

# Identificación temprana de características transversales en el lenguaje de la aplicación capturado con el Léxico Extendido del Lenguaje

Leandro Antonelli

Director: Gustavo Rossi

Co-director: Julio Cesar Sampaio do Prado Leite

Asesor técnico: Alejandro Oliveros

**Tesis presentada para la obtención el grado de Doctor en Ciencias Informáticas,  
Facultad de Informática, Universidad Nacional de La Plata**

**Fecha de exposición: 27 de Abril 2012**

## Abstract

En este trabajo de tesis se presenta una estrategia para identificar características transversales del software utilizando el Léxico Extendido del Lenguaje (LEL). Debido a que la construcción del LEL se realiza tempranamente en el proceso de desarrollo del software, esto redundará en beneficios al evitar el retrabajo que pudiera ocasionar la detección a mitad del desarrollo. El lenguaje que utiliza para representar el conocimiento de la aplicación (LEL) posee buena expresividad, pero por sobre todo, utiliza el lenguaje conversacional sin utilizar ningún tipo de formalismo, con lo cual puede ser utilizado por todas las personas que participan del desarrollo de software. En esta tesis se muestra tanto la aplicabilidad como la efectividad de la estrategia propuesta. La aplicabilidad se ilustra a través de 3 ejemplos del mundo real. Además, la presente tesis también describe un experimento que se llevó a cabo con el fin de mostrar la efectividad de la estrategia. Por último, la tesis describe una herramienta que permite asistir en la aplicación de la estrategia propuesta.

## 1 Introducción

El software tiene que enfrentarse a muchas características (*concerns*). Con el fin de hacer frente a esta complejidad, la ingeniería de software tradicional tiene como estrategia la descomposición, esto es la estrategia de *divide y vencerás*: dividir al problema en piezas más pequeñas las cuales pueden ser resueltas cada una por separado. Luego de obtenida la solución de cada una de las piezas, todas las soluciones parciales se deben integrar para conformar la solución al problema original. Sin embargo, esta estrategia falla cuando es necesario trabajar con piezas que están dispersas en otras piezas (porque es una pieza común y se vincula con todas ellas). Esto generalmente ocurre cuando se inyectan atributos de calidad, ya que la implementación de estos atributos es común a las diferentes piezas. Un ejemplo recurrente es el de autorización: diferentes piezas de software pueden requerir operaciones que garanticen que la autorización adecuada está presente en la ejecución de cada pieza. Como tal, autorización es un ejemplo de característica transversal, esto es, una característica que atraviesa diferentes piezas que conforman al todo.

Las características transversales se caracterizan por ser enmarañadas (*tangled*) y dispersas (*scattered*). Cuando una misma pieza posee diferentes características (*concerns*) se denomina enmarañada (*tangled*), y por otro lado, cuando una misma característica está distribuida por diferentes piezas, se dice que está dispersa (*scattered*). La comunidad de Desarrollo de Software Orientado a Aspectos (*Aspect-Oriented Software Development, AOSD*) se ocupa de las características transversales con el fin de atacar los problemas de enmarañamiento (*tangling*) y dispersión (*scattering*).

El desarrollo de software es una sucesión de descripciones en diferentes lenguajes donde a partir de una descripción se produce la siguiente. Por lo cual, si se incorporan cambios en una descripción deben ajustarse tanto las descripciones previas como la siguiente para que se mantenga la consistencia entre todas ellas. Bohem [Boehm 1997] indica que si un error se desliza en la descripción de requerimientos y recién en la etapa de codificación es detectado, la corrección del mismo costaría entre 100 y 200 veces más que si se lo hubiera corregido en la etapa de requerimientos. Por este motivo, es necesario tomar las decisiones lo antes posible en el desarrollo del software.

