



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 obtener el grado de Doctor en Ciencias Informáticas

**Facultad de Informática
Universidad Nacional de La Plata**

Febrero 2012

Abstract

En este trabajo de tesis presentamos 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 en 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). Debido a que la construcción del LEL se realiza tempranamente en el proceso de desarrollo del software, la estrategia propuesta detecta características transversales tan temprano como es posible y 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, lo cual redundará en beneficios para 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. Se describe una pequeña aplicación bancaria que se utiliza para ejemplificar la estrategia y su aplicación. También se describen dos casos de estudios sobre aplicaciones de mediana magnitud. Ambos casos de estudios están basados en aplicaciones reales. Uno de ellos es una aplicación de control antievasión de impuestos y la otra es un portal web que publica noticias. Además, la presente tesis también describe un experimento que se llevó a cabo con el fin de mostrar la efectividad de la estrategia. El experimento fue planteado suficientemente realista y con una población de 20 sujetos como para obtener un muestreo de resultados suficiente. Por último, la tesis describe una herramienta que permite asistir en la aplicación de la estrategia propuesta.

Índice general

Abstract	2
Índice de figuras	5
Índice de tablas.....	7
1 Introducción	9
1.1 Motivación.....	9
1.2 Características transversales en el código fuente	11
1.3 Características transversales en los requerimientos.....	14
1.4 Léxico Extendido del Lenguaje (LEL).....	16
1.5 Resumen	16
1.6 Contribución	17
1.6.1 Objetivo	17
1.6.2 Aportes	17
1.7 Publicaciones asociadas a la tesis	18
1.8 Organización del documento.....	18
2 Revisión de trabajos relacionados	19
2.1 Introducción.....	19
2.2 Descubriendo aspectos tempranos	20
2.3 Theme / DOC.....	22
2.4 EA-Miner	23
2.5 Análisis semántico para la identificación de aspectos	24
2.6 Modularización y composición de requerimientos aspectuales.....	25
2.7 Clasificación comparativa de estrategias de minería de aspectos.....	26
2.8 Conclusiones	27
3 Identificación de características transversales a través del Léxico Extendido del Lenguaje.....	28
3.1 Léxico Extendido del Lenguaje (LEL).....	28
3.2 Identificación de características transversales con LEL	36
3.2.1 Construcción de LEL organizado en grupos.....	41
3.2.2 Conteo de referencias	44
3.2.3 Ranqueo de grupos	46
3.2.4 Análisis final	48
3.3 Casos de estudio.....	49
3.3.1 Caso de estudio antievasión de impuestos.....	50
3.3.2 Caso de estudio del portal web.....	56
4 Experimentación	62
4.1 Planificación	63
4.1.1 Objetivo	63
4.1.2 Participantes	64
4.1.3 Materiales del experimento	65
4.1.4 Tarea.....	65
4.1.5 Hipótesis y variables.....	66
4.1.6 Diseño.....	67
4.1.7 Procedimiento	67
4.1.8 Procedimiento de análisis.....	67
4.2 Ejecución	68
4.3 Análisis	72

4.4	Conclusiones	75
4.4.1	Amenazas a la validez (Treaths to validity).....	76
5	Soporte automatizado a la estrategia	78
5.1	Motivación.....	78
5.2	C&L	79
5.3	Arquitectura	80
5.4	Tecnologías utilizadas.....	81
5.5	Uso	83
6	Conclusiones	87
6.1	Contribuciones	87
6.2	Beneficios	87
6.3	Limitaciones.....	89
6.4	Trabajos futuros	90
7	Referencias	92
8	Anexos	95
8.1	Descripción caso de estudio antievasión de impuestos.....	95
8.2	Descripción caso de estudio portal web	100
8.3	Material del experimento	105
8.3.1	Descripción de la aplicación.....	106
8.3.2	Vistas de acciones	106
8.3.3	Símbolos de LEL agrupados por estados	109

Índice de figuras

Figura 1-1. Clase <i>BankAccount</i> con su funcionalidad base.....	12
Figura 1-2. Clase <i>BankAccount</i> con funcionalidad para autorizar y dejar registro de las operaciones.....	12
Figura 1-3. Declaración de los aspectos de autorización y <i>logging</i>	14
Figura 1-4. Requerimientos para una aplicación bancaria.....	14
Figura 1-5. Requerimientos refactorizados de la aplicación bancaria.....	15
Figura 3-1. Descripción del símbolo <i>cliente</i>	33
Figura 3-2. Descripción del símbolo <i>cuenta</i>	34
Figura 3-3. Descripción del símbolo <i>extraer</i>	34
Figura 3-4. Descripción del símbolo <i>solicitada</i>	34
Figura 3-5. Descripción del símbolo <i>depositar</i>	36
Figura 3-6. Descripción del símbolo <i>extraer</i>	36
Figura 3-7. Descripción del símbolo <i>consultar el saldo</i>	36
Figura 3-8. Descripción del símbolo <i>registrar</i>	36
Figura 3-9. Descripción del símbolo <i>corroborar</i>	37
Figura 3-10. Referencias a los símbolos <i>corroborar</i> y <i>registrar</i>	38
Figura 3-11. Definición alternativa para los símbolos <i>corroborar</i> y <i>registrar</i>	38
Figura 3-12. Máquina de estados de la interacción con un cajero automático.....	40
Figura 3-13. Definición de los métodos <i>acceptPassword</i> y <i>acceptOperation</i>	40
Figura 3-14. Resumen de la estrategia.....	42
Figura 3-15. Máquina de estados correspondiente a la apertura de una cuenta bancaria.....	43
Figura 3-16. Estrategia de conteo.....	45
Figura 3-17. Nivel de dispersión (<i>scattering</i>) y acoplamiento (<i>coupling</i>).....	47
Figura 3-18. Ranking de los grupos del ejemplo bancario.....	48
Figura 3-19. Ranking de los grupos del sistema antievasión.....	52
Figura 3-20. Ranking de grupos del portal web.....	58
Figura 4-1. <i>Plantilla</i> de requerimientos.....	69
Figura 4-2. Requerimientos de ejemplo para cada plantilla.....	69
Figura 4-3. Requerimientos de la aplicación de pedidos.....	69
Figura 4-4. Definición del símbolo <i>pedido</i>	70
Figura 4-5. Definición del símbolo <i>nuevo</i>	70
Figura 4-6. Definición del símbolo <i>jefe de área</i>	70
Figura 4-7. Definición del símbolo <i>crear</i>	71
Figura 4-8. Definición del símbolo <i>asignar</i>	71
Figura 5-1. Pantalla de edición de un símbolo de LEL en C&L.....	79
Figura 5-2. Ejemplo de archivo XML generado por C&L.....	81
Figura 5-3. Etapas de ejecución.....	82
Figura 5-4. Pantalla para cargar un archivo XML.....	83
Figura 5-5. Pantalla inicial para agrupar símbolos en estados.....	84
Figura 5-6. Algunos símbolos agrupados en estados.....	84
Figura 5-7. Características transversales candidatas ordenadas por posibilidad de ser consideradas como tales.....	85
Figura 5-8. Diagrama XY que muestra la dispersión de los grupos.....	85
Figura 5-9. Reporte detallado de referencias por grupos.....	85
Figura 5-10. Reporte de referencias por símbolos.....	86

Figura 8-1. <i>Vista de acción (action view)</i> del sujeto #1.....	106
Figura 8-2. <i>Vista de acción (action view)</i> del sujeto #2.....	107
Figura 8-3. <i>Vista de acción (action view)</i> del sujeto #3.....	107
Figura 8-4. <i>Vista de acción (action view)</i> del sujeto #4.....	107
Figura 8-5. <i>Vista de acción (action view)</i> del sujeto #5.....	107
Figura 8-6. <i>Vista de acción (action view)</i> del sujeto #6.....	108
Figura 8-7. <i>Vista de acción (action view)</i> del sujeto #7.....	108
Figura 8-8. <i>Vista de acción (action view)</i> del sujeto #8.....	108
Figura 8-9. <i>Vista de acción (action view)</i> del sujeto #9.....	108
Figura 8-10. <i>Vista de acción (action view)</i> del sujeto #10.....	109

Índice de tablas

Tabla 3-1. Categorías de símbolos y su descripción.....	30
Tabla 3-2. Símbolos correspondientes a la aplicación bancaria	33
Tabla 3-3. Matriz de adyacencias de las referencias de los símbolos de la aplicación bancaria.	35
Tabla 3-4. Referencias a los símbolos <i>registrar</i> y <i>corroborar</i> en la aplicación bancaria.....	37
Tabla 3-5. Relación entre los símbolos y los estados para la aplicación bancaria	44
Tabla 3-6. Conteo de referencias entre los distintos grupos de la aplicación bancaria.....	45
Tabla 3-7. Conteo de referencias detallado	46
Tabla 3-8. Orden final.....	48
Tabla 3-9. Análisis detallado.....	49
Tabla 3-10. Conteo de referencias para la aplicación de antievasión de impuestos.....	52
Tabla 3-11. Orden final de los grupos del sistema antievasión	53
Tabla 3-12. Análisis detallado del sistema antievasión de impuestos.....	54
Tabla 3-13. Características transversales identificadas por la estrategia y por el equipo de desarrollo.....	57
Tabla 3-14. Conteo de referencias para el caso de estudio del portal web de noticias	58
Tabla 3-15. Orden final de los grupos del portal web	59
Tabla 3-16. Análisis detallado de los grupos del portal web.....	60
Tabla 3-17. Características transversales candidatas identificados por la propuesta y por el equipo de desarrollo.....	61
Tabla 4-1. Variables del experimento	66
Tabla 4-2. Símbolos de ejemplo con su categoría.....	70
Tabla 4-3. Símbolos de LEL agrupados por estado.....	71
Tabla 4-4. Características transversales identificadas a través de las especificaciones de requerimientos.....	73
Tabla 4-5. Características transversales identificadas a través de la especificación de LEL.....	73
Tabla 4-6. Promedio de características transversales identificadas por la estrategia Theme/Doc.....	74
Tabla 4-7. Promedio de características transversales identificadas por la estrategia de LEL	74
Tabla 4-8. Comparación de características transversales identificadas por cada estrategia.....	75
Tabla 5-1. Funcionalidad de C&L.	80
Tabla 8-1. Lista completa de símbolos definidos para el caso de estudio antievasión de impuestos	95
Tabla 8-2. Detalle de símbolos y sus referencias para el grupo de establecimiento de deuda	96
Tabla 8-3. Detalle de símbolos y sus referencias para el grupo de procedimiento a nivel administrativo.....	96
Tabla 8-4. Detalle de símbolos y sus referencias para el grupo de procedimiento a nivel judicial	97
Tabla 8-5. Detalle de símbolos y sus referencias para el grupo procedimiento a nivel cautelar.....	97
Tabla 8-6. Detalle de símbolos y sus referencias para el grupo cancelación de la deuda.....	98
Tabla 8-7. Detalle de símbolos y sus referencias para el grupo de Medidas preventivas	98
Tabla 8-8. Detalle de símbolos y sus referencias para el grupo Plan de facilidades de pago	99
Tabla 8-9. Detalle de símbolos y sus referencias para el grupo de Subasta	100
Tabla 8-10. Detalle de símbolos y sus referencias para el grupo Deuda reclamada por error	100
Tabla 8-11. Lista completa de símbolos definidos para el caso de estudio portal web	100
Tabla 8-12. Detalle de símbolos y sus referencias para el grupo de redactado.....	101
Tabla 8-13. Detalle de símbolos y sus referencias para el grupo de Rechazado.....	102
Tabla 8-14. Detalle de símbolos y sus referencias para el grupo de Aceptado.....	102
Tabla 8-15. Detalle de símbolos y sus referencias para el grupo de HTML construido.....	102
Tabla 8-16. Detalle de símbolos y sus referencias para el grupo de Versiones auxiliares.....	103
Tabla 8-17. Detalle de símbolos y sus referencias para el grupo de Indexado	104
Tabla 8-18. Detalle de símbolos y sus referencias para el grupo de Calendarizado	104

