

Uso de Escenarios en la Derivación de Software

Tesis Doctoral

Graciela Dora Susana Hadad

ghadad@unlam.edu.ar

Director: Prof. Dr. Julio Cesar Sampaio do Prado Leite

Co-director: Prof. Ing. Jorge Horacio Doorn

Universidad Nacional de La Plata, Facultad de Ciencias Exactas

14 de Octubre del 2008

Resumen

Esta tesis está desarrollada en el marco de una estrategia en la Ingeniería de Requisitos, denominada SDRES (Scenario Driven Requirements Engineering Strategy), que intenta abordar temas poco tratados en la práctica real, tales como los cambios constantes en los requisitos, defectos del software originados en los requisitos, el contexto organizacional que rodea al sistema de software y la consideración de requisitos de calidad.

Esta estrategia está dirigida por modelos (Léxico Extendido del Lenguaje, Escenarios y Documento de Requisitos). Además está orientada al cliente, por lo que utiliza modelos construidos en lenguaje natural como medio de comunicación y elicitación. SDRES tiene en cuenta la calidad de los modelos que produce mediante procesos de verificación y validación. Para cada actividad de la estrategia se presenta un conjunto de heurísticas y recomendaciones. Se encara el tema de evolución y versionado de los modelos, así como distintas modalidades de aplicación de la estrategia según la complejidad del problema, el conocimiento sobre el mismo y otras características del dominio del problema y del proyecto. La contribución de la tesis a la estrategia consistió en su afianzamiento como proceso, detallando e incorporando nuevas actividades, heurísticas, modelos, guías de aplicación, y profundizando lo existente.

1. Motivación

Diversas estadísticas realizadas a lo largo de las últimas décadas informan sobre fracasos en proyectos de software, tales como los reportes CHAOS del Standish Group y los reportes GAO de US General Accounting Office, o artículos como [Alford 79] [Boehm 81] [Faulk 92] [Lutz 93] [Lindstrom 93] [Gibbs 94] [Stix 94] [Finkelstein 96]. En ellos se concluye que los requisitos son la fuente principal de problemas en la mayoría de dichos proyectos: requisitos inadecuados, cambios en los requisitos durante el ciclo de vida, requisitos no bien comprendidos y requisitos incompletos.

Por otro lado, la Ingeniería de Software ha estado prestando más atención al modelado e implementación (codificación) de productos, proveyendo métodos, herramientas y procedimientos para tal fin, que a examinar aspectos relacionados con la definición de requisitos y su intrínseca dependencia de los aspectos sociales que rodean el desarrollo del software y que rodearán su operación. Por ende, la Ingeniería de Requisitos debe atender aspectos no técnicos: i) establecer una buena comunicación entre todos los involucrados, y ii) comprender cabalmente el mundo del usuario.

Existen además otras dificultades que hacen que el desarrollo de software no sea una actividad lineal y sencilla, que alcance con sólo concentrarse inicialmente en la definición de requisitos para luego dedicarse a diseñar e implementar la solución. Uno de esos obstáculos en el desarrollo del software es la evolución de los requisitos. Cambios continuos en el software están motivados por un

gran espectro de causas que frecuentemente deben atenderse en simultáneo y casi inmediatamente a su aparición. La naturaleza de estos cambios en el software está altamente relacionada con cambios en los requisitos. Estadísticas realizadas marcan que el 50% o más de los requisitos van a cambiar antes que el sistema de software se ponga en operación [Kotonya 98]. En base a estos resultados, debe cuestionarse cuál es el origen de estos cambios en el software. Muchas veces estos se asocian a cambios en el universo de discurso. Pero es frecuente que algunos requisitos se descubran parcialmente o no salgan a la luz hasta avanzado el desarrollo, cuando los diseñadores o incluso los programadores deben realizar tareas a muy bajo nivel de detalle y se chocan con ambigüedades, contradicciones u omisiones. En estos casos, la esencia del cambio no corresponde a un mundo en evolución sino a requisitos no descubiertos oportunamente. Con respecto a los cambios “reales” en el universo de discurso, no sólo se deben a cambios externos o internos a la organización como consecuencia de factores políticos, económicos, legales, sociales o tecnológicos, sino muchas veces a un cambio en las expectativas de los mismos clientes y usuarios una vez que ven más claramente las posibilidades que permitirá el sistema de software. Lehman [Lehman 80] va más allá expresando que un sistema una vez instalado se convierte en parte del universo de discurso alterándolo y, por ende, él mismo (el software) altera sus propios requisitos.