Yu et al. [Yu 2004] se enfocan en descubrir aspectos desde los objetivos. Los modelos basados en objetivos proveen la descripción y el análisis de las intenciones que subyacen a un nuevo sistema de software. Algunos modelos orientados a objetivos como i\* también modelan los actores que poseen tales intenciones. Estos objetivos son “groseramente hablando, precursores de los requerimientos”. Nuseibeh [Nuseibeh 2004] argumenta que el mundo del problema, el cual está habitado por clientes y usuarios, es un terreno fértil para identificar características (*concerns*). En verdad, él indica que el mundo real es a menudo la fuente más apropiada para identificar características transversales. Rashid et al. [Rashid 2006] argumentan que, es posible identificar características transversales mientras se aplica la estrategia de análisis del dominio en la etapa de ingeniería de requerimientos cuando se desarrolla software orientado al reuso.

El Léxico Extendido del Lenguaje [Leite 1993] es un glosario que tiene por finalidad describir el lenguaje del contexto de la aplicación. LEL es anclado en una idea bien simple “entender el lenguaje del problema sin preocuparse por entender el problema”. De esta forma, luego de comprender el lenguaje, el analista podrá escribir requerimientos utilizando como base el conocimiento adquirido a través del lenguaje capturado.

En este trabajo de tesis se presenta una estrategia para identificar las características transversales utilizando el Léxico Extendido del Lenguaje (LEL). La estrategia identifica características transversales en una forma similar a la que son identificados de los requerimientos. Sin embargo, mientras las técnicas tradicionales se basan en las acciones, la estrategia propuesta se basa en estados (aunque también utiliza sustantivos y acciones).

Este artículo está organizado de la siguiente manera. La sección 2 describe los trabajos relacionados. La sección 3 presenta la estrategia propuesta. La sección 4 discute aspectos de aplicabilidad, efectividad y usabilidad. La sección 5 presenta conclusiones. Finalmente, la sección 6 muestra nuevas líneas de trabajo.

## 2 Trabajos relacionados

Este capítulo describe distintos trabajos que principalmente se ocupan de identificar características transversales, ya que este es el objetivo de esta tesis: identificar características transversales en una etapa temprana del desarrollo de software. Sin embargo, en este capítulo también se describen técnicas de identificación de características transversales en el código fuente, puesto que por analogía se podría extrapolar las técnicas a una etapa temprana.

Baniassad [Baniassad 2006] presenta una forma de organizar los requerimientos descriptos en texto coloquial. El objetivo es factorizar los mismos sin repetir información, es por ello que los autores presentan una serie de actividades con este objetivo. La primera actividad es la identificación de las características repetidas y esta actividad puede ser realizada de varias formas: identificando términos aspectuales, identificando requerimientos que impactan o identificando características dispersas.

Baniassad [Baniassad 2004] describe una estrategia concreta para identificar aspectos. La estrategia se basa principalmente en analizar requerimientos identificando acciones y entidades. Las acciones son analizadas y luego de realizada una selección sobre las mismas son definidos *themes*. Luego, se organiza la información de los requerimientos guiados por los *themes*.

Sampaio [Sampaio 2005] propone una herramienta configurable para identificar características transversales a partir del texto. La herramienta es configurable porque permite identificar características transversales a partir de puntos de vistas, casos de uso, etc... Sin embargo, la identificación no comienza con estos productos, sino que la herramienta realiza un análisis del lenguaje coloquial, el que obtiene de documentos de requerimientos y a partir de ellos comienza el análisis.

Rago [Rago 2009] considera que es necesario tomar en cuenta no sólo verbos para la identificación, así que su estrategia considera además los objetos sobre los cuales trabajan los verbos.

Rashid [Rashid 2003] plantea un modelo con puntos de vista que agrupan requerimientos. El agrupamiento lo realiza a través de puntos de vistas y luego identifican puntos de vista transversales que impactan en otros requerimientos.

Bounour [Bounour 2006] describe en su trabajo varias estrategias de minería de datos basadas en el código fuente. Modelan flujo de control o flujo de datos, estructura del código fuente o dispersión del mismo, y también analizan conceptos (entidades) en función de sus referencias.

### 3 Identificación de características transversales a través del Léxico Extendido del Lenguaje

La estrategia de identificación de características transversales se basa en un modelo que captura el lenguaje de la aplicación (LEL) y a partir de él realiza el análisis que determina las características transversales candidatas.