Tabla 8-19. Detalle de símbolos y sus referencias para el grupo de Taggeado	105
Tabla 8-20. Detalle de símbolos y sus referencias para el grupo de Publicado	105
Tabla 8-21. Detalle de símbolos y sus referencias del sujeto #1	109
Tabla 8-22. Detalle de símbolos y sus referencias del sujeto #2	110
Tabla 8-23. Detalle de símbolos y sus referencias del sujeto #3	110
Tabla 8-24. Detalle de símbolos y sus referencias del sujeto #4	111
Tabla 8-25. Detalle de símbolos y sus referencias del sujeto #5	112
Tabla 8-26. Detalle de símbolos y sus referencias del sujeto #6	112
Tabla 8-27. Detalle de símbolos y sus referencias del sujeto #7	113
Tabla 8-28. Detalle de símbolos y sus referencias del sujeto #8	114
Tabla 8-29. Detalle de símbolos y sus referencias del sujeto #9	115
Tabla 8-30. Detalle de símbolos y sus referencias del sujeto #10	115

1 Introducción

El presente capítulo describe la pertinencia y necesidad de este trabajo de tesis, como así también los elementos necesarios para comprender el objetivo del trabajo mismo. Por lo cual, a continuación se describirá la motivación que llevó al presente trabajo. Y luego de ello se describirán los conceptos relacionados con características transversales (*crosscutting concerns*) tanto en el código fuente, como en los requerimientos a pesar de que esta tesis se ocupa de identificar características transversales en etapa previa a los requerimientos. Para terminar con este capítulo, se describirá más concretamente cual es la contribución de la presente tesis.

1.1 Motivación

El desarrollo de sistemas de software es una tarea muy compleja por diferentes motivos. Los mismos incluyen características propias del software como así también tareas que se deben llevar a cabo para su construcción.

La gestión del desarrollo es un proceso complejo ya que en la construcción intervienen muchos actores con diferentes capacidades y background que realizan tareas muy distintas y variadas. Toda esta diversidad hace difícil que se pueda organizar la construcción del software en un cronograma preciso que sea definido al inicio del proyecto. En general, al comienzo del proyecto no se posee mucho detalle, por lo cual, se construye un cronograma bastante preciso, en el cual, se definen hitos, los cuales son más bien un objetivo a cumplir que un tiempo de finalización estimado. Como consecuencia, es muy difícil cumplirlos y más aún, sobreponerse a los retrasos.

En otras disciplinas de la ingeniería, como la ingeniería civil por ejemplo, es usual realizar tareas repetitivas, en las cuales un incremento de las personas que las realizan, se traduce en forma inversamente proporcional al tiempo necesario para completar la tarea. Sin embargo, esto no ocurre con la construcción de software. En general, las tareas son muy específicas y si se incluye más personas para sobreponerse a un retraso, se terminará perdiendo aún más tiempo (al menos al comienzo de la incorporación de las nuevas personas). Este hecho se conoce como la Ley de Brooks [Brooks 1995].

Además, por la naturaleza propia del software es que su construcción es una tarea compleja. Brooks [Brooks 1995] define 4 características que hacen que el software sea un producto muy particular y así también su construcción.

(i) Invisibilidad: el software no es un producto visible como lo puede ser un edificio, un auto o un avión. Esta característica impone al ser humano una barrera de percepción de tamaño, forma y estructura.

(ii) Modificabilidad: el software es construido a través de descripciones, las cuales son altamente maleables. Sin embargo, el hecho de que se puedan describir, no significa que se pueda modificar. Entendiendo por modificar a introducir cambios de una forma coherente y consistente. Esta capacidad de modificabilidad crea la ilusión de que es fácilmente cambiabile.

(iii) Conformidad: el software tiene que conformarse a su infraestructura, la cual puede ser a su vez otro software, el cual, incluso puede no existir aún. Por lo cual, cambios que pudieran ocurrir en su infraestructura originan cambios en el sistema.

(iv) Complejidad: el software es complejo por naturaleza ya que debe abstraer la complejidad que existe en su contexto para proveer un soporte o automatizarlo. Si el software se simplificara, no cumpliría con su función.

Como tal, 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. Esta solución es utilizada en el desarrollo de software en uno de los paradigmas más difundidos: el diseño estructural. En él, el sistema es descompuesto por una jerarquía de módulos, donde cada módulo resuelve una parte del problema. La integración de todos los módulos conforma el sistema final. La técnica se basa en encapsular cada característica (*concern*) en un bloque de software (módulo). 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*). Una misma pieza posee diferentes características (*concerns*) es por eso que se denomina enmarañada (*tangled*), y por otro lado, una misma característica esta distribuida por diferentes piezas, es por eso que 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. Más aún, Mizuno desarrollo un modelo de “*catarata de errores*” [Mizuno 1983] en el cual, establece que en cada etapa, la posibilidad de que ocurran errores es mayor que en la etapa previa, ya que cada etapa se basa en los productos de etapa previa, la descripción producida en una etapa podría contener un error y se trabaje con el mismo en la siguiente etapa. Luego, en la siguiente etapa, dado que se está trabajando a partir de una descripción con errores, aunque el trabajo y el proceso realizado sea correcto, no lo será la descripción, ya que

estará basado en descripciones incorrectas y la descripción resultante también lo será. 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, el indica que el mundo real es a menudo la fuente más apropiada para identificar características transversales. Rashid et al. [Rashid 2006] argumentan que por ejemplo, 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.

1.2 Características transversales en el código fuente

Una característica transversal en el código fuente es denominado aspecto. Como cualquier otra característica transversal en cualquier otra etapa del desarrollo de software, un aspecto es un artefacto que necesita ser incorporado en distintos bloques (clases, métodos o procedimientos). A pesar de la jerarquía de descomposición dominante en la cual las aplicaciones se organizan, la misma se encuentra atravesada por esta clase de características (*concern*).

La programación orientada a aspectos (Aspect-oriented programming, AOP) es una forma de modularizar las características transversales, al igual que la programación orientada a objetos es una forma de modularizar las características tradicionales. AOP provee una forma de modularizar los aspectos en módulos, y luego, a través de un mecanismo conocido como entrejido de aspectos (*weaving aspects*), el código aspectual es inyectado en todos los módulos en donde el aspecto es necesario. AspectJ [AspectJ 2011] es una extensión al lenguaje de programación JAVA que agrega capacidades orientadas a aspectos al mismo.

Para entender el concepto de aspectos considere una aplicación bancaria la cual posee una clase *BankAccount* con las siguientes operaciones: *withdraw* (extraer), *deposit* (depositar) and *getBalance* (consultar el saldo). La siguiente figura muestra la definición de la clase en Java con las 3 operaciones. El ejemplo muestra como cada método posee encapsulado una sola característica (*concern*).

```
public class BankAccount {  
  
    private int balance;  
  
    public void deposit(int anAmount) {  
        balance = this.getBalance() + anAmount;  
    }  
  
    public void withdraw(int anAmount) {  
        if (this.getBalance() >= anAmount) {  
            balance = this.getBalance() - anAmount;  
        }  
    }  
}
```

```

    }
}

public int getBalance() {
    return balance;
}
}

```

Figura 1-1. Clase *BankAccount* con su funcionalidad base

Considere que el banco del ejemplo previo necesita cierto nivel de seguridad, por lo cual, cada operación debe incluir el comportamiento para proveer autorización. Además, el banco necesita dejar registro de cada operación (*log*). En la siguiente figura se incorpora a la funcionalidad básica de la Figura 1-1 el comportamiento necesario para proveer autorización y para dejar registro de las operaciones.

```

public class BankAccount {

    private int balance;
    private ArrayList <Subject> owners;
    private ActivityHistory history;

    public void deposit(int anAmount) {
        this.logMessage("deposit requested");
        if (this.isSubjectAuthorizedToOperate ()) {
            balance = this.getBalance() + anAmount;
        }
    }

    public void withdraw(int anAmount) {
        this.logMessage("withdraw requested");
        if (this.isSubjectAuthorizedToOperate ()) {
            if (this.getBalance() >= anAmount) {
                balance = this.getBalance() - anAmount;
            }
        }
    }

    public int getBalance() {
        this.logMessage("getBalance requested");
        if (this.isSubjectAuthorizedToOperate ()) {
            return balance;
        }
    }

    private boolean isSubjectAuthorizedToOperate() {
        return owners.contains (history.activeSession.subject ());
    }

    private void logMessage(String message) {
        history.log.add (message);
    }
}

```

Figura 1-2. Clase *BankAccount* con funcionalidad para autorizar y dejar registro de las operaciones

La Figura 1-2 muestra que las acciones para implementar autorización y *logging* se repiten en los métodos *deposit*, *withdraw* y *getBalance*. Sin embargo, utilizando los conceptos de la orientación a aspectos, es posible modularizar las características (concerns) de autorización y *logging* de forma que el código se simplifique y haciendo más fáciles los cambios, ya que sólo se debería modificar en un sólo lugar.

