII, núm. 1, 2006, pp. 7-10 Instituto Superior Politécnico José Antonio Echeverría La Habana, Cuba ISSN: 0258-5960, 2006.
- [Palacio 09] Liliana González Palacio, “MÉTODO PARA GENERAR CASOS DE PRUEBA FUNCIONAL EN EL DESARROLLO DE SOFTWARE”, *Revista Ingenierías Universidad de Medellín*, vol. 8, No. 15 especial, pp. 29-36 - ISSN 1692-3324, 150 p. Medellín, Colombia, 2009
- [Riebisch 03] M. Riebisch, I. Philippow, and M. Götze, “UML-Based Statistical Test Case Generation,” in *Revised Papers from the International Conference NetObjectDays on Objects, Components, Architectures, Services, and Applications for a Networked World*, 2003, pp. 394-411.
- [Salem 11] Yasmine Ibrahim Salem y Riham Hassan. 2011. Requirement-Based Test Case Generation and Prioritization. 978-1- 61284- 185-4/111 - 2011 IEEE.
- [Wendland 11] Marc-Florian Wendland, Ina Schieferdecker and Alain Vouffo-Feudjio..Requirements-driven testing with behavior trees. Fourth International Conference on Software Testing, Verification and Validation Workshops. 2011.