#### 3.1 Léxico Extendido del Lenguaje (LEL)

El Léxico Extendido del Lenguaje [Leite 1993] es un glosario que tiene por finalidad describir el lenguaje del contexto de la aplicación. Su objetivo es el describir ciertas palabras o frases peculiares a la aplicación y que su comprensión son vitales para poder comprender el contexto de la misma. LEL es anclado en una idea bien simple “entender el lenguaje del problema sin preocuparse por entender el problema”. De esta forma, luego de comprender el lenguaje, el analista podrá escribir requerimientos utilizando como base el conocimiento adquirido a través del lenguaje capturado.

El LEL es un glosario en el cual se definen símbolos (términos o frases). A diferencia del diccionario tradicional que posee sólo un tipo de definición por término (puede haber muchas acepciones, sin embargo, todas son significados del término que están describiendo), en el LEL cada expresión se define a través de dos atributos: la noción (*notion*) y los impactos (*behavioural responses*). La noción describe la denotación, es decir, describe las características intrínsecas y substanciales del símbolo. Por su parte, los impactos describen la connotación, es decir, un valor secundario que adopta por asociación con un significado estricto.

Se determinan cuatro categorías básicas de símbolos: sujeto, objeto, verbo y estado. A partir de las 4 categorías y a la descripción general de los símbolos, se desprende la descripción de cada categoría que se resume la

Tabla 3-1.

Tabla 3-1. Categorías de símbolos y su descripción

Categoría	Características	Noción	Impactos
Sujeto	Elementos activos que realizan acciones	Características o condiciones que los sujetos satisfacen	Acciones que el sujeto realiza
Objeto	Elementos pasivos con los cuales los sujetos realizan acciones	Características o atributos que los objetos poseen	Acciones que son realizadas con los objetos
Verbo	Acciones que los sujetos realizan con los objetos	Objetivo que el verbo persigue	Pasos necesarios para completar la acción
Estado	Situaciones en las cuales se pueden encontrar los sujetos y los objetos	Situación que representa el estado	Acciones que deben realizarse para cambiar a otro estado

El LEL es una herramienta adecuada y efectiva para modelizar el lenguaje de una aplicación por su adecuación a los mecanismos que posee el cerebro para organizar el conocimiento experto [Wood 1997]. La utilidad del LEL como modelo ha sido observada en los trabajos de Breitman [Breitman 2003] y Gruber [Gruber 1993]. En principio LEL es una herramienta muy conveniente para expertos sin habilidades técnicas, sin embargo, las personas con tales habilidades obtendrán un mayor beneficio con su uso. La conveniencia del LEL proviene de 3 características significativas: es fácil de aprender, es fácil de usar y posee buena expresividad. Hay experiencias en dominios complejos que validan la conveniencia del LEL. Gil et al. [Gil 2000] indican que: “la experiencia de construir un LEL de una aplicación completamente desconocida para los ingenieros de requerimientos y con un lenguaje altamente complejo, puede ser considerada exitosa, desde el momento en que los usuarios fueron los que notaron que los ingenieros de requerimientos habían desarrollado un gran conocimiento sobre la aplicación”. Por su parte, Cysneiros et al. [Cysneiros 2001] indican que: “el uso del LEL fue muy bien aceptado y comprendido por los interesados (*stakeholders*). Dado que los interesados (*stakeholders*) no eran expertos en los dominios tan complejos y específicos en los que trabajaron, los autores creen que el LEL puede ser adecuado para utilizarse en muchos dominios”.

### 3.2 Identificación de características transversales con LEL

La esencia del enfoque propuesto está basado en identificar como características transversales los símbolos que con mayor frecuencia son referenciados desde los impactos de otros símbolos [Antonelli 2010]. El enfoque es mucho más complejo, a continuación se detallan los pasos que se resumen en la Figura 3-1:

- (i) Construcción del LEL organizado en grupos. Se debe definir la noción y los impactos de cada símbolo y también se debe relacionar cada símbolo con un estado.
- (ii) Conteo de referencias. Se deben considerar (contar) las referencias desde los impactos de los símbolos que forman parte de un estado distinto al estado del símbolo donde llegan las referencias. Luego, se debe calcular las referencias para el grupo en su totalidad ya que el enfoque se basa en identificar grupos como candidatos a ser características transversales.