En general, las estadísticas realizadas sobre la evolución de los requisitos no discriminan adecuadamente el origen de los cambios en el desarrollo del software, y simplemente lo adjudican a la evolución del universo de discurso [Yeh 90] [Kotonya 96] [Leite 97] [Madhavji 97] [Goedicke 99]. Lientz & Swanson [Lientz 78] determinaron que los cambios en el software durante su mantenimiento se debían en un 80% a cambios en los requisitos y el restante 20% a correcciones de errores. Sin embargo, no se discrimina en estos valores cuántas de las correcciones de errores se debían a requisitos con defectos ni cuántos de los cambios en los requisitos eran por la naturaleza evolutiva del universo de discurso o por defectos en los requisitos (requisitos descubiertos tardíamente o requisitos mal interpretados). Un estudio más reciente sobre mantenimiento de software [Pigoski 96] informa que alrededor del 55% de los pedidos de cambio corresponden a requisitos nuevos o cambios en requisitos por mejoras, mientras que un 25% son cambios técnicos por adaptaciones.

En resumen, la visión muy difundida de la evolución de los requisitos se basa habitualmente en un único componente: el cambio del universo de discurso, pero prácticamente omite un segundo componente: los requisitos descubiertos o comprendidos tardíamente. Es obvio que este último componente produce también modificaciones en los requisitos, y se sospecha, en base a la experiencia acumulada, que lo hace con mucha mayor frecuencia de lo mencionado en la literatura. Ya en 1994 Michael Jackson afirmaba que sobre los requisitos se saben dos cosas: i) van a cambiar y ii) van a ser mal comprendidos [Jackson 94]. Aún hoy, no hay evidencia sobre si los avances en la Ingeniería de Requisitos, a través de nuevas técnicas, métodos y herramientas propuestos e incorporados al proceso, han dado una solución o han podido mitigar este aspecto del problema.

Luego, los puntos clave que un proceso de requisitos debiera tener en cuenta son:

- i. Involucrar a los clientes y usuarios en el proceso de desarrollo. Esto requiere establecer la participación activa y constante de los clientes y usuarios. Una buena comunicación es esencial para comprender adecuadamente sus necesidades.
- ii. Asegurar la calidad de los requisitos. El uso de técnicas adecuadas de verificación y validación debe formar parte de las actividades continuas del proceso de requisitos. Además, la calidad de los requisitos debe conservarse a lo largo de todo el ciclo de vida del software.
- iii. Tratar la evolución de los requisitos como una situación persistente a lo largo del ciclo de vida del software. El proceso de desarrollo debe reconocer esta evolución y manejarla. Por lo tanto

debe implementarse una adecuada gestión de requisitos, asegurando la rastreabilidad de los mismos para facilitar la realización de los cambios en los modelos y en el software, evitando la introducción de nuevos defectos a consecuencia de los cambios.

- iv. Considerar aspectos sociales, organizacionales, políticos y legales del ambiente donde el software operará. Esto impone una inmersión de los ingenieros de software en dicho ambiente y el uso de técnicas de modelado organizacional y de técnicas de recolección de datos apropiadas al problema bajo estudio.
- v. Procurar la completitud de los requisitos. Esto es, por un lado, un aspecto relacionado con la calidad y, por otro lado, requiere una comprensión acabada del universo de discurso, ambos aspectos tratados en los puntos anteriores.

2. Objetivo de la tesis

Se propone una estrategia para la definición de requisitos orientada al cliente y dirigida por modelos: Léxico Extendido del Lenguaje (LEL), Escenarios y Documento de Definición de Requisitos, que cubre todos los aspectos antes mencionados.

Los modelos propuestos se caracterizan por tener una representación textual basada en lenguaje natural. El registro de los requisitos de sistemas informáticos mediante el uso del lenguaje natural es sumamente atractivo ya que permite involucrar a la totalidad de los agentes intervinientes en el proceso de desarrollo sin las barreras de conocimiento clásicas que se producen cuando se utilizan otras formas de representación. Se ha comprobado que el uso de una estructura ligeramente formalizada del lenguaje natural, como es el uso de los Escenarios y del Léxico Extendido del Lenguaje, no introduce una dificultad apreciable cuando es leída por personas no especializadas en el desarrollo de aplicaciones de software.