Para declarar un aspecto en AspectJ es necesario definir dos características: *pointcut* y *advice*. Un *pointcut* declara un conjunto de puntos en la ejecución de un programa donde el código de los aspectos es inyectado. Estos puntos no pueden ser cualquier punto en el programa, por el contrario, hay un conjunto de puntos bien definidos que son conocidos como *join points*. Luego, el código inyectado es denominado *advice*. La Figura 1-3 muestra dos aspectos: autorización y *logging*. Cada operación de la clase *BankAccount* (*deposit*, *withdraw* and *getBalance*) necesita ser autorizado y *logueada*, por lo cual, los *pointcuts* definidos en cada aspecto se refieren a cada operación de la clase *BankAccount*. Esto se realiza de la siguiente manera: `(* BankAccount.* ..)`. Esto significa que cualquiera sea la firma (*signature*) del método (*return_type class_name.name_of_the_method (parameters)*), si el método es de la clase *BankAccount*, el mismo debe ser considerado. Además, el *pointcut* de este ejemplo además hace una declaración necesaria para trabajar dentro del *advice* con el objeto receptor del mensaje. Esta declaración es: `(BankAccount bankAccount)` y `&& target (bankAccount)`. Finalmente, es importante mencionar que el aspecto autorización previene la ejecución de la funcionalidad base (*core*) si el sujeto no es autorizado, mientras que el aspecto de *logging* solo agrega comportamiento pero no altera la funcionalidad base. Por lo cual, las *keywords call* y *execution* son agregadas en la declaración del *pointcut*. Los *advices* requieren menos explicación. El *advice* de *logging* es ejecuta antes de la invocación del método y consiste en registrar (*log*) el mensaje invocado. El *advice* de autorización es más complejo ya que verifica si el usuario está habilitado o no para realizar la operación, y previene la ejecución de la operación si el usuario no lo está. Mientras que si lo está el método se ejecuta normalmente. Finalmente, cada aspecto agrega una definición de variable y un método. El aspecto de autorización necesita la variable *owners* y el método *isSubjectAuthorizedToOperate*, mientras que el aspecto de *logging* necesita la variable *history* y el método *logMessage*.

```
public aspect Authorization {

    private ArrayList <Subject> BankAccount.owners;

    private boolean BankAccount.isSubjectAuthorizedToOperate() {
        return owners.contains (history.activeSession.subject());
    }

    pointcut accountOperationExecution (BankAccount bankAccount) :
    execution (* BankAccount.* ..) && target (bankAccount);

    around (BankAccount bankAccount): accountOperationExecution
    (bankAccount) {
        if (bankAccount.isSubjectAuthorizedToOperate ()) {
            proceed(bankAccount);
        }
    }
}
```

```

}

public aspect Logging {
    private ActivityHistory BankAccount.history;

    private void BankAccount.logMessage(String message) {
        history.log.add (message);
    }

    pointcut accountOperationCall (BankAccount bankAccount) : call
(* BankAccount.* (..) && target (bankAccount);

    before      (BankAccount      bankAccount):accountOperationCall
(bankAccount) {
        Signature sig = thisJoinPointStaticPart.getSignature();
        bankAccount.logMessage(sig.getName() + " requested" );
    }
}

```

Figura 1-3. Declaración de los aspectos de autorización y logging

La modularización provista por los aspectos provee a las cuentas bancarias con la incumbencia de autorización y *logging*. Sin embargo, la funcionalidad base (Figura 1-1) y los aspectos (Figura 1-3) se definen en forma separada. Luego, utilizando los mecanismos de entretejido (*weaving*) de la AOP, se genera el programa expandido representado en Figura 1-2.

1.3 Características transversales en los requerimientos

Los requerimientos pueden incluir características transversales de la misma forma que lo poseen otros artefactos [Rashid 2003] [Rashid 2008]. Para producir una especificación de requerimientos robusta generalmente se necesita de una exploración del espacio de soluciones. Y a menudo, durante el proceso de análisis y especificación de requerimientos, aunque fuera implícito se toma una posición arquitectural. Esto puede definir que es y que no es un aspecto [Nuseibeh 2004]. Por lo tanto, los requerimientos pueden tener características enmarañadas y dispersas (*concerns tangled* y *scattered*) los cuales deben ser factorizados.

Por ejemplo, considere los siguientes requerimientos para una aplicación bancaria.

Req 1.- El sistema debe depositar dinero en una cuenta, verificando que el usuario está habilitado para realizar el deposito y dejando registro para auditar la operación.

Req 2.- El sistema debe extraer dinero de una cuenta, verificando que el usuario está habilitado para realizar la extracción y dejando registro para auditar la operación.

Req 3.- El sistema debe consultar el balance de una cuenta, verificando que el usuario está habilitado para realizar la consulta y dejando registro para auditar la operación.

Figura 1-4. Requerimientos para una aplicación bancaria.

Desde que cada requerimiento incluye información sobre la operación básica que modela como así también incluye información común a otros requerimientos, los mismos pueden ser reorganizados como muestra la siguiente figura con el fin de refactorizar la información.

Req 1.a.- El sistema debe depositar dinero en una cuenta.

- Req 2.a.- El sistema debe extraer dinero de una cuenta.
- Req 3.a.- El sistema debe consultar el balance de una cuenta.
- Req 4.- El sistema debe verificar que el usuario está habilitado para realizar la operación. Este requerimiento aplica a los req 1.a, 2.a, y 3.a.
- Req 5.- El sistema debe dejar registro para auditar la información. Este requerimiento aplica a los req 1.a, 2.a, y 3.a.

Figura 1-5. Requerimientos refactorizados de la aplicación bancaria

Los requerimientos 1.a, 2.a y 3.a son requerimientos base (*core*) y sólo especifican la funcionalidad elemental, mientras que los requerimientos comunes de autorización y *logging* están descritos en los requerimientos 4 y 5, los cuales son requerimientos transversales. Estos últimos dos requerimientos especifican que deben ser aplicados a los requerimientos previos.

Baniassad et al. [Baniassad 2006] identifican 4 etapas para lograr esta modularización: (i) identificación, (ii) captura, (iii) composición y (iv) análisis. Identificación implica determinar las características transversales. En el ejemplo previo consiste en identificar “*verificando que el usuario está habilitado para realizar...*” y “*dejando registro para auditar la operación*”. Baniassad et al. identifican 3 técnicas diferentes para realizar la identificación (técnicas que se describen debajo). La segunda etapa para lograr la modularización, captura, se basa en aislar los requerimientos transversales. En el ejemplo, se identificaron dos nuevos requerimientos, los requerimientos número 4 y 5. La etapa de composición consiste en vincular los requerimientos transversales con los requerimientos base (*core*). En el ejemplo esto implica determinar que ambos requerimientos nuevos deben ser aplicados a los requerimientos 1.a, 2.a y 3.a. La última etapa, la de análisis, consiste en estudiar las relaciones entre los requerimientos base (*core*) y los transversales para identificar conflictos e inconsistencias.

Baniassad et al. [Baniassad 2006] describen 3 técnicas para llevar a cabo la etapa de identificación:

- (i) Identificar términos aspectuales (*aspect terms*) los cuales representan atributos de calidad del software, también conocidos como requerimientos no funcionales (*non functional requirements*, NFR). Por ejemplo, Req 1, 2 y 3 muestran dos términos aspectuales (*aspect terms*). El primero “*verificando que el usuario está habilitado para realizar...*” el cual, se refiere a autorización (NFR relacionado con seguridad). El segundo, es “*dejando registro para auditar la operación*”, el cual contiene el término “auditar” (NFR con el mismo nombre).
- (ii) Identificar características dispersas (*scattered concerns*) los cuales son elementos que están repetidos en varios requerimientos. Por ejemplo, Req 1, 2 y 3, comparten la expresión “*verificando que el usuario está habilitado para realizar...*” y “*dejando registro para auditar la operación*”.
- (iii) Identificar requerimientos que impactan (*impact requirements*) los cuales son requerimientos que impactan, restringen o influyen otros requerimientos. Por ejemplo, Req 4 y 5 modularizan requerimientos que impactan en Req 1.a, 2.a y 3.a. Es importante mencionar que si se identifican requerimientos que impactan (*impact requirements*) no es necesario realizar las etapas de captura y composición ya que los requerimientos se encuentran organizados de tal forma que las tareas de las etapas no son necesarias.

1.4 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. 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 el LEL cada expresión se define a través de dos atributos: la noción y los impactos. 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.

Los símbolos en el LEL se agrupan en cuatro categorías básicas: sujeto, objeto, verbo y estado. Los sujetos se corresponden con elementos activos dentro del contexto de la aplicación, mientras que los objetos se corresponden con elementos pasivos. Por su parte, los verbos son las acciones que realizan los sujetos utilizando los objetos. Finalmente, los estados representan las situaciones en las que se pueden encontrar los sujetos o los objetos antes y después de realizar las acciones.

1.5 Resumen

Al principio de este capítulo se describió que el software es complejo por su naturaleza, es decir, por características que son propias del software y lo hacen diferente a otras obras de la ingeniería. Esta complejidad propia del software se traduce en complejidad en su construcción, ya que el proceso de construcción también posee características que lo hacen diferente a la construcción de otras obras de ingeniería. A nivel práctico, la construcción de software se percibe generalmente como un proceso con muchos problemas que hacen que no se cumplan con los tiempos, costos y calidades acordadas.

Con el fin de hacer frente a esta complejidad, la estrategia de *divide y vencerás* consiste en dividir el problema (sistema) en piezas más pequeñas y manejables. Sin embargo, esta descomposición jerárquica no alcanza para dividir el problema en subproblemas. La comunidad del AOSD muestra que hay elementos como por ejemplo atributos de calidad, los cuales no pueden ser modularizados siguiendo una descomposición jerárquica, sino que una descomposición de ese estilo haría que los atributos de calidad estén dispersos por varios módulos.