- (iii) Ranqueo de grupos. Se deben ordenar los grupos de símbolos (estados) de acuerdo a la probabilidad de que sean considerados características transversales.
- (iv) Análisis final. En esta etapa se deben identificar los grupos con mayor probabilidad de que sean consideradas características transversales. Además, se debe realizar un análisis a nivel de símbolo para determinar si hay símbolos que distorsionan el ranking del grupo en su totalidad. Incluso, se puede realizar un análisis más fino, para determinar si ciertos símbolos particulares del grupo son los que le dan la condición de característica transversal al estado. En este punto, puede verse que un análisis a dos niveles: a nivel estado y luego a nivel símbolo dentro del estado.

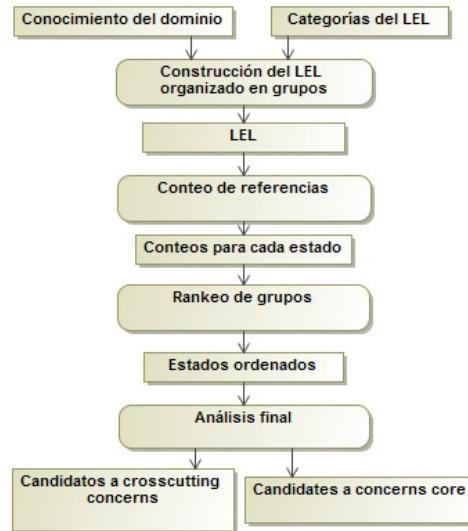


Figura 3-1. Resumen de la estrategia

### 3.2.1 Construcción de LEL organizado en grupos

La estrategia propuesta requiere que los símbolos del LEL identificados y descriptos deben estar organizados en grupos determinados por los símbolos estados. Es necesario conformar grupos de símbolos porque se necesita organizar el conocimiento en algún bloque de trabajo con el fin de determinar que bloque representa funcionalidad *core* y que bloque representa funcionalidad *aspectual*. La organización en estados radica en que el software pueden ser descrito por máquinas de estado según Mahoney [Mahoney 2005] [Mahoney 2007]. El proceso de identificación y descripción de símbolos debe cumplir las siguientes etapas:

El primer paso es la construcción del LEL. El LEL debe ser construido de la forma tradicional siguiendo los siguientes pasos: (a) identificación de símbolos; (b) categorización de cada símbolo; (c) descripción de los símbolos de acuerdo a la categoría e (d) identificación de sinónimos.

El Segundo paso consiste en identificar al símbolo más representativo y significativo de la aplicación. Debe ser un sujeto o un objeto el cual sea el elemento más relevante en el contexto de la aplicación y que además posea una máquina de estados asociada. Luego de identificado el símbolo principal, es necesario identificar los estados vinculados con ese símbolo. Es decir, es necesario identificar y describir símbolos de categoría estados.

Como tercer paso, es necesario vincular a cada uno de los símbolos del LEL (de las categorías sujetos, objetos y verbos) con alguno de los estados determinados en el paso anterior. Un símbolo se

puede relacionar con un estado por diferentes razones. Se lo puede relacionar porque el símbolo aparece o es creado en el estado, o bien, porque el acoplamiento entre el símbolo y el estado implica una gran integración, a pesar de que el símbolo es creado en otro estado.