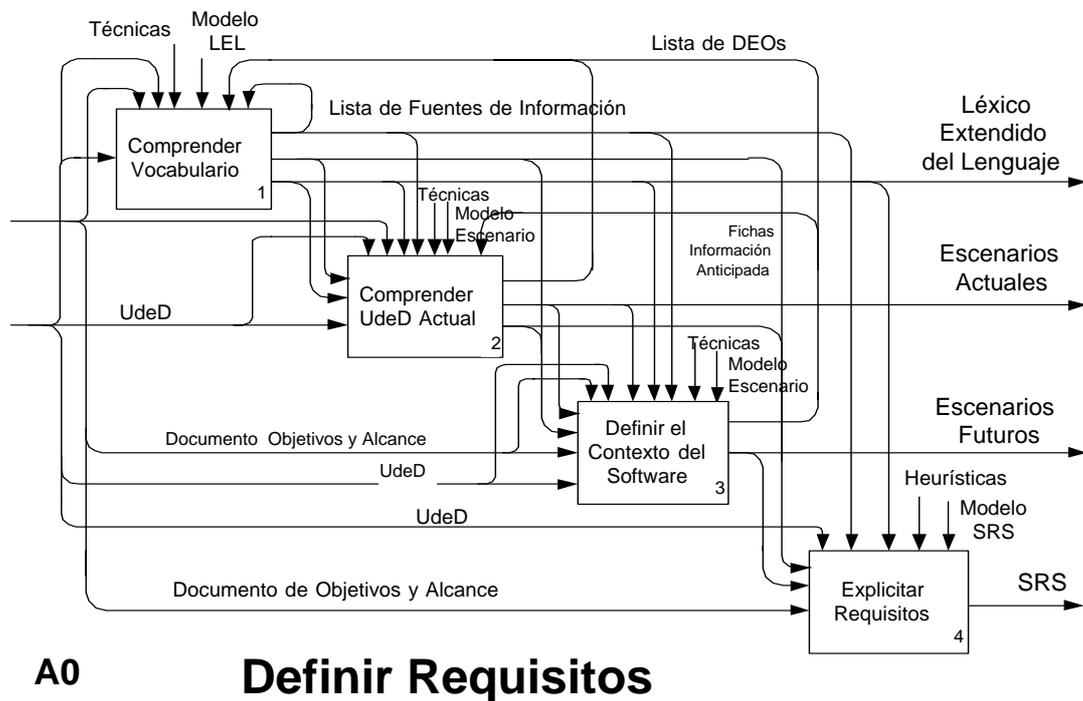
El modelo Léxico Extendido del Lenguaje permite comprender el vocabulario de la aplicación, mediante una descripción de la denotación y connotación de cada palabra / frase utilizada en el universo de discurso. El modelo de Escenarios tiene como objetivo comprender el problema, mediante la descripción de situaciones del dominio de la aplicación. Cabe destacar que, Escenarios es un modelo de representación ampliamente difundido en la literatura [Potts 95], [Carroll 95], [Sutcliffe 98], [Rolland 98], [Rolland 98b] y utilizado exitosamente en la práctica [Weidenhaupt 98]. El Documento de Definición de Requisitos, conocido por sus siglas en inglés SRS (Software Requirements Specification), permite individualizar los requisitos en forma precisa. Esta individualización facilita la rastreabilidad de los mismos, el control de sus cambios, el análisis de su interdependencia, el apareamiento de los requisitos con los componentes de software, el seguimiento de la evolución de los requisitos y el versionado de los mismos, entre otras actividades.

SDRES no se limita a presentar un conjunto de modelos a construir y cómo y cuándo construirlos, sino que, por el contrario, lo que aporta mayor valor es que brinda pautas claras de qué, cómo y cuándo elicitar, y de qué, cómo y cuándo analizar (verificar, validar, negociar). La estrategia apunta a obtener una mayor comprensión del problema de manera tal de crear mejores soluciones negociadas. Establece un puente sólido entre las necesidades de los clientes y las soluciones de los diseñadores, proveyendo un correcto entendimiento del dominio y un ancla en el vocabulario del dominio.

3. Características de la estrategia

SDRES involucra todas las actividades de la ingeniería de requisitos: elicitación, modelado, análisis y gestión, y consiste básicamente en (ver Figura 1):

- i. Comprender el vocabulario usado en el universo de discurso, ayudado por el Léxico Extendido del Lenguaje;
- ii. Comprender el universo de discurso actual, ayudado por un conjunto de escenarios actuales, que representa las situaciones observadas en el dominio de la aplicación;
- iii. Definir el contexto del sistema de software, produciendo un conjunto de escenarios futuros, que representa las situaciones imaginadas en el futuro dominio de la aplicación cuando esté operando el nuevo sistema de software; y
- iv. Explicitar los requisitos, produciendo un SRS, a partir del conjunto de escenarios futuros, donde los requisitos están claramente individualizados y donde se le asignan prioridades.



A0 Definir Requisitos

LEL: Léxico Extendido del Lenguaje, UdeD: Universo de Discurso, DEO: Discrepancias, Errores y Omisiones, SRS: Documento de Definición de Requisitos

Figura 1. SADT de la estrategia en la ingeniería de requisitos

En la Figura 1 se muestra mediante el modelo SADT las cuatro etapas en las cuales se desarrolla la estrategia y los productos obtenidos en cada una. En esta figura se han incorporado los reciclos que ocurren por mejoras en la comprensión del universo de discurso, donde se vuelve a refinar el LEL, los Escenarios Actuales y / o los Escenarios Futuros. En la cuarta etapa no hay un feedback a etapas anteriores, porque es una etapa que trata sobre reorganizar la información ya elicitada y modelada en la etapa anterior.