Es por ello que es necesario no sólo descomponer jerárquicamente un sistema, sino que es necesario identificar los aspectos y modularizarlos. Esta modularización permite una mejor comprensión del sistema lo cual es importante, puesto que los sistemas son elementos naturalmente complejos, por lo cual, cualquier elemento que favorezca a su comprensión es muy valiosa. Y esta modularización, permite también modificar el sistema de una forma más sencilla, puesto que logra una factorización que hace que los cambios sean localizados. Y por otro lado, también favorece el reuso al aislar una característica del software en un único módulo.

Sin embargo, si bien el diseño orientado a aspectos posee los beneficios de modificabilidad indicados, también se presentó en este capítulo los riesgos de introducir cambios en productos intermedios del proceso de desarrollo de software. Básicamente, cualquier cambio implica la modificación de todos los productos previos y posteriores a

los productos cambiados y estos cambios pueden generar errores. Por lo cual, sería deseable el identificar los aspectos lo antes posible y realizar el diseño inicial considerando los mismos y no introducirlos a mitad del desarrollo.

Hay muchas estrategias de identificación de aspectos en el código fuente. Generalmente automatizadas en herramientas que analizan el código e infieren que características pueden ser consideradas como características transversales. En la etapa de requerimientos la identificación no es tan inmediata y automática. El nivel de abstracción en la etapa de requerimientos es mucho más alto, por lo cual, es el ingeniero de requerimientos el que tiene que valerse de su experiencia y su criterio para determinar que características del sistema podrían ser consideradas características transversales. Esta tarea es humana intensiva y por ende propensa a errores o al menos sujeta al criterio de quien lo realiza.

Es por ello que es necesario contar con estrategias de identificación de características transversales del software que trabajen con los productos de las etapas iniciales del proceso de desarrollo de software. Y tal estrategia debe ser sistemática y con la menor subjetividad posible.

1.6 Contribución

En este trabajo de tesis presentamos 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). Debido a que el modelado se realiza tempranamente en el proceso de desarrollo del software, la estrategia propuesta detecta características transversales tan temprano como es posible y 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, lo cual redundará en beneficios para ser utilizado por todas las personas que participan del desarrollo de software.

1.6.1 Objetivo

El objetivo de esta tesis es desarrollar una estrategia de identificación de características transversales en una etapa temprana del desarrollo del software a partir del lenguaje del contexto de la aplicación capturado por el Léxico Extendido del Lenguaje.

1.6.2 Aportes

Esta tesis posee dos aportes significativos. En primer lugar provee una estrategia para identificar características transversales en una etapa muy temprana del proceso de desarrollo. En segundo lugar provee una herramienta más a las personas que utilizan el LEL, ya que sin costo adicional, con sólo construir el LEL pueden identificar características transversales.

1.7 Publicaciones asociadas a la tesis

“Using the problem domain language to specify navigational concerns in web applications”, Proceedings of the Workshop on Requirement Engineering, WER 2006, ISSN 1413-9014, Rio de Janeiro, Brasil, Julio 2006.

“Early identification of crosscutting concerns in the domain model guided by states”, Proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, ISBN:978-1-60558-639-7.

“Early identification of crosscutting concerns with the Language Extended Lexicon” Enviado para revision en Julio 2011 a Requirements Engineering Journal, en revisión desde Agosto 2011.

“Deriving requirements specifications from the application domain language captured by Language Extended Lexicon” enviado para revisión a Wer 2012, Buenos Aires, Argentina.

“TICCWL: A Tool That Identifies Crosscutting Concerns Through LEL Approach”, enviado para revisión a Wer 2012, Buenos Aires, Argentina.

“Assessing the effectiveness of the LEL approach for detecting crosscutting concerns” enviado para revision a Journal of Universal Computer Science.

1.8 Organización del documento

Este documento está organizado de la siguiente manera. El capítulo 2 hace una revisión de trabajos relacionados con la identificación, captura y modelización de características transversales. Luego, en el capítulo 3 se describe la estrategia de identificación propuesta. En ese capítulo se describe uno de los pilares de la estrategia, el LEL, se describe la estrategia en detalle y también se presentan dos casos de estudio para mostrar la aplicabilidad de la estrategia. El capítulo 4 reporta un experimento que se realizó con el fin de validar la efectividad de la estrategia. La sección 5 muestra una herramienta construida con el fin de asistir en la aplicación de la estrategia. Luego, el capítulo 6 presenta las conclusiones. Finalmente, se detallan las referencias en el capítulo 7 y se agrega un capítulo 8 con anexos que poseen información detallada de los casos de estudio y del experimento.

2 Revisión de 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 incluye una sección que describe técnicas de identificación de características transversales en el código fuente, puesto que por analogía se podría extrapolar las técnicas a una etapa temprana.

El capítulo se encuentra organizado de la siguiente manera. En primer lugar se enumeran los trabajos que se describen en esta sección, luego se describen los mismos y finalmente se presenta una conclusión.

2.1 Introducción

En primer lugar describimos el trabajo de Baniassad [Baniassad 2006] que presenta una forma de organizar los requerimientos descritos 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.

Luego, describimos otro trabajo de Baniassad [Baniassad 2004] el cual consiste en 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*.

Estos dos trabajos son relevantes para nuestra tesis puesto que nuestro objetivo es identificar características transversales y si bien es relevante el hecho de hacerlo desde lenguaje coloquial, nuestro objetivo es hacerlo con un producto previo a los requerimientos.

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 partir de ellos comienza el análisis. Es por este motivo que consideramos interesante el analizar y describir este trabajo.

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. A pesar de que trabaja sobre casos de uso (en particular los flujos de datos de los mismos), es interesante este trabajo por el hecho de no circunscribirse solamente a los verbos en el análisis.

Rashid [Rashid 2003] plantea un modelo con puntos de vista que agrupan requerimientos. Es interesante el hecho de agrupar los requerimientos en lugar de trabajar dentro de ellos como lo hace la técnica Theme/Doc por ejemplo. El agrupamiento lo realiza a través de puntos de vistas y luego identifican puntos de vista transversales que impactan en otros requerimientos. Es interesante el agrupamiento de requerimientos y el análisis realizado, por este motivo lo describimos en este capítulo.

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. Si bien las técnicas de identificación de características transversales son aplicadas en el código fuente, son de relevancia para esta tesis, puesto que por analogía se podrían extrapolar para realizar en una etapa temprana.

Además de los trabajos indicados previamente que son descriptos en las secciones siguientes, existen otros trabajos interesantes sobre el tema. Son trabajos que básicamente se ocupan de la identificación de características transversales, sin embargo los trabajos tratan sobre técnicas de análisis más que técnicas sistemáticas de identificación. Es por ello que los describimos en esta introducción, pero no las describimos en las próximas secciones.

Sousa [Sousa 2004] describe una estrategia de separación de características transversales a partir de los requerimientos y para llegar a los casos de uso. Es interesante como combina dos modelos ya existentes: por un lado utiliza el desarrollo dirigido por casos de uso (*use case driven development*) al que integra con un Framework de requerimientos no funcionales (*NFR framework*).

Yu [Yu 2004] plantea una estrategia para identificar aspectos a partir de los objetivos. El utiliza el *modelo v* para plasmar requerimientos funcionales y no funcionales y trabaja sobre este modelo analizándolo e identificando características transversales. Es interesante como trabaja con un modelo basado en objetivos combinando los requerimientos funcionales y no funcionales para identificar características transversales.

Finalmente, Silva [Silva 2005] presenta un lenguaje de modelado que utiliza conceptos de la programación orientada a aspectos, de manera de registrar el impacto que los requisitos realizan los unos sobre otros. El método propuesto aborda la separación y composición de características transversales. La separación es realizada por un modelado de características a través de un modelo de metas. Y el lenguaje es una extensión del *modelo v*. Este trabajo se basa principalmente en el modelado.

A continuación se describen con más detalle los trabajos indicados en esta sección.

2.2 Descubriendo aspectos tempranos

Baniassad et al. [Baniassad 2006] proponen una forma de organizar los requerimientos descriptos con texto coloquial, como así también presentan una serie de actividades para organizar los requerimientos descriptos tradicionalmente a la forma que ellos proponen.

Los autores hacen una analogía entre los documentos de requerimientos y el código fuente y sugieren que los requerimientos se organicen de la siguiente forma. Establecen que por un lado se especifiquen los requerimientos base (*core*), separados de los requerimientos *aspectuales*. Esto es análogo al código fuente en donde el código base se

ubica separado de los aspectos. Luego, tanto en el modelo de organización de requerimientos como en los aspectos, es necesario vincular de alguna forma ambas descripciones: base y *aspectuales*. Es por ello que Baniassad et al. determinan que es necesario especificar requerimientos que impactan (*impact requirements*), los cuales constituyen un vínculo entre los requerimientos *aspectuales* y los requerimientos base.

Además de proponer este modelo para especificar los requerimientos, los autores organizan la construcción del mismo a través de cuatro actividades. Ellos determinan que para obtener el modelo propuesto es necesario realizar: (i) identificación, (ii) captura, (iii) composición y finalmente (iv) análisis.

La actividad de identificación se refiere a analizar el conjunto de requerimientos para separar los requerimientos base de los requerimientos *aspectuales*. Esta selección se puede realizar siguiendo tres técnicas. Por un lado se puede realizar identificando términos *aspectuales* (*aspects terms*), también se puede realizar analizando requerimientos que impactan (*impact requirements*) y finalmente se pueden analizar características dispersas (*scattered concerns*).

Términos *aspectuales* (*aspects terms*) son términos o expresiones que se refieren a atributos de calidad, es decir requerimientos no funcionales. Por ejemplo, cuestiones de seguridad, de performance, etc... son atributos de calidad bien conocidos y cualquier palabra o expresión que haga referencia a alguno de ellos puede ser reconocida automáticamente por estrategias de minería de datos (*data mining*) que se describen en otros trabajos relacionados, por lo cual, haciendo uso de tales técnicas y herramientas que la soportan, puede automáticamente detectarse.