### 3.2.2 Conteo de referencias

El conteo de referencias consiste en calcular el número de referencias que cada grupo de símbolos posee desde los impactos de símbolos que se encuentran en otro estado. Es decir, para cada uno de los grupos de símbolos (determinado por cada estado del símbolo principal y todos los símbolos asociados a él) se deben revisar todos los impactos de todos los símbolos relacionados con otros estados, y computar aquellos en donde en la descripción de los impactos hacen referencia a símbolos del estado del que se están contando las referencias. Es importante el excluir las referencias entre los símbolos del mismo grupo, ya que se pretende medir el acoplamiento entre distintos grupos, es por ello que sólo hay que concentrarse en las referencias que trascienden desde un grupo a otro. Por otro lado, es importante notar que no se mide directamente la frecuencia absoluta de referencias, sino que se debe calcular el promedio. Sucede que los distintos grupos pueden estar conformados por distintos símbolos, por lo cual, grupos con más símbolos naturalmente tendrán más referencias, sin embargo, esta gran cantidad de referencias no se debe al nivel de acoplamiento entre los grupos, sino que se debe simplemente a la mayor cantidad de símbolos que un grupo puede tener. Es por ello que el análisis de acoplamiento se debe realizar el promedio de referencias que cada símbolo posee, y esto se debe realizar dividiendo la cantidad de referencias por la cantidad de símbolos que cada grupo posee. Además, es importante el contar las referencias desde los otros grupos por separado, es decir, en lugar de tener un único valor que represente la suma de todas las referencias, se deben acumular las referencias por cada uno de los distintos grupos. Sucede que esta es una medida de dispersión, puesto que determina el nivel de acoplamiento de cada grupo por separado, por lo cual, en definitiva indica cuando disperso está referenciado un grupo entre todos los demás. Finalmente, con el fin de obtener cierta precisión en el análisis, los promedios de referencias se deben calcular por cada categoría, es decir, se deben calcular las referencias producidas sólo por sujetos, sólo por objetos y sólo por verbos. Es decir, se debe determinar en que medida los sujetos de otros grupos referencian al grupo bajo análisis, y por separado como lo referencian los objetos y por otro lado los verbos. Esta distinción es importante ya que las metodologías tradicionales solo analizan los verbos (acciones, comportamiento) mientras que la estrategia propuesta además considera sujetos y objetos. Por lo cual, el hacer esta distinción permite comparar nuestra propuesta con otras.

### 3.2.3 Ranqueo de grupos

Van Den Berg et al. [Van Den Berg 2005] definen dos características que determinan la presencia de características transversales: *scattering* (dispersión) y *tangling* (enmarañamiento). Él determina que *scattering* ocurre cuando “un elemento origen es relacionado a múltiples elementos destinos”, mientras que *tangling* ocurre cuando “un elemento destino es relacionado a múltiples elementos fuentes”. Para el enfoque propuesto, los elementos (ya sean origen o destino) son los grupos de símbolos. Y se debe usar las referencias desde los impactos de los símbolos para medir *scattering* y *tangling*. Luego, los grupos candidatos a ser considerados características transversales deben satisfacer dos condiciones: (i) deben poseer muchos otros grupos que los referencian y (ii) deben poseer un alto promedio de referencias. Es importante maximizar ambas variables ya que (i) el número de grupos que los referencian indica cuan disperso (*scattered*) está el grupo, mientras que (ii) el promedio de referencias indica cuando acoplado (*tangled*, enmarañado) se encuentra el grupo.

Estas dos variables, determinan 3 situaciones posibles más. La Figura 3-2 muestra un diagrama xy donde el eje x representa el promedio de referencias (es decir, cuan acoplado el grupo es) mientras

que el eje y representa los grupos que referencian (es decir, cuando disperso el grupo está). El diagrama está sombreado de acuerdo a las dos variables, de forma tal que la porción más oscura indica la ubicación de las características transversales candidatas.

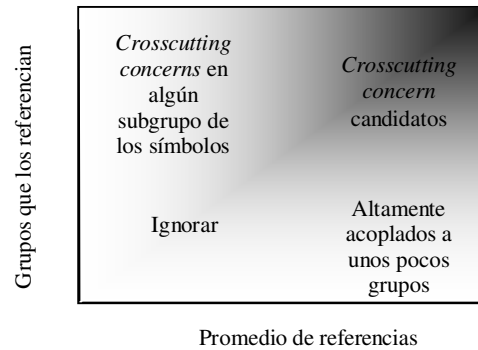


Figura 3-2. Nivel de dispersión (*scattering*) y acoplamiento (*coupling*)

### 3.2.4 Análisis final

Un análisis detallado debe ser realizado luego de que el conteo de referencias y el ranking se hayan realizado. El análisis detallado consiste en revisar los símbolos y determinar si algún símbolo podría alterar y sesgar el ranking de los grupos. Es necesario prestar atención a aquellos símbolos principales (sujetos, objetos y verbos) los cuales no se modularizarán como características transversales, sino que se hará como funcionalidad *core*. Estos símbolos podrían tener una gran cantidad de referencias y podrían generar que todo el grupo lo posea, por lo cual, el símbolo debe ser separado del grupo (no cuentan las referencias que recibe, sin embargo, si cuentan las referencias que posee). Como parte de este análisis final, es necesario analizar si dentro de un grupo además de haber símbolos *core* que no deben ser considerados características transversales también existen símbolos *aspectuales* los cuales surgen de un análisis más detallado y son más representativos de las características transversales que el mismo grupo.