Además, se visualiza en la Figura 1 un producto intermedio de las tres primeras etapas, denominado Fichas de Información Anticipada, donde se vuelca todo lo que se elicita y es importante pero no corresponde al modelo que se está construyendo en dicha etapa sino que es información útil para alguna etapa posterior. Este modelo auxiliar evita la pérdida de información capturada en forma adelantada al modelo receptor de la misma.

Esta estrategia, aunque expuesta siguiendo un modelo en cascada, es adaptable a diversos procesos

de software, y es utilizable en una amplia variedad de problemas. En la tesis se presentan pautas de cómo adaptar la estrategia a distintos proyectos y organizaciones, y a distintos modelos de proceso de software.

4. Aportes de la tesis

En la Figura 1 se muestran los modelos utilizados por la estrategia (Léxico Extendido del Lenguaje, Escenarios Actuales, Escenarios Futuros, Lista de Requisitos y Fichas de Información Anticipada), y se señalan los aportes de esta tesis a la estrategia (los tonos grises más fuertes implican mayor aporte a la misma), y se indican los desprendimientos de otras tesis y líneas de investigación a partir de uno o varios modelos de la estrategia.

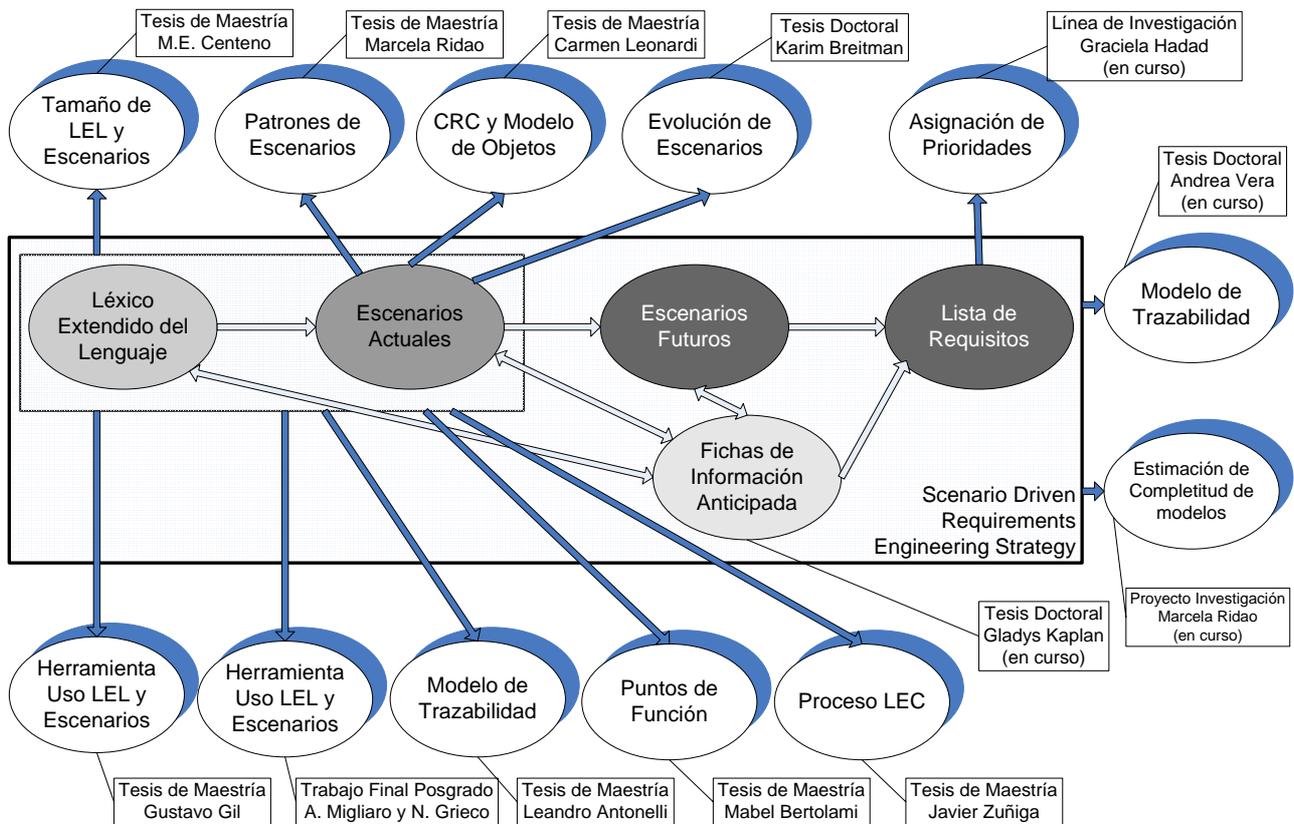


Figura 1. Aportes de la tesis y de SDRES

4.1. Aportes de la tesis a la estrategia

En esta tesis se han refinado y ampliado las heurísticas para la construcción del LEL, presentando nuevos problemas y soluciones, tales como el tratamiento de homónimos, las relaciones de sinonimia e hiperonimia (tipo y super-tipo), uso del género en la noción, consideración del mecanismo de creación de los términos y la evolución del LEL. Se ha enfatizado la búsqueda de fuentes de información y la perspectiva de la vigencia de información que ellas trasuntan.