Otra forma de seleccionar a los requerimientos base de los *aspectuales* consiste en analizar requerimientos que impactan (*impact requirements*). Esto consiste en analizar la relación entre los distintos requerimientos. Es decir, podría existir varios requerimientos que de alguna forma mencionan otro requerimiento, esto muestra un vínculo entre esos varios requerimientos para con el otro. Este vínculo es la pista de que existe comportamiento *aspectual* que puede ser detectado por técnicas tales como Theme/Doc por ejemplo que se describen en otra sección.

La última forma de seleccionar los requerimientos *base* de los *aspectuales* consiste en identificar las características dispersas (*scattered concerns*). Este análisis consiste en revisar los requerimientos buscando conceptos o comportamiento que se repita en muchos requerimientos. Este hecho es una analogía bien directa de lo que se realiza en el código fuente para identificar código *aspectual*. En los requerimientos también puede suceder ya que generalmente se obtiene y especifica la información en forma desordenada y redundante y es necesario un análisis y reorganización de los mismos para poder eliminar esta redundancia.

Luego de identificados los requerimientos *aspectuales* por alguna o varias de las técnicas indicadas anteriormente, la siguiente actividad consiste en realizar la captura de estos requerimientos *aspectuales*. Esto consiste en reorganizar los requerimientos, de forma tal de aislar los requerimientos *aspectuales* de los requerimientos base, es decir, reorganizar los requerimientos de forma de escribirlos por separado, para evitar la redundancia y aumentar la modularidad de los mismos.

Una vez realizada la captura, se cuentan con los dos grupos de requerimientos por separado, sin embargo, los requerimientos *aspectuales* se caracterizan por influir en

más de un requerimientos base es por ello que es necesario realizar la composición de los mismos. Y esto consiste en explícitamente indicar que requerimientos *aspectuales* se vincula con que requerimientos base. Una vez realizada esta composición, se produce un modelo de requerimientos que describe los mismos requerimientos que le modelo inicial, sólo que el mismo se encuentra reorganizado con los 3 grupos que plantean los autores: requerimientos base, requerimientos *aspectuales* y requerimientos *que impactan*.

Una vez obtenido este modelo que reorganiza la información, es necesario realizar la actividad de análisis sobre el mismo. Con esta reorganización es mucho más sencillo entender los requerimientos y poder realizar un análisis sobre los mismos y a la vez detectar conflictos e inconsistencias.

2.3 Theme / DOC

La estrategia Theme/Doc [Baniassad 2004] provee vistas de las especificaciones textuales, exponiendo las relaciones entre las distintas piezas del sistema con comportamiento. Estas vistas asisten al desarrollador a determinar cuales elementos de funcionalidad son aspectos y cuales son base.

Theme/Doc está basado en la noción de *theme*, el cual representa un rasgo (*feature*) del sistema. Múltiples *themes* se pueden combinar para conformar toda la funcionalidad del sistema. Hay dos tipos de *themes*: *themes* base, los cuales pueden compartir algunas estructuras y comportamiento con otros *themes* base y *themes* transversales los cuales poseen comportamiento que se superpone con funcionalidad de los *themes* base.

Theme/Doc opera con la premisa básica de que si dos comportamientos se describen en el mismo requerimiento, los mismos están relacionados. La estrategia consta de 4 etapas: (i) identificar acciones y entidades, (ii) categorizar las acciones en *themes*, (iii) identificar *themes* transversales y (iv) ver *themes* individuales.

Identificar acciones y entidades consiste en procesar las expresiones que describen los requerimientos del sistema. A partir de ciertas claves provistas por el desarrollador, la herramienta identifica acciones claves (*key-actions*) como así también entidades claves (*key-entities*).

El categorizar las acciones en *themes* se ocupa de analizar las distintas acciones para determinar cual de ellas merecen ser descriptas como *themes*, ya que no todas las acciones se van a diseñar como *features* independientes del sistema, incluso, algunas acciones, se diseñaran como sub-acciones de otras. Por lo tanto, la etapa de categorizar las acciones en *themes* consiste en identificar las *features* que tienen la relevancia necesaria para ser modeladas como *themes*. En esta etapa, se construye la vista de acciones (*action-view*). La misma es un diagrama compuesto por dos elementos: acciones y requerimientos. Las acciones se modelan con rombos y los requerimientos con rectángulos de puntas redondeadas. Si una acción es mencionada en un requerimiento, se debe trazar una línea entre la acción (rombo) y los requerimientos (rectángulo). Con esta vista, se debe analizar y determinar cuales acciones deben ser *themes* o simplemente comportamiento dentro de los *themes*. El definir que acciones son lo suficientemente importantes para se consideradas *themes* es un proceso altamente intuitivo.

La identificación de *themes* transversales se realiza a partir de la vista de acciones principal (*major action view*), el cual es un diagrama similar al vista de acciones, sólo que está formado por las acciones más relevantes identificadas en el paso previo. Básicamente, este modelo busca ilustrar los requerimientos que son compartidos por más de un *theme*. Si un requerimiento es compartido por dos o más *themes*, es una pauta de que pudimos haber identificado un aspecto en nuestro sistema. Esto se debe a que dos *themes* no pueden operar sin que cada uno de ellos se apoye en el comportamiento del otro. Sin embargo, antes de determinar que se encontró un aspecto hay que asegurarse de que no encontramos un requerimiento vagamente descripto que está enmascarando varios requerimientos. Por lo tanto, es necesario verificar si el requerimiento no puede ser rescrito en varios requerimientos distintos los cuales referencien solamente a un *theme* cada uno. Si no es posible la reescritura de requerimientos que provea una relación 1 a 1 entre requerimientos y *themes*, definitivamente encontramos un aspecto.

La última etapa consiste en analizar los *themes* en forma individual. En este punto los *themes* son acciones relevantes las cuales se deben analizar en relación a los requerimientos asociados con las acciones principales como así también con las acciones menores. Las entidades claves también se muestran como cuadrados. Esta vista se utiliza para verificar que las relaciones que permitieron la identificación se realizaron correctamente.

2.4 EA-Miner

Sampaio et al. [Sampaio 2005] [Sampaio 2005b] proponen una herramienta que provee soporte automatizado para identificar y separar *características aspectuales* de los *no aspectuales*, como así también provee soporte para la identificación de las relaciones que los requerimientos proveen de ellos. Específicamente los autores hablan de identificar características *base*, aspectos tempranos (*early aspects*) y relaciones transversales entre los características base y los aspectos tempranos.

La herramienta es configurable, por lo cual, las características base pueden tomar distintas formas en función de la técnica que se utilice. Por ejemplo, si el análisis de descomposición es a través de un modelo orientado a puntos de vista (*viewpoints*), una característica base es un *viewpoint*. Mientras que si la técnica usada está basada en UML, un característica base podría ser un caso de uso (*use case*). Por su parte, los aspectos tempranos son las entidades que semánticamente atraviesan a los concerns base.

EA-Miner utiliza técnicas de procesamiento del lenguaje natural como por ejemplo: parte del discurso (*part-of-speech*, *POS*) y marcación semántica (*semantic tagging*), concordancia y frecuencia de palabras (*word frequencies and concordances*), etc. Todo esto lo hace a través de la herramienta WMatrix.

Los autores sostienen que en la etapa de requerimientos hay grandes volúmenes de información que se debe analizar, por lo cual, ellos sostienen que la mejor forma de analizarla para poder identificar características transversales es a través de herramientas que automaticen esta tarea. Sin embargo, ellos aclaran que estos *aspectos* identificados automáticamente son potenciales aspectos, los cuales deben ser validados por algún experto con criterio.

Por otro lado, los autores también resaltan el hecho de que la estrategia utiliza documentos de requerimientos sin ningún tipo de estructura para identificar los aspectos.

La estrategia de detección consta de las siguientes etapas. En la actividad de identificación, EA-Miner *parsea* los archivos de diferentes formatos y estructuras provistos por el ingeniero de requerimientos y se los envía a WMatrix para que realice el análisis de parte del discurso y marcación semántica. WMatrix devuelve el archivo procesado en el cual incluye anotaciones a partir de los análisis realizados. Luego, EA-Miner a partir de la información incluida por WMatrix produce un modelo basado en puntos de vista (*viewpoints*), casos de uso, etc. según la configuración que realizó el usuario. En este punto se desarrolla una nueva actividad, la de presentación de resultados. Para luego, refinar el modelo con la siguiente actividad, la de descarte (*screening out*) y generación de especificación, en donde el modelo producido es refinado por parte del usuario a través de la eliminación de entidades irrelevantes o agregando nuevas. En el caso del agregado de nuevas entidades, la herramienta provee ayuda para generar la sección correspondiente del documento de requerimientos.

El punto más sensible de esta estrategia es el preprocesamiento que realiza WMatrix el cual consiste en realizar el análisis parte del discurso en donde se identifican los sustantivos y verbos del texto. Luego, WMatrix también realiza un análisis semántico el cual consiste en relacionar palabras y expresiones de varias palabras con conceptos.

El análisis que EA-Miner realiza está guiado por el preprocesamiento que realiza WMatrix. Para el caso del modelado en puntos de vista (*viewpoints*) por ejemplo, EA-Miner busca sustantivos y reúne a todos los requerimientos en los que está presente el sustantivo para determinar que es un características base. La herramienta adolece del inconveniente de no detectar que la misma palabra puede tener pequeñas variaciones (por ejemplo del singular al plural). Luego, la herramienta se ayuda de un catálogo de requerimientos no funcionales para detectar que palabras de los documentos son aspectos tempranos. Luego, las relaciones transversales las determina, al identificar las relaciones entre las características base y los aspectos tempranos. Esto lo determina por simple intersección entre ambas entidades.

2.5 Análisis semántico para la identificación de aspectos

Rago et al. [Rago 2009] indican que la mayoría de los análisis de identificación de aspectos se basan en las búsquedas de verbos para identificar comportamiento transversal. Sin embargo, ellos consideran que sería más adecuado y preciso identificar sobre que objetos actúan estas acciones o verbos, para aumentar la efectividad del enfoque. Esto es debido a que una misma acción o verbo al ser aplicada sobre diferentes objetos, puede llegar a considerarse como diferentes comportamientos.

