

VALIDACIÓN DE ESCENARIOS FUTUROS CON PROTOTIPOS

Gladys N. Kaplan^(1,2)
gladyskaplan@speedy.com.ar

Jorge H. Doorn^(1,3)
jdoorn@exa.unicen.edu.ar

Graciela D. S. Hadad^(1,2)
ghadad@ub.edu.ar

⁽¹⁾Facultad Regional Buenos Aires, Universidad Tecnológica Nacional, Buenos Aires, Argentina

⁽²⁾LINTI, Fac. de Informática, Universidad Nacional de La Plata, La Plata, Buenos Aires, Argentina

⁽³⁾INTIA, Fac. de Ciencias Exactas, Univ. Nac. del Centro de la Pcia.de Bs.As., Tandil, Bs.As., Argentina

INTRODUCCIÓN

La comprensión de la documentación, mensajes, preguntas, comunicación, etc. que recibe el cliente/usuario se facilita si esta presentada en lenguaje natural en vez de utilizar esquemas técnicos. Más aun si el lenguaje natural utilizado es conocido por ellos con precisión y se minimiza el riesgo de ambigüedad, siempre presente cuando se trabaja utilizando lenguaje natural. Esta estrategia ha sido probada en numerosos casos de estudio y describe en lenguaje natural las situaciones del presente y aquellas en las que el nuevo software a construir se desempeñará. La utilización del lenguaje natural presenta una ventaja relevante: "la facilidad y seguridad con que los modelos pueden ser validados con el cliente/usuario".

La estrategia comienza construyendo un glosario, luego comprendiendo el universo de discurso (UdeD) del presente y pasando a comprender el universo de discurso para el futuro.

Los modelos utilizados son el Léxico Extendido del Lenguaje (**LEL**) [Leite 89] y los **escenarios** [Carrol 95] [Rubin 92] [Potts 94] [Jacobson 92]. El LEL tiene como propósito exclusivo conocer la semántica del vocabulario de la aplicación, dejando para una etapa posterior la comprensión del problema [Franco 92]. En este paso se vence el primer obstáculo entre el cliente/usuario y el equipo de desarrollo: el vocabulario. Los escenarios se construyen anclando todas sus descripciones en el LEL ya desarrollado. Los escenarios se utilizan para describir los procesos del negocio actuales, sean estos manuales o automatizados, (escenarios actuales) y en una segunda etapa la funcionalidad del sistema futuro (escenarios futuros).

El modelo de escenarios [Leite 97], es una estructura compuesta de **título**, **objetivo**, **contexto**, **recursos**, **actores**, **episodios** y **excepciones**. El título, el objetivo, el contexto, los recursos, los actores y las excepciones son sentencias declarativas, mientras que los episodios son un conjunto de sentencias expresadas en un lenguaje simple, que muestran descripciones operacionales de comportamiento.

Podemos afirmar en base a un número considerable de trabajos anteriores, propios y de muchos otros autores, que el modelo de escenarios es eficaz para representar el UdeD actual. Pero se han observado inconvenientes reiterados en el proceso de validación de los escenarios futuros. Es aquí donde se genera el presente proyecto de investigación basado en la hipótesis que al complementar los escenarios con prototipos para la actividad de validación se minimizan estos inconvenientes.

Nuestra propuesta se basa en la generación semiautomática de prototipos basado en Escenarios futuros completos y verificados, con el objetivo de Validar y Negociar los requisitos del sistema y capturar requisitos tardíos.

PROTOTIPOS

Los prototipos han sido utilizado de diferentes maneras. Algunos autores proponen aplicar herramientas RAD (Rapid Application Development)) (Kerr 94) que proponen la generación de prototipos muy tempranamente. Pasando al diseño con un conjunto mínimo de requisitos. Otros, como [Sommerville 02] proponen la validación y elicitación de requisitos utilizando prototipos desechables o prototipos evolutivos, que luego se utilicen en el desarrollo del producto final. Otros autores [Elkoutbi 99] se han centrado en la generación de prototipos para validar escenarios, dentro del marco de los “use case” (específicamente en el ambiente de UML), lo que supone un cierto grado de diseño de la aplicación. [Elkoutbi 99] propone la generación del prototipos después del análisis parcial e integración de las especificaciones obtenidas de los escenarios. [Logrippo 99] usa los prototipos después de generar, desde los escenarios, “Use Case Map” (UCMs) que es utilizado para capturar e integrar requisitos.

En el presente proyecto se propone utilizar los prototipos como la representación visual de los Escenarios Futuros. Estas representaciones no tienen ningún tipo de cosmética y no están orientadas a precisar la interfase del sistema, solo representan la interacciones del sistema con el UdeD que lo rodea. Toda acción que llevará a cabo el artefacto de software es mostrado mediante mensajes. Cada interacción con el sistema contiene una pantalla anexa para ingresar texto libre con el fin de registrar los comentarios realizados por el cliente/usuario. La información es guardada en un base de datos y queda asociada al escenario futuro correspondiente. La información registrada es luego utilizada para actualizar los escenarios futuros.

PROCESO DE OBTENCIÓN DE REQUISITOS

Como se muestra en la Figura 1 el proceso consta de tres etapas: comprender el Universo de Discurso Actual (UdeDa) luego, comprender el Universo de Discurso Futuro (UdeDf) y finalmente especificar los requisitos del software. Se dispone además de un repositorio con muy bajo nivel de estructuración en el que se toma nota de los requisitos candidatos espontáneos. Este agregado es muy eficaz ya que el Ingeniero de Requisitos suele elicitar requisitos en etapas

tempranas del desarrollo de los modelos. Ni el LEL ni los escenarios son aptos para registrar este conocimiento el que de ninguna manera debe ser ignorado o postergado.