Respecto a los escenarios, se ha actualizado el modelo de escenarios en base a la experiencia obtenida en la variedad y cantidad de conjuntos de escenarios construidos, tanto en casos de estudio como en casos reales. Se han refinado las heurísticas para la construcción de escenarios actuales, principalmente la aplicación de operaciones y la creación de escenarios integradores. También se ha

mejorado la heurística de descripción y se ha dado una guía para el uso de roles en caso de actores alternativos en un escenario. Se han comparado los posibles enfoques top-down y bottom-down frente a la modalidad middle-out presentada en la tesis para la construcción de escenarios actuales.

Se ha comprobado a través de su uso en varios casos de estudio que las inspecciones son una técnica de verificación de gran efectividad aplicada a documentos de requisitos.

Se han presentado heurísticas para la evolución de escenarios actuales a escenarios futuros bajo el marco de los objetivos del sistema de software. Se han comparado éstas con las propuestas de otros autores que manejan objetivos y escenarios. Se ha enfatizado en la distinción entre los objetivos de los procesos del negocio y los objetivos del sistema, considerando que los escenarios introducidos en esta tesis representan situaciones bajo un contexto organizacional. Se ha incentivado la elicitación de RNF que puedan asociarse al comportamiento que involucran los escenarios.

Se han expuesto heurísticas para poner en evidencia los requisitos del software, facilitando su trazabilidad y permitiendo la creación de un SRS.

Se ha descrito la evolución de los modelos, tipificando las causas de dicha evolución y cómo gestionarla. Se ha justificado el mantenimiento de los modelos que componen SDRES durante el proceso de desarrollo, y se ha tratado su versionado. Asimismo, se ha expuesto cómo aplicar SDRES y cómo adaptarlo según las características del problema propiamente dicho y de la organización cliente.

4.2. Aportes de la estrategia

SDRES brinda una solución práctica a los siguientes tópicos:

Comunicación

- ✓ Se crea un glosario con la terminología usada en el universo de discurso (Léxico Extendido del Lenguaje).
- ✓ Modelos basados en lenguaje natural y toda otra documentación generada en el proyecto puede escribirse usando la terminología usada en el universo de discurso.

Evolución

- ✓ Los modelos de Escenario ayudan a imaginar, estimar y realizar cambios en los requisitos.
- ✓ Un Documento de Definición de Requisitos es una herramienta simple para la gestión del proceso de desarrollo, pues provee una lista de control para verificar el avance del diseño y el avance de las pruebas del software.

Rastreabilidad

- ✓ Los modelos con vínculos hipertextuales facilitan la pre y post navegación.
- ✓ Un modelo de rastreabilidad soportado por una pila de trazas vincula eficientemente las necesidades del universo de discurso actual con el sistema de software.

Asegurar la completitud en estrategias semi-formales es extremadamente difícil. Esto también se cumple en la estrategia propuesta. Sin embargo, se han realizado esfuerzos significativos para ser exhaustivo en minimizar esta desventaja a lo largo del proceso completo de ingeniería de requisitos propuesto. Los puntos clave sobre esto son:

- i. El Léxico Extendido del Lenguaje es consistentemente mejorado e incrementado a lo largo del

proceso.

- ii. El proceso de integración de escenarios que genera escenarios de nivel más alto puede sugerir la existencia de brechas semánticas como consecuencia de omisiones. Esta operación se aplica tanto al conjunto de escenarios actuales como al conjunto de escenarios futuros.
- iii. La construcción de escenarios futuros combinando los objetivos del sistema de software con escenarios actuales amplía el espectro de la elicitación.
- iv. Las actividades de verificación y validación de los modelos de requisitos producidos indican un enfoque orientado a la calidad.

Registrar los requisitos del software mediante estructuras en lenguaje natural es un enfoque atractivo dado que facilita la participación de todos los involucrados en el proceso de desarrollo, evitando las clásicas barreras de comunicación que se levantan cuando se usan otros esquemas de representación. Se ha observado que el uso de estructuras en lenguaje natural ligeramente formalizadas, tales como el Léxico Extendido del Lenguaje y Escenarios, no introduce grandes dificultades al ser leídas por gente no especializada en el desarrollo de software.