## 4 Aplicabilidad, efectividad y usabilidad

La estrategia se aplicó a dos aplicaciones reales. Por un lado, una aplicación relacionada con un sistema antievasión de impuestos en Argentina y por otro lado, un portal web de una compañía que publica noticias en diferentes países de América Latina. Los casos de estudio tienen por objetivo mostrar la aplicabilidad de la estrategia en dos casos reales y no se pretende demostrar la efectividad, la cual se demuestra a través de un experimento. De todas formas, junto con cada caso de estudio se comparan los resultados obtenidos por la estrategia con la opinión de diferentes expertos que estuvieron involucrados con ambos sistemas.

La efectividad se ha probado comparando la estrategia del LEL contra la estrategia Theme/Doc [Baniassad 2004]. Con este fin, se definió una misma aplicación para la cual se aplicó ambas estrategias. Luego, se contrastaron los resultados en términos de las características transversales identificadas por cada una de las estrategias. La comparación se hizo a través de un experimento de tipo aleatorio simple entre sujetos (*simple between-subjects randomized*) con características realísticas.

Finalmente, la usabilidad está asegurada puesto que se desarrolló una herramienta que implementa la estrategia. La herramienta (llamada TICCWL) fue desarrollada como trabajo de grado [Aramayo 2012] e implementa los cálculos necesarios para identificar las características transversales, mientras que se apoya en otra herramienta (llamada C&L) para la edición del LEL. Es decir, TICCWL debe recibir el LEL construido a través de C&L [Felicísimo 2004] [Almentero 2009] [C&L 2009] y una vez importado, permite configurarlo, realizar los cálculos establecidos por la estrategia de identificación de características transversales del LEL y finalmente visualizar los resultados, para eventualmente hacer cambios de configuración y volver a realizar los cálculos.

## 5 Conclusiones

La presente tesis describe una estrategia para identificar características transversales del software (i.e. aspectos) en el lenguaje del contexto de la aplicación capturado a través del Léxico Extendido del Lenguaje. A continuación se describen los beneficios y limitaciones de la estrategia.

En primer lugar, la estrategia propuesta tiene la ventaja de ser muy fácil de usar. Para ello reúne varias características. La estrategia comienza con una descripción del contexto de la aplicación en lenguaje natural. Esta descripción se debe realizar a través del LEL. Esta tarea puede ser realizada por cualquier interesado (*stakeholder*) sin ninguna clase de experiencia y con poco esfuerzo. Hay experiencias en dominios complejos que validan la facilidad de utilización del LEL: Gil et al. [Gil 2000] y Cysneiros et al. [Cysneiros 2001]. Por otro lado, luego de construido el LEL la estrategia consiste en realizar sistemáticamente una serie de cálculos. Si bien los mismos son críticos y deben realizarse cuidadosamente, existe una aplicación que automatiza todos los cálculos y obtiene los resultados finales. Además, LEL que es una herramienta propuesta desde hace varios años por lo cual, la sobrecarga de trabajo necesario para identificar características transversales es mínimo y la ventaja de contar con esta información es por demás valiosa. Finalmente, la estrategia propuesta puede ser aplicada casi en cualquier etapa del desarrollo de software, incluidas las etapas de requerimientos y de codificación. El elemento clave que permite utilizar la estrategia en cualquier etapa es el LEL, ya que es él el que sintetiza el conocimiento de la aplicación, el cual puede originarse directamente del lenguaje de la aplicación, de la especificación de requerimientos o del código fuente. Este punto de la abstracción del LEL para identificar las características transversales, provee otra ventaja más. Las técnicas tradicionales generalmente se basan en las acciones para identificar características transversales, la estrategia propuesta se basa también en sujetos, objetos y estados.