De todas formas, como en el texto natural no siempre se incluye información sobre el objeto al que afecta el verbo, es necesario también considerar los verbos solos (sin el objeto directo) para no perder información. Por otro lado, la estrategia propuesta por los autores considera el uso de UML como base de la aplicación para identificar aspectos.

Básicamente la estrategia consiste de los siguientes pasos. En primer lugar se realiza un análisis léxico, sintáctico y semántico del texto de las especificaciones, así como también un estudio estadístico y de ubicación de ese texto. Básicamente consiste en un etiquetado del discurso (usando un procesador de lenguaje natural) y la desambiguación

semántica de las palabras. Esto se realiza en primer lugar sobre los textos de los flujos básicos y alternativos de los casos de uso. Luego, se realiza sobre los requerimientos no funcionales del sistema y sobre los requerimientos adicionales de cada una de las especificaciones.

Con la información previa se construye un grafo que muestra la utilización de nodos que pueden ser verbos como así también objetos directos y que son utilizados por otros nodos verbos y objetos directos. Esta utilización no se realiza directamente de nodos a nodos, sino que se agrega un nodo intermedio que agrupa pares de verbos semánticamente relacionados y de sustantivos semánticamente relacionados, y la relación se realiza a través de ellos.

Luego de la creación del grafo, se realiza un simple recorrido del mismo para determinar aspectos candidatos ordenándolos según su relevancia. La identificación de los aspectos candidatos se realiza recorriendo el grafo y contando aquellos grupos de verbos que cortan transversalmente una cantidad de casos de uso que superan un umbral predeterminado. Para cada grupo de verbos que es considerado transversal, se realiza una exploración en busca del término (un verbo) que es más representativo del grupo, para proponerlo como nombre candidato de la característica (*concern*) detectada. Adicionalmente, se establece si la característica (*concern*) es funcional o no funcional, analizando de donde provienen los verbos. Una vez identificados las características transversales (*crosscutting concerns*), se establece el ranking entre los mismos. Para ello se toman varios parámetros: el número de casos de uso cortados transversalmente, el número de cortes transversales totales del grupo de verbos, la relevancia de ocurrencia del grupo de verbos, la relevancia de ocurrencia de los grupos de objetos directos relacionados con el grupo de verbos en cuestión y por último si el grupo de verbos es considerado funcional o no funcional.

2.6 Modularización y composición de requerimientos aspectuales

Rashid et al. [Rashid 2003] proponen un enfoque para modularizar funcionalidad base (modelizada a través de puntos de vista que agrupan un conjunto de requerimientos) y características transversales (*crosscutting concerns*) y componer los mismos. Los mismos autores plantearon un modelo previo en el cual mantenían relaciones de composición dentro de la definición de los aspectos la cual era de una granularidad muy gruesa, por lo cual, sólo era posible modelar como un aspecto afectaba a un conjunto de puntos de vista (*viewpoints*). Sin embargo, en este modelo inicial, no se podía modelar la influencia de un aspecto a un requerimiento específico dentro del *viewpoint* afectado.

El nuevo modelo propuesto consiste en comenzar identificando y especificando las características (*concerns*) y los requerimientos de los interesados (*stakeholders*). Es muy útil el relacionar las características (*concerns*) a los requerimientos a través de una matriz. Luego, observando la matriz es posible identificar rápidamente que característica (*concern*) atraviesa transversalmente a los requerimientos de los interesados (*stakeholders*).

Luego de que la relación de grano grueso entre las características (*concerns*) y los requerimientos que los interesados (*stakeholders*) se estableció, y de que los aspectos candidatos se identificaron, el siguiente paso es definir detalladamente las reglas de composición entre las características (*concerns*) y los requerimientos para definir las

relaciones entre ellos en un grano más fino. Ya que estas reglas operan a una granularidad de requerimientos individuales y no sólo a nivel de los módulos que las agrupa, las reglas de composición implican no solamente el vincular los aspectos y los requerimientos, sino que requieren de acciones y operadores, los cuales se utilizan para describir restricciones (*constraints*) y logros (*outcome*). Tanto las acciones como los operadores son informales, sin embargo, deben poseer un significado y una semántica bien clara. Algunos ejemplos de acciones relacionadas con las restricciones (*constraints*) son: imponer (*enforce*), asegurar (*ensure*), proveer (*provide*), aplicar (*apply*), etc. Luego, algunos ejemplos de operadores de restricción (*constraint*) son: durante (*during*), entre (*between*), por (*for*), etc. Y finalmente, acciones de logro (*outcome action*) son: satisfecho (*satisfied*) y realizado (*fulfilled*).

Puede suceder que ocurran conflictos entre diferentes aspectos porque restringen al mismo requerimiento. Es por ello que en este punto es necesario realizar un balance entre los aspectos. Se debe realizar una identificación y resolución de conflictos. Cada aspecto puede contribuir negativa o positivamente a los otros. Incluso, puede suceder que la contribución sea del tipo “no interesa”, esto es, cuando no se especifica. Para los aspectos que contribuyen negativamente es necesario especificar un valor real entre 0 y 1 y representa la prioridad de un aspecto en relación al conjunto de requerimientos del interesado (*stakeholder*).

2.7 Clasificación comparativa de estrategias de minería de aspectos

Bounour et al. [Bounour 2006] presentan una clasificación de estrategias de minería de aspectos. Algunas técnicas son basadas en análisis estático del código fuente, mientras otras se basan en el flujo de ejecución. Bounour identifica una categoría que llama detección de clones (*clone detection*) en la cual, las estrategias que incluye en la misma conducen un análisis léxico de la estructura de la información. Dentro de este grupo, existe una estrategia que consiste en construir un grafo de dependencias del programa (*program dependence graph, PDG*) que contiene información de naturaleza semántica como el control y el flujo de datos de un programa. Los links entre los nodos en el PDG representan el flujo de ejecución de las acciones.

Bounour et al. también mencionan una técnica la cual combina un árbol de sintaxis abstracta (*abstract syntax tree, AST*) con representación con *token* del código fuente. Esta técnica usa *parsers* para obtener una representación sintáctica del código fuente (*AST*). Luego, el algoritmo de detección de clones busca subárboles similares en el *AST*. Esta técnica tiene como limitación el que sólo logre identificar características (*concerns*) homogéneas, dado que la estrategia se basa en detectar patrones similares en el árbol y pequeñas diferencias entre las partes del árbol hacen imposible detectar el aspecto.

Bounour et al. también mencionan un estrategia denominada análisis de conceptos formales (*formal concept analysis, FCA*). La misma se basa en encontrar agrupamiento significativos de los elementos que poseen propiedades comunes. Con el fin de lograr este agrupamiento, varios *uses cases* son ejecutados y dos restricciones deben ser analizadas para identificar a los conceptos candidatos. La primera restricción establece que los atributos de los conceptos deben pertenecer a más de una clase. La segunda establece que los diferentes métodos de la misma clase deben estar contenidos en más de un caso de uso.

Finalmente, Bounour et al. también describen una metodología basada en dispersión (*scattering*) llamada análisis de fundido (*fade in análisis*) la cual consiste en hacer minería del código fuente para identificar síntomas de dispersión (*scattering*) de código. Esta técnica establece que un método con una gran cantidad de llamados distribuidos debe ser considerada una característica (*concern*) dado que el método implementa una funcionalidad transversal. La técnica considera que la cantidad de llamadas (*fan-in*) es una buena medida de la relevancia y de la dispersión (*scattering*) de la característica (*concern*) descubierta.

2.8 Conclusiones

El presente capítulo describió principalmente técnicas para identificar características transversales en etapas tempranas del desarrollo de software. Varias de las técnicas se basan en requerimientos o casos de uso, aunque hay otras que se ocupan de analizar productos previos a ellos. Hay un consenso generalizado en considerar a los verbos como candidatos a características transversales, sin embargo, algunos trabajos utilizan objetos por ejemplo. Luego, el agrupamiento de los elementos a analizar es fundamental para descubrir como se entrecruzan las características. A continuación se enumera para cada uno de los trabajos descritos en las secciones previas y los puntos débiles que deberían ser considerados por nuestra estrategia.

Respecto del trabajo de [Baniassad 2006] los autores proponen una serie de actividades para factorizar los requerimientos y proveen una guía o categorización sobre como identificar esta información repetida, sin embargo, no proveen ningún método concreto y específico.

Respecto del trabajo de [Baniassad 2004] en este caso se provee un método concreto y específico para la identificación de características transversales, sin embargo, la estrategia depende de la subjetividad del profesional que en algún momento analiza las acciones para determinar que acciones merecen ser consideradas como *themes*. Incluso, hay ciertas palabras claves que deben ser provistas para comenzar el análisis.

Respecto del trabajo de Sampaio [Sampaio 2005] la estrategia requiere de mucha intervención humana, puesto que si bien propone una herramienta que automatiza la detección, hay mucha intervención humana para analizar los resultados parciales y que la herramienta continúe el proceso.

Respecto del trabajo de Rago [Rago 2009] si bien es interesante el considerar objetos además de verbos, el análisis se basa en casos de usos ya escritos, por lo cual, si bien es una momento temprano en el desarrollo, al llegar a los casos de uso ya se dedicó bastante esfuerzo para construirlos.

Respecto de la técnica de Rashid [Rashid 2003] si bien es interesante y no muy común el hecho de agrupar requerimientos en puntos de vista, la técnica es subjetiva dependiendo en gran medida del profesional que la aplica.

Básicamente, los elementos a disminuir son la subjetividad y realizar el análisis en los productos más tempranos como sea posible.

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

Este capítulo presenta la estrategia de identificación de características transversales junto con un par de casos de estudio que muestra la aplicabilidad de la estrategia. En otro capítulo se describe un experimento el cual demuestra la efectividad.

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

El capítulo está organizado de la siguiente manera. En primer lugar se presenta el Léxico Extendido del Lenguaje la estrategia utilizada para modelar el lenguaje del contexto de la aplicación. Luego se presenta la estrategia de identificación de características transversales. Finalmente se presentan dos casos de estudios.

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.

Existen dos principios que se deben seguir al describir símbolos: el principio de circularidad (también llamado principio de cierre o clausura) y el principio de vocabulario mínimo. El principio de circularidad afirma que durante la descripción de los símbolos se debe maximizar el uso de otros símbolos descriptos en el LEL. Por su parte, el principio de vocabulario mínimo complementa el principio de circularidad y afirma que en las descripciones se debe minimizar el uso de símbolos externos al LEL y los que se utilicen deben tener una definición clara, unívoca y no ambigua.