Los Escenarios no solo permiten modelar las características del sistema de software actual y del futuro sino también el comportamiento actual y futuro del universo de discurso. Los Escenarios Futuros permiten analizar las posibles influencias del sistema de software sobre el propio contexto esperado.

La propuesta de esta tesis para la construcción de escenarios actuales, una estrategia middle-out, intenta ser una contribución a la diversidad de propuestas existentes. Esta creencia se basa en la experiencia previa de haber utilizado una estrategia top-down. La diferenciación entre escenarios actuales y escenarios futuros clarifica el proceso de la ingeniería de requisitos. En tal sentido, muchas estrategias hablan sólo de los “to-be escenarios” (término asociado a escenarios futuros) dejando de lado el conocimiento actual del universo de discurso. Esto induce a crear escenarios futuros menos anclados en la cultura del universo de discurso que lo que se logra creando previamente los escenarios actuales.

Se debe enfatizar que la investigación en la Ingeniería de Requisitos enfrenta un problema de experimentación significativa [Ravid 00] [Kaindl 02], ya que se están desarrollando propuestas metodológicas para sistemas medianos, medianos-grandes y grandes pero los casos de estudio se realizan con problemas a veces visualizados como “problemas de juguete”. Esto trae la grave consecuencia que el observador casual percibe en la propuesta una inmensa burocracia frente a un problema que definitivamente no lo justifica. Cabe mencionar que la estrategia presentada fue aplicada no sólo a casos de estudio de diverso tamaño y en distintas universidades, sino también en casos reales por diversos profesionales.

La autora de la tesis ha sido revisora de aproximadamente 150 casos de estudio siguiendo esta estrategia, realizados por alumnos de grado avanzados en las universidades UTN y UNLaM, aplicados en casos reales en distintas organizaciones argentinas. La estrategia ha sido promovida en cursos de postgrado asistidos por profesionales del país y extranjeros.

5. Trabajos Futuros

Se enuncia a continuación una serie de temas expuestos en esta tesis que se abren para investigaciones futuras.

- i. **Jerarquías en el LEL:** esta tesis sólo intenta señalar estas relaciones descubiertas después de la creación de varios léxicos, dejando para una investigación posterior un análisis consciente y un tratamiento apropiado, especificando un proceso de detección. Está en curso una línea de investigación que incluye este tema, dirigida por Gladys Kaplan en UNLaM.
- ii. **Modelo de trazas:** la tesis presentó un bosquejo de modelo de trazas, pero no se ha ahondado en el tema sobre el tratamiento de las trazas a modelos externos a la línea base de requisitos. El modelo de trazas es preliminar y deberá aplicarse en varios casos de estudio de diferente nivel de complejidad para poder obtener un modelo refinado que abarque todos los posibles rastreos de requisitos. Está en curso una línea de investigación sobre este tema, dirigida por Jorge Doorn en UNLaM, acompañando la realización de la tesis doctoral de Andrea Vera en UNLP.
- iii. **Diseño de arquitectura:** otro tema abierto se refiere a la relación entre los escenarios futuros y el diseño de la arquitectura. Si se parte de un conjunto pre-existente de posibles arquitecturas, los escenarios futuros pueden estar excesivamente influenciados por esta condición, considerando entonces un proceso orientado al reuso. Por otro lado, si el software es suficientemente ingenuo, entonces es muy probable que los escenarios futuros sean el punto de partida donde tener en cuenta el diseño.
- iv. **Negociación:** es también un tema de futura investigación cómo los escenarios futuros pueden impactar las negociaciones entre los ingenieros de requisitos, los diseñadores y los clientes. Esto es particularmente verdadero en el caso de sistemas que dependen de interfaces orientadas al usuario.
- v. **Elicitación:** un tema a ampliar es considerar un modelo de fuentes de información, y cómo detectar eficientemente los usuarios claves y sus relaciones sociales en la organización. Está en curso un proyecto de investigación sobre Análisis de Redes Sociales y su aporte a la Ingeniería de Requisitos, dirigida por Jorge Doorn en colaboración entre UNLaM y UNCPBA.
- vi. **Validación:** es un problema abierto en el que podrían surgir nuevas contribuciones que mejoren la actividad, principalmente en la validación de escenarios futuros que requieren un esfuerzo extra por parte de los usuarios al tratar sobre un contexto no existente aún. Está en curso una línea de investigación sobre este tema dirigida por Gladys Kaplan en UNLaM, acompañada por la tesis doctoral de Gladys Kaplan en UNLP y la tesis de maestría de Renata Guatelli en UNLaM.
- vii. **Asignación de prioridades:** existen muchas técnicas de asignación de prioridades a requisitos, pero, en general están aisladas del proceso que dio origen a los mismos, por lo tanto presentan algunos inconvenientes, tales como el manejo de inconsistencias en los requisitos, la compatibilización de prioridades entre distintos usuarios, entre otros. Las técnicas existentes no dan pautas claras de en qué momento o a qué artefactos durante el proceso de definición de requisitos aplicarlas.
Está en curso una línea de investigación que incluye este tema, dirigida por la autora de esta tesis en UNLaM.