A pesar de los beneficios, la estrategia presenta algunas limitaciones. La estrategia se basa en el LEL, por lo cual la descripción de los símbolos es determinante en la efectividad de la estrategia. En primer lugar, las descripciones deben ser lo más precisas como sea posible, ya que si el LEL no es correcto, las características transversales identificadas no serán de valor. Por otro lado, también es importante el nivel de detalle con que se describe cada uno de los estados. Los mismos se deben definir uniformemente, es decir, deben ser descriptos usando el mismo nivel de abstracción y detalle. Si un estado posee muchos símbolos, este estado potencialmente podría contener más referencias, y esto podría ocasionar que un estado sea identificado como característica transversal porque simplemente se lo describió con más detalle, mientras que otro estado que si debe serlo, se lo definió en forma muy sencilla. Es por ello que es necesario definir símbolos uniformemente con el mismo nivel y criterio de abstracción, incluso con el criterio de no sumar símbolos insignificantes que alteren los resultados. Y no sólo hay que considerar los símbolos a describir, también es necesario prestar atención a la descripción de los símbolos en sí. Ya que la descripción de los impactos afecta a la estrategia de la misma forma. Si un símbolo posee una descripción extensa con un gran nivel de detalle, mientras que el resto poseen descripciones más simples, el símbolo con el



mayor nivel de detalle afectará las referencias y con ello los conteos, y tal vez ocasionará que el estado al que pertenece sea considerado como característica transversal aunque no lo sea.

## 6 Trabajos futuros

Dado que la descripción del LEL es esencial para la efectividad de la estrategia, estamos trabajando en como asegurar que el LEL utilizado es de calidad. Para ello vamos a proponer guías bien precisas sobre como construir el LEL. Hemos realizado un experimento en donde 20 grupos de participantes construyen un LEL para una aplicación única y a partir de los inconvenientes que se presentan en el proceso de construcción y con ayuda de la literatura identificamos las guías. Por otro lado, la identificación de características transversales a partir del LEL es una de las componentes de la estrategia de desarrollo de software que estamos desarrollando. Estamos trabajando en obtener requerimientos funcionales, requerimientos no funcionales y cuestiones navegacionales a partir del LEL. De esta forma, con las características transversales, los requerimientos, atributos de calidad (requerimientos no funcionales) y cuestiones de navegación a partir del LEL se dispone de un núcleo importante de información para diseñar la arquitectura y planificar el desarrollo del software. Planificación que incluso será enriquecida con las métricas que también estamos diseñando estableciendo una relación con otras métricas como *use case points*.

## 7 Referencias

- [Almentero 2009] Kinder Almentero E. “Re-engenharia do software C&L para plataforma Lua-Kepler utilizando princípios de transparência” dissertação de Mestrado PUC Rio, Rio de Janeiro, 8 abril de 2009.
- [Antonelli 2010] Antonelli L, Rossi G, Leite JCSP “Early identification of crosscutting concerns in the domain model guided by states”, In: proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, ISBN:978-1-60558-639-7, 2010.
- [Aramayo 2012] Aramayo A, Rossi J, “Una herramienta para identificar crosscutting concerns a través de LEL” tesis de grado en curso, Facultad de Informática, Universidad Nacional de La Plata, 2012.
- [Baniassad 2004] Baniassad E., Clarke S., “Finding Aspects In Requirements with Theme/Doc”, Position Paper for the Early-Aspects Workshop held as part of AOSD, March 2004.
- [Baniassad 2006] Baniassad E., Clements P.C., Araujo J., Moreira A., Rashid A., Tekinerdogan B., “Discovering early aspects, In: IEEE software”, ISSN:0740-7459, Volume 23, Issue 1, January, pp 61 – 70, 2006.
- [Boehm 1997] Boehm B. W., Software Engineering, Computer society Press, IEEE, 1997.
- [Bounour 2006] Bounour N., Ghoul S., Atil F., “A comparative classification of aspect mining approaches”, Journal of Computer Science Volume 2, Number 4, ISSN 1549-3636, 2005 Science publication, pp 322 - 325, 2006.
- [Breitman 2003] Breitman K.K., Leite J.C.S.P., “Ontology as a Requirements Engineering Product”, In: Proceedings of the 11th IEEE International Conference on Requirements Engineering (RE), IEEE Computer Society, Monterey Bay, California, USA, ISBN 0-7695-1980-6, 2003.
- [C&L 2009] C&L Tool, available at <http://pes.inf.puc-rio.br/cel/>, accessed in October, 2009.