Ambos principios son vitalmente importantes para obtener un LEL autocontenido y altamente conectado, características que redundan en beneficios para comprender el lenguaje de la aplicación y características que hacen que el LEL pueda ser visto como

un grafo, el cual, al contener nodos con información textual no es otra cosa más que un hipertexto.

Como se indicó previamente cada símbolo se describe a través de dos atributos: la noción y los impactos. La noción describe las características, mientras que los impactos describen un segundo valor por asociación con otros símbolos. Esta descripción de los símbolos es una descripción general, sin embargo, los símbolos se deben categorizar en una de cuatro categorías básicas con el fin de especializar la descripción de los atributos. En principio hay 3 ontologías básicas que permiten modelar el mundo [Breitman 2003]. Ellas son: entidades, actividades y afirmaciones. Con estos 3 elementos se puede describir al mundo, puesto que permiten modelar las cosas, las actividades con las que interactúan las cosas y finalmente las afirmaciones o verdades que las cosas o actividades poseen o persiguen.

Para el LEL se determinan cuatro categorías básicas que son una extensión de las tres ontologías. Las cuatro categorías básicas son: sujeto, objeto, verbo y estado. Tanto los sujetos como los objetos son una especialización de las entidades. Luego, las actividades se corresponden con los verbos. Y finalmente las afirmaciones se corresponden con estados. Los sujetos se corresponden con elementos activos dentro del contexto de la aplicación, mientras que los objetos se corresponden con elementos pasivos. Por su parte, los verbos son las acciones que realizan los sujetos utilizando los objetos. Finalmente, los estados representan las situaciones en las que se pueden encontrar los sujetos o los objetos previo y luego de realizar las acciones.

A partir de la caracterización presentada de las 4 categorías y a la descripción general de los símbolos, se desprende que la noción de un sujeto debe describir las características que cierto sujeto debe cumplir para calificar como tal, mientras que los impactos describen las acciones que el sujeto realiza. Luego, el objeto debe tener como noción los atributos que caracterizan o conforman al objeto, mientras que los impactos describen las acciones que los sujetos realizan sobre ellos. Por su parte, los verbos describen en la noción la razón de ser de la misma, es decir, el objetivo que persigue la acción, mientras que los impactos describen la descomposición de pasos para lograr el objetivo. Finalmente, los estados describen en su noción, la situación que describe el estado, mientras que los impactos describen las transiciones hacia otros estados. La Tabla 3-1 resume las cuatro categorías de símbolos y como describirlos.

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 [Wood 1997] describe que los conceptos en la memoria humana son asociados unos con otros y la experiencia de recordar consiste en recorrer estas asociaciones, condición que está relacionada con el principio de circularidad y vocabulario mínimo. Por otro lado, Wood sostiene que el conocimiento experto está generalmente organizado en jerarquías. En un macro nivel se presentan organizaciones taxonómicas, mientras que en un micro nivel el conocimiento experto se liga a patrones de acciones. Wood denomina a los elementos del macro nivel objetos y a los del micro nivel procesos. El conocimiento de objetos incluye las entidades (concretas y abstractas) de un dominio particular con sus variadas categorías y relaciones. Relaciones taxonómicas como tipos, subtipos y rasgos identificatorios o características de objetos son particularmente importantes. El conocimiento de procesos es el conocimiento requerido para cumplir con un trabajo utilizando los objetos relevantes. Claramente, el

nivel macro tiene relación con los símbolos de categoría sujeto y objeto del LEL, mientras que el nivel micro tiene relación con la categoría verbo.

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

La utilidad del LEL como modelo ha sido observada en los trabajos de Breitman [Breitman 2003] y Gruber [Gruber 1993]. Breitman describe un proceso para estudiar el LEL con el fin de producir una ontología, por lo cual, ella estudia el lenguaje del contexto de la aplicación y a partir del mismo construye una ontología. Gruber por su parte define ontología como “una especificación explícita y formal de una conceptualización compartida”. Si esta conceptualización compartida mencionada por Gruber es una conceptualización compartida sobre el mundo real, lo será sobre la aplicación con la cual se está trabajando y por lo tanto puede ser considerada como modelo de la misma. Por lo tanto, con el LEL se captura el lenguaje del contexto de la aplicación y a partir de él se construye una ontología la cual puede ser considerada como modelo de la aplicación, lo que en definitiva nos lleva a que el mismo LEL puede ser considerado como un modelo de la aplicación. Y esto se indica explícitamente en [Breitman 2003] ya que se menciona que “la filosofía subyacente al LEL cae en la categoría de contextualización, de acuerdo a la cual, las particularidades de un contexto de uso de aplicación deben ser comprendidas en detalle antes de que se puedan producir los requerimientos”.

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”.

La calidad del modelo depende del esfuerzo invertido en la construcción como así también en la capacidad de representación. En el párrafo anterior se mostró que hay experiencias que justifican la efectividad del LEL como instrumento para capturar conocimiento y lo más importante aún, es que es una herramienta adecuada para la comunicación del conocimiento capturado. Durante el desarrollo participan muchos actores que poseen diferentes capacidades y habilidades, y dado que el LEL utiliza lenguaje natural, es una herramienta apropiada para expertos, ingenieros de requerimientos y desarrolladores. Y esto es de gran relevancia al contribuir positivamente en los esfuerzos necesarios para realizar validación de estos modelos. Kaplan et al. [Kaplan 2000] han observado que la simplicidad de construcción y organización previstas por el LEL es un factor clave en la validación con los usuarios.

El proceso de construcción de LEL [Leite 1993] [Breitman 2003] se puede resumir en las siguientes etapas: (i) identificación de los símbolos, (ii) categorización de los símbolos identificados, (iii) descripción de los símbolos en función de la categoría definida, (iv) identificación de sinónimos a partir de las descripciones, (v) control de las definiciones y especialización de las categorías y finalmente (vi) validación con los interesados (*stakeholders*).

La identificación de símbolos comienza por determinar las fuentes a partir de las cuales se va a estudiar el lenguaje de la aplicación. Básicamente es posible realizarlo a partir de texto escrito como podría ser documentación relacionada con un sistema o también se podría realizar a partir de interacciones con los interesados (*stakeholders*) como podrían ser entrevistas. Dado que el objetivo del LEL es capturar el lenguaje, es necesario identificar los símbolos más representativos para definirlos en el LEL. La lengua escrita y la oral son muy distintas. Al expresarse en forma escrita se suele ser más preciso y se suelen utilizar muchos sinónimos con el fin de mejorar la redacción. De todas formas, los términos específicos característicos de un dominio se suelen utilizar sin recurrir a sinónimos, por lo cual los símbolos candidatos a identificar son aquellos que más se repiten, los que se encuentran enfatizados (con negrita, cursiva o subrayados), aquellos que se ubican en los títulos, descripción de figuras o tablas. En la expresión oral se suele ser más desorganizado y se suele repetir términos. En este caso, aquellos términos más nombrados y en los cuales se enfatiza cambiando el tono de voz por ejemplo, son aquellos a los cuales hay que prestarle atención para considerarlos.

Una vez identificados los símbolos y previo a la descripción de los mismos es necesario categorizarlos, es decir, se debe determinar si el símbolo es sujeto, objeto, verbo o estado. Es necesario esta categorización, puesto a partir de ella se determina de que forma (es decir, con que información) se debe caracterizar (describir) cada uno de los símbolos. Esta categorización no es definitiva, sino que es una categorización preliminar y luego en la descripción puede surgir que la categoría determinada no era la correcta.

La descripción de los símbolos implica escribir la noción y los impactos de cada símbolo, de forma tal que los atributos se ajusten a la plantilla (*template*) establecido según la categoría a la que pertenece el símbolo. Es importante tener presente los principios de circularidad y de vocabulario mínimo durante esta fase, para cumplir con

los mismos. Por otra parte, puede suceder que la descripción de los símbolos no surja del conocimiento que el analista recopiló, sino que necesita recurrir a las fuentes para completar la descripción. En este caso, podría suceder que aparezcan nuevos términos que necesiten ser definidos.

Una vez descritos los símbolos es posible identificar sinónimos. Podría suceder que al identificar los símbolos no se haya reparado en que dos símbolos eran sinónimos por el vago conocimiento que se poseía sobre el dominio. Sin embargo, luego de describirlos es el momento para verificar si dos símbolos independientes poseen descripciones similares. En ese caso, se deben unir ambas descripciones en una sola y ambos términos se utilizan como sinónimos que identifican al concepto.

La identificación de sinónimos es en cierta forma un control del LEL por parte del analista, puesto que consiste en una revisión de los símbolos para factorizarlos. Sin embargo, no es el único control que es posible hacer. Tal vez se descubre que un concepto es muy amplio, por lo cual, se podría definir un concepto genérico y todas las especializaciones que fueran necesarias. También podría suceder que se utilizan términos para describir otros, y estos términos que se utilizan merecen ser definidos. Además, podría surgir la necesidad de especializar la categorización básica de símbolos. Por ejemplo, podría suceder que el dominio con el cual se trabaja merece describir una gran cantidad de sujetos y fuera necesario determinar distintas especializaciones de sujetos, porque habría que describir símbolos que se corresponden con estas subcategorías. Esta necesidad debe surgir en el control y deben crearse las subcategorías con la plantilla (*template*) que deben cumplir los símbolos de estas nuevas subcategorías y también debe adecuarse la descripción de los símbolos con las nuevas subcategorías.

Finalmente, la última etapa en la construcción del LEL es la validación del LEL obtenido. Claramente es la fase más costosa y sensible, y en cierta forma es utópica, dada la necesidad de validarlo con las mismas fuentes que permitieron su construcción y que no siempre están disponibles. Sin embargo, siempre es posible solicitar a algún interesado (*stakeholder*) que realice alguna revisión exploratoria con el fin de validar el producto obtenido.

Si bien todas las fases de construcción del LEL requieren un esfuerzo importante, el hecho de describir en símbolos el conocimiento adquirido es una etapa con cierta criticidad. Para ello existe una herramienta que ofrece ciertas bondades y facilidades: C&L [C&L 2009] [Leite 2005].