6. Publicaciones que acompañaron la tesis

Publicaciones Internacionales en Capítulo de Libro con referato

- “Defining System Context using Scenarios”, Leite, J.C.S.P., Doorn, J.H., Kaplan, G.N., Hadad,

G.D.S., Ridao, M.N., en el libro “Perspectives on Software Requirements”, editor Kluwer Academic Publishers, Estados Unidos, ISBN: 1-4020-7625-8, capítulo 8, pp.169-199, 2004.

- “Creating Software System Context Glossaries”, Hadad, G.D.S., Doorn, J.H., Kaplan, G.N., en Encyclopedia of Information Science and Technology. Editorial: IGI Global, Mehdi Khosrow-Pour (ed), Information Science Reference, Hershey, PA, EEUU, ISBN: 978-1-60566-026-4, 2º edición, Vol. II, pp. 789-794, 2008 (4500 p.).
- “Handling Extemporaneous Information in Requirements Engineering”, Kaplan, G.N., Doorn, J.H., Hadad, G.D.S., en Encyclopedia of Information Science and Technology. Editorial: IGI Global, Mehdi Khosrow-Pour (ed), Information Science Reference, Hershey, PA, EEUU, ISBN: 978-1-60566-026-4, 2º edición, Vol. IV, pp.1718-1722, 2008 (4500 p.).

Publicaciones Internacionales en Revistas con referato

- “Integración de escenarios con el léxico extendido del lenguaje en la elicitación de requerimientos: aplicación a un caso real”, Hadad, G., Kaplan, G., Oliveros, A., Leite, J.C.S.P., Revista de Informática Teórica y Aplicada (RITA), 6(1):77-103, Instituto de Informática da Universidade Federal do Rio Grande do Sul, Brasil, 1999.
- “A Scenario Construction Process”, Leite, J.C.S.P., Hadad, G.D.S., Doorn, J.H., Kaplan, G.N., Requirements Engineering Journal, 5(1):38-61, Springer-Verlag London, Gran Bretaña, 2000.
- “Scenario Inspections”, Leite, J.C.S.P., Doorn, J.H., Hadad, G.D.S., Kaplan, G.N., Requirements Engineering Journal, 10(1):1-21, Springer-Verlag London, Gran Bretaña, 2005.

Publicaciones en Anales de Congresos con referato

- “Enfoque Middle-Out en la Construcción e Integración de Escenarios”, Hadad, G.D.S., Doorn, J.H., Kaplan, G.N., Leite, J.C.S.P., WER’99 - Workshop on Requirements Engineering, dentro de XXVIII JAIIO, Buenos Aires, Argentina, pp.79-94, Septiembre 1999.
- “Inspección del Léxico Extendido del Lenguaje”, Kaplan, G.N., Hadad, G.D.S., Doorn, J.H., Leite, J.C.S.P., WER’00 - Workshop on Requirements Engineering, Río de Janeiro, Brasil, pp.70-91, Julio 2000.
- “Comprendiendo el Universo de Discurso Futuro”, Doorn, J.H., Hadad, G.D.S., Kaplan, G.N., WER’02 - Workshop on Requirements Engineering, Valencia, España, pp.117-131, 2002.
- “Construcción de Escenarios Futuros”, Hadad, G.D.S., Doorn, J.H., Kaplan, G.N., WICC 2003, UNCPBA, Tandil, Argentina, Mayo 2003.
- “Una visión cognitiva de modelos de requisitos de software”, Kaplan, G.N., Doorn, J.H., Hadad, G.D.S., WICC 2008, ISBN: 978-950-863-101-5, Universidad Nacional de La Pampa, General Pico, La Pampa, pp.386-391, Mayo 2008.
- “Derivar casos de uso de un glosario”, Hadad, G.D.S., Migliaro, A., Grieco, N., CACIC 2008, ISBN: 978-987-24611-0-2, Universidad Nacional de Chilecito, La Rioja, pp.722-734, 2008.