PRIMERA ETAPA: COMPRENSIÓN DEL UNIVERSO DEL DISCURSO ACTUAL

La primer etapa comienza con la construcción del **Léxico Extendido del Lenguaje**. Durante la descripción de los escenarios el LEL podrá sufrir variaciones menores en cuanto a modificar alguna descripción, agregar algún símbolo y encontrar sinónimos. Paralelamente a la construcción del LEL el Ingeniero de Requisitos se enfrenta con información que no pertenece a la etapa en estudio, pero que no puede perder. Esta información se denomina **Requisitos Candidatos Espontáneos**. En los modelos donde es preciso comprender el problema antes de pensar en la solución, se torna habitual que el cliente/usuario mencione servicios para el nuevo software cuando dicha información aun no es requerida, específicamente durante la etapa de conocimiento del problema. Esta “información adelantada” menciona actividades de las que no se conocen si serán manuales o automáticas, ni quien las realizará; también suele hablar de la seguridad relacionada con los procesos, recursos y los permisos de acceso al nuevo software que se incorporará en la solución; menciona problemas a cerca de la escasez o necesidad de nuevo recursos; etc. Esta información el usuario las comenta a veces con relevancia y otras casi al pasar. Es alta la probabilidad de que el cliente/usuario no vuelva a mencionarlo ya que puede pensar que si ya lo dijo no es necesario volverlo a hacer. O que haya sido una idea que hasta el mismo cliente/usuario desconoce su importancia, y el Ingeniero de Requisitos no sabe aún cómo impactará en el sistema. Con el fin de no perder dicha información se completa una ficha de Requisitos Candidatos Espontáneos que será utilizada en etapas posteriores.

A partir del LEL se **derivan los escenarios actuales**, obteniéndose una primera versión de los mismos. Luego se completa el proceso de descripción para finalmente **Integrar los Escenarios Actuales** y obtener la visión global del problema en estudio.

SEGUNDA ETAPA: COMPRENSIÓN DEL UNIVERSO DEL DISCURSO FUTURO

Completados el LEL y los Escenarios Actuales, el Ingeniero de Requisitos ha adquirido y modelado el conocimiento suficiente para pasar a la siguiente etapa. En este contexto comienza la comprensión del UdeD futuro con el fin de entender y registrar lo que el cliente/usuario necesita y desea para el nuevo sistema. Basándose en los objetivos del sistema y en el conocimiento adquirido, se construyen los **Escenarios Futuros**. Estos escenarios incorporan naturalmente muchos de los requisitos del software en sus descripciones, siendo la mayor parte de los mismos requisitos funcionales. Los escenarios futuros son el resultado de la síntesis de los objetivos del sistema y de los procesos del negocio actual. En otras palabras, se debe definir como el objetivo del sistema afecta el proceso del negocio y en consecuencia cuáles serán las

A partir de los escenarios futuros se construyen los **Escenarios Futuros Integradores**. La elicitación de requisitos y la negociación de los mismos se completa con la prueba del **Prototipo**, obteniéndose además requisitos tardíos, aquellos que comúnmente son detectados recién en la implementación del software. Con esta información se modifican los escenarios futuros que serán la fuente de información para la siguiente etapa.

TERCERA ETAPA: ESPECIFICACIÓN DE REQUISITOS

Con los escenarios futuros terminados comienza el análisis de los requisitos y su posterior especificación. Esta etapa será estudiada en proyectos futuros.

GRADO DE AVANCE

La primera etapa del proceso descrito más arriba se ha investigado exhaustivamente, mientras que la segunda se encuentra en su fase final, constituyendo este proyecto uno de los aspectos pendientes.

REFERENCIAS

- [Leite 89] "Application Languages: A Product of Requirements Analysis", Leite, J.C.S.P., Departamento de Informática, PUC-/RJ, 1989.
- [Carrol 95] "Scenario Based Design: Envisioning Work and Technology", Carrol, J. (ed.), System Development, Wiley, New York, 1995.
- [Potts 94] "Inquiry-Based Requirements Analysis", Potts, C., Takahashi, K., Antón, A.I., IEEE Software, Vol. 11, n.2, Mar. 1994.
- [Jacobson 92] "Object-Oriented Software Engineering - A Use Case Driven Approach"; Jacobson, Y., Christerson, M., Jonsson, P., Overgaard, G., Reading, MA: Addison Wesley, New York: ACM Press, 1992.
- [Kerr 94] "Inside RAD"; Kerr J; Hunter R. McGraw-Hill 1994.
- [Sommerville 02] "Ingeniería de Software"; Sommerville, I; Pearson Educación 2002
- [Elkoutbi 99] "Generation User Interface Prototypes from Scenarios", Elkoutbi M, Khriiss I, Keller R. IEEE Software, 1999.
- [Logrippo 99] "Use Case Maps for the Capture and Validation of Distributed Systems Requirements", Logrippo D, Buhr R, Gray T . IEEE Software, 1999.
- [Rubin 92] "Object Behavior Analysis", Rubin, K.S., Goldberg, J.. Communications of the ACM, Vol. 35, N° 9, Sep. 1992.
- [Lelite 97] "Enhancing a Requirements Baseline with Scenarios", Leite, J.C.S.P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., Oliveros, A., Requirements Engineering Journal, Vol.2, No. 4, 1997, pp 184-198.
- [Franco 92] "Uma estratégia de Suporte a Engenharia de Requisitos", Franco, A.P.M., Leite, J.C.S.P., Anais do XIX Seminario Integrado de Software y Hardware, Rio de Janeiro, 1992, pp.200-213.