- [Cysneiros 2001] Cysneiros L.M., Leite J.C.S.P., “Using the Language Extended Lexicon to Support Non-Functional Requirements Elicitation” in proceedings of the Workshops de Engenharia de Requisitos, Wer’01, Buenos Aires, Argentina, 2001.
- [Felicíssimo 2004] Felicíssimo C. H., Leite J.C.S.P., Breitman K. K., Fernandes da Silva L., “C&L: Um Ambiente para Edição e Visualização de Cenários e Léxicos”, in sessão de Ferramentas do Simpósio Brasileiro de Engenharia de Software Brasília, Brasil, October, pp 43 – 48, 2004.
- [Gil 2000] Gil D., Figueroa D. A., Oliveros A., “Producción del LEL en un Dominio Técnico.Informe de un caso.”, in proceedings of the Workshops de Engenharia de Requisitos, Wer’00, Rio de Janeiro, Brazil, 2000.
- [Gruber 1993] Gruber T.R., “A translation approach to portable ontology specifications”, In: Knowledge acquisition journal, volume 5, June 1993, pp 199 – 220, 1993.
- [Leite 1993] Leite J.C.S.P., Franco A.P.M., “A Strategy for Conceptual Model Acquisition”, In: Proceedings of the First IEEE International Symposium on Requirements Engineering, San Diego, California, IEEE Computer Society Press, pp 243-246, 1993.
- [Mahoney 2005] Mahoney M., “Modeling Crosscutting Concerns in Reactive Systems with Aspect-Orientation”, In: Models 2005, Doctoral Symposium, 2005.
- [Mahoney 2007] Mahoney M., Elrad T., “Generating code from scenario and state based models to address crosscutting concerns”, In: proceedings of the sixth international workshop on scenarios and state machines (SCESM’07), IEEE, 2007.
- [Nuseibeh 2004] Nuseibeh B., “Crosscutting requirements, In: proceedings of the 3rd international conference on Aspect-oriented software development”, ISBN:1-58113-842-3, Lancaster, UK, pp 3 – 4, 2004.
- [Rago 2009] Rago A., Marcos C., “Análisis Semántico para la Identificación de Aspectos”, in Jornadas Chilenas de Computación, 2009.
- [Rashid 2003] Rashid A., Moreira A., Araújo J., “Modularisation and composition of aspectual requirements”, In: proceedings of the 2nd international conference on Aspect-oriented software development, ISBN:1-58113-660-9, Boston, Massachusetts, pp 11 – 20, 2003.
- [Rashid 2006] Rashid A., Moreira A., “Domain Models are NOT Aspect Free”, In: Proceedings of MoDELS/UML, Springer, Lecture Notes in Computer Society, pp. 155-169, 2006.
- [Sampaio 2005] Sampaio A., Loughran N., Rashid A., Rayson P., “Mining Aspects in Requirements”, In Proceeding of the Workshop on Early Aspects (held with AOSD 2005) Illinois, Chicago, USA, 2005.
- [Van Den Berg 2005] Van Den Berg K., Conejero J.M., “A Conceptual Formalization of Crosscutting in AOSD”, In: Proceeding of Desarrollo de Software Orientado a Aspectos, Granada, España, 2005.
- [Wood 1997] Wood L.E., Semi-structured interviewing for user-centered design, Interactions of the ACM, april-may, pp 48-61, 1997.
- [Yu 2004] Yu Y., Leite J.C.S.P., Mylopoulos J., “From Goals to Aspects: Discovering Aspects from Requirements Goal Models”, In: International Requirements Engineering Conference, pp 38-47, 2004.