7. Referencias

- [Alford 79] Alford, M., Lawson, J., “Software Requirements Engineering Methodology (Development)”, RADDC-TR-79-168, US Air Force Rome Air Development Center, 1979.
- [Boehm 81] Boehm, B.W., “Software Engineering Economics”, NJ: Prentice-Hall, 1981.
- [Carroll 95] Carroll, J., “Introduction: The Scenario Perspective on System Development”, en el libro Scenario-Based Design: Envisioning Work and Technology in System Development, J. Carroll (ed), John Wiley & Sons, Nueva York, 1995, pp.1-18.
- [Faulk 92] Faulk, S., et al., “The Core Method for Real-Time Requirements”, IEEE Software, 9(5): 22-33, Septiembre 1992.
- [Finkelstein 96] Finkelstein, A., Dowell, J., “A comedy of Errors: The London Ambulance Service

- Case Study”, IWSSD’96, 8th international Workshop on Software Specification and Design, IEEE Computer Society Press, Alemania, 1996.
- [Gibbs 94] Gibbs, W.W., “Trends in Computing: Software’s Chronic Crisis”, *Scientific American*, Septiembre 1994, pp.72-81.
- [Goedicke 99] Goedicke, M., Meyer, T., Taentzer, G., “ViewPoint-oriented Software Development by Distributed Graph Transformation: Towards a Basis for Living with Inconsistencies”, RE’99, Limerick-Irlanda, 1999, pp.92-99.
- [Jackson 94] Jackson, M., “The Role of Architecture in Requirements Engineering”, ICRE’94, IEEE Computer Society, Colorado Springs, Abril 1994, p.241.
- [Kaindl 02] Kaindl, H., Brinkkemper, S., Bubenko Jr., J.A., Farbey, B., Greenspan, S.J., Heitmeyer, C.L., Leite, J.C.S.P., Mead, N.R., Mylopoulos, J., Siddiqi, J., "Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda", *Requirements Engineering Journal*, Springer-Verlag London Ltd., 7(3):113-123, 2002,.
- [Kotonya 96] Kotonya, G., Sommerville, I., “Requirements Engineering with viewpoints”, en “Software Requirements Engineering”, Richard H. Thayer y Merlin Dorfman (eds), IEEE Computer Society Press, 2º edición, Los Alamitos, CA, 1997, pp. 150-163.
- [Kotonya 98] Kotonya, G., Sommerville, I., “Requirements Engineering: Process and Techniques”, John Wiley & Sons, 1998.
- [Lehman 80] Lehman, M.M., “Programs, Life Cycles, and Laws of Software Evolution”, *IEEE Special Issue on Software Engineering*, 68(9): 1060-1076, Septiembre 1980.
- [Lientz 78] Lientz, B.P., Swanson, E.B., “Characteristics of Application Software Maintenance”, *Communications of the ACM*, 21(6): 466-481, Junio 1978.
- [Lindstrom 93] Lindstrom, D.R., “Five ways to Destroy a Development Project”, *IEEE Software*, 10(5): 55-58, 1993.
- [Lutz 93] Lutz, R., “Analyzing Software Requirements Errors in Safety-Critical Embedded Systems”, RE’93, IEEE Computer Society Press, Los Alamitos, CA, 1993, pp.126-133.
- [Madhavji 97] Madhavji, N.H., “Impact of Environmental Evolution on Requirements Changes”, panel en RE’97, IEEE Computer Society Press, Enero 1997, pp.98-99.
- [Pigoski 96] Pigoski, T.M., “Practical Software Maintenance”, Wiley Computer Publishing, 1996.
- [Potts 95] Potts, C., “Using Schematic Scenarios to Understand User Needs”, DIS’95 - Symposium on Designing Interactive Systems: Processes, Practices and Techniques, ACM Press, University of Michigan, 1995, pp.247-256.
- [Ravid 00] Ravid, A., Berry, D. M., “A Method for Extracting and Stating Software Requirements that a User Interface Prototype Contains”, *Requirements Engineering Journal*, Springer-Verlag London Ltd., 5(4): 225-241, 2000.
- [Rolland 98] Rolland, C., Souveyet, C, Ben Achour, C., "Guiding Goal Modeling Using Scenarios", *IEEE TSE*, 24(12): 1055–1071, 1998.
- [Rolland 98b] Rolland, C., Ben Achour, C., Cauvet, C., Ralyté, J., Sutcliffe, A., Maiden, M., Jarke, et al., “A Proposal for a Scenario Classification Framework”, *Requirements Engineering Journal*, Springer-Verlag, 3(1):23-47, 1998.
- [Stix 94] Stix, G., “Trends in Air Transportation: Aging Airways”, *Scientific American*, Mayo 1994, pp.70-78.
- [Sutcliffe 98] Sutcliffe, A.G., Maiden, N.A.M., Minocha, S., Manuel, D., "Supporting Scenario-Based Requirements Engineering", *IEEE TSE*, 24(12): 1072-1088, 1998.
- [Weidenhaupt 98] Weidenhaupt, K., Pohl, K., Jarke, M., Haumer, P., “Scenarios in System Development: Current Practice”, *IEEE Software*, 1998, pp.34-45.
- [Yeh 90] Yeh, R.T., Ng, P.A., “Software Requirements – A Management Perspective”, *System and Software Requirements Engineering*, M. Dorfman y R.H. Thayer (eds.), IEEE Computer Society Press, 1990, pp.450-461.