A continuación se va a mostrar un ejemplo de LEL para un dominio bancario. El ejemplo es bien simple. Existe un banco el cual permite que sus clientes operen con cuentas. Las operaciones posibles a realizar con las cuentas son tres: es posible depositar dinero, extraer dinero y consultar el saldo. Si bien estas operaciones son sencillas el banco necesita proveer cierto nivel de seguridad en las mismas, es por ello, que ante cada una de estas tres operaciones, el banco debe verificar la identidad del cliente. Luego de verificada la identidad y de realizada la operación, el banco mantiene un registro de la operación para posterior revisiones de auditoría si fuera necesario. Finalmente, además de las operaciones para operar con una cuenta, existen las operaciones necesarias para abrir una cuenta. La apertura de una cuenta no es inmediata, sino que hay distintos pasos que se deben seguir hasta que la cuenta esta finalmente operativa. En primer lugar, el cliente solicita la apertura de una cuenta, por lo cual, el primer estado por el que pasa una cuenta (en realidad no es una cuenta porque en esta

instancia no está creada), es el de *solicitada*. Luego, en algún momento entre todas las tareas que tiene que realizar el banco, procesa la solicitud de creación de cuenta, y efectivamente crea una cuenta, sin embargo, esta cuenta creada no es disponible para ser operada por el cliente. Esta cuenta se encuentra *bloqueada* hasta que ciertas verificaciones de deudas y solvencias sean realizadas. Finalmente, luego de realizadas estas verificaciones, la cuenta está finalmente *activada*.

La siguiente tabla muestra los símbolos que deben ser descriptos para capturar la esencia del contexto de la aplicación. La tabla muestra los símbolos agrupados por categorías.

Tabla 3-2. Símbolos correspondientes a la aplicación bancaria

Categoría	Símbolos
Sujetos	Banco Cliente
Objetos	Cuenta Log
Verbo	Abrir una cuenta Activar una cuenta Registrar Corroborar Operar Depositar Extraer Consultar saldo
Estados	Solicitada Bloqueada Activada

Los siguientes ejemplos proveen la descripción de un símbolo de cada una de las categorías con el fin de mostrar como se ajustan las descripciones de las nociones y los impactos a la plantilla (*template*) definida. Las descripciones de los símbolos poseen palabras subrayadas, estas palabras son expresiones que también están definidas en el LEL. Simbolizan un especie de link para mostrar que puede ser explorada la definición de otras palabras. Esta definición cíclica está alentada por el principio de circularidad. Los ejemplos son los siguientes. El primer símbolo es *cliente* el cual es un sujeto, por lo cual la noción describe quien es el *cliente*, mientras que los impactos describen las acciones que el *cliente* puede realizar con su *cuenta*.

Sujeto: cliente
Noción
 Persona que abre una cuenta en el banco.
Impactos
 El cliente puede depositar dinero en su cuenta.
 El cliente puede extraer dinero de su cuenta.
 El cliente puede consultar el saldo de su cuenta.

Figura 3-1. Descripción del símbolo *cliente*.

El próximo símbolo es *cuenta*, el cual es un objeto, por lo cual, la noción describe los atributos que posee la cuenta, mientras que los impactos describen las operaciones que el *cliente* puede realizar con su *cuenta*.

Objeto: cuenta
Noción

La cuenta posee un saldo y el cliente puede operarla.

Impactos

El cliente puede depositar dinero en su cuenta.

El cliente puede extraer dinero de su cuenta.

El cliente puede consultar el saldo de su cuenta.

Figura 3-2. Descripción del símbolo *cuenta*.

El tercer símbolo es el verbo *extraer*, por lo cual la noción describe el objetivo de la acción, mientras que los impactos describen cada uno de los pasos los cuales son necesarios para realizar la extracción.

Verbs: extraer

Noción

Acción de tomar dinero desde la cuenta.

Impactos

El banco debe corroborar que la persona quien opera es el propietario de la cuenta.

El banco debe verificar que la cuenta posee suficiente dinero para realizar la extracción.

El banco debe verificar que el propietario de la cuenta no ha realizado más extracciones que la cantidad permitida.

El banco registra la operación en un log.

Figura 3-3. Descripción del símbolo *extraer*.

Finalmente, el símbolo *solicitada* es un estado, de modo que posee una noción la cual describe la noción, mientras que los impactos describen como una *cuenta solicitada* se convierte en una cuenta *bloqueada*.

Estado: Solicitada

Noción

Situación en la cual el cliente ha decidido abrir una cuenta y el banco esta listo para abrirla.

Impactos

El banco abre la cuenta y el cliente es el propietario de una cuenta bloqueada.

Figura 3-4. Descripción del símbolo *solicitada*.

Finalmente, con el fin de mostrar como se consideró el principio de circularidad en la construcción de este ejemplo, presentamos una tabla (Tabla 3-3) la cual contiene a todos los símbolos del ejemplo. La tabla es una matriz de adyacencias, que representa los links que conectan los símbolos de la primera columna con los símbolos de la primera fila, es decir la relación es en el sentido indicado y no al revés. En esta matriz de adyacencias, los arcos poseen un costo el cual se muestra en la intersección de dos símbolos. Se pueden ver valores entre 1 y 4 en este ejemplo, lo cual representa el número de referencias que el primer símbolo tiene contra el segundo. Es muy importante recordar que sólo consideramos las referencias desde los impactos, ya que nuestra propuesta utiliza solo ellos.

Cada sentencia de los impactos debe describirse respondiendo a la estructura *sujeto + verbo + [objeto]*. En la descripción de los impactos de un sujeto, el *objeto* puede ser opcional, puesto que si el verbo no es transitivo no es necesario ningún *objeto* para completar la descripción. Sin embargo, en la descripción de los *objetos*, este definitivamente va a estar en la oración, puesto que es él en el cual recae la acción. Cabe mencionar que podría suceder que una misma expresión se utilice tanto para describir un sujeto como para describir un objeto. Esto es correcto y no causa ningún problema la redundancia de información, es más, es necesaria para poder caracterizar a cada uno de los símbolos con los cuales se vincula. En el caso de los verbos, la expresión es similar a la de los *sujetos*, en donde el *objeto* también es opcional y esto también es así para los

estados. Sin embargo, al final de la sentencia de los estados se incluye el nuevo estado al que se alcanza luego de realizar la acción, por lo cual, la expresión tiene la forma *sujeto + verbo + [objeto] + estado*.

Tabla 3-3. Matriz de adyacencias de las referencias de los símbolos de la aplicación bancaria.

	Banco	Cliente	Cuenta	Log	Abrir una cuenta	Activada	Registrar	Corroborar	Operar	Depositar	Extraer	Consultar el saldo	Solicitada	Bloqueada	Activada
Banco		1	1	1	1		1	1	1						
Cliente			3							1	1	1			
Cuenta		3								1	1	1			
Log	2						1		2			1			
Abrir una cuenta	1				1										
Activar	1		1												
Registrar	1			1			1		1						
Corroborar	1	1													
Operar		1	1						1						
Depositar	3		2	1			1	1	2						
Extraer	4		2	1			1	1	2						
Consulta el saldo	3		1	1			1	1	2						
Solicitada	1	1	2		1									1	
Bloqueada	1	1	2			1			1						1
Activada		1	1						1						

De acuerdo al principio de circularidad, es recomendado que los sujetos, verbos, objetos y estados que se utilizan en la descripción de los símbolos también estén definidos en el LEL salvo que posean un significado trivial. Con el fin de reforzar el principio de circularidad que persigue lograr una mejor comprensión del lenguaje, es necesario utilizar los términos concretos explícitamente en las descripciones, en lugar de usar pronombres. Si bien podría parecer redundante el repetir constantemente el sujeto que realiza las acciones, esta repetición mejora la lectura y evita cualquier ambigüedad que podría presentarse.

Finalmente, durante la descripción de los símbolos, además de referir a otros símbolos es posible organizarlos desde un término genérico a uno más específico. Es decir, podría suceder que haya una familia de símbolos relacionados y se defina un símbolo el cual describa la generalidad de todos ellos y luego, se definan cada uno de los símbolos específicos los cuales solamente describan la particularidad de cada uno de ellos. Esto no es más que una estructura jerárquica de generalización / especialización y para plasmarlo en los símbolos del LEL, sólo es necesario utilizar la expresión “es un” en la noción, seguido del término que identifica a la parte general. De la misma forma, en el término general, es posible enumerar cada uno de los símbolos específicos.

3.2 Identificación de características transversales con LEL

Considerando al LEL como un modelo adecuado de capturar el conocimiento relacionado con una aplicación, hacemos uso del mismo para identificar características transversales en el contexto de la aplicación. La esencia de nuestro enfoque está basada 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 puesto que debe trabajar con un gran número de símbolos, sin embargo, en primer lugar se va a presentar el mecanismo básico subyacente y en próximas secciones se explica en detalle la estrategia.

Considere los siguientes símbolos de categoría verbo y sus impactos. Si bien los ejemplos muestran la definición completa de los símbolos (con noción e impactos) nuestra propuesta sólo hace uso de los impactos.

Verbo: Depositar

Noción

Acción de agregar dinero a una cuenta.

Impactos

El banco debe corroborar que la persona quien opera es el propietario de la cuenta.

El banco agrega el dinero a la cuenta.

El banco registra la operación en un log.

Figura 3-5. Descripción del símbolo *depositar*

Verbs: Extraer

Noción

Acción de tomar dinero desde la cuenta.

Impactos

El banco debe corroborar que la persona quien opera es el propietario de la cuenta.

El banco debe verificar que la cuenta posee suficiente dinero para realizar la extracción.

El banco debe verificar que el propietario de la cuenta no ha realizado más extracciones que la cantidad permitida.

El banco registra la operación en un log.

Figura 3-6. Descripción del símbolo *extraer*

Verbo: Consultar el saldo

Noción

Acción de informar el monto de dinero que la cuenta posee.

Impactos

El banco debe corroborar que la persona quien opera es el propietario de la cuenta.

El banco informa el monto.

El banco registra la operación en un log.

Figura 3-7. Descripción del símbolo *consultar el saldo*

Verbo: Registrar

Noción

Acción realizada por el banco para agregar la operación al log.

Impactos

El banco registra la operación en un log.

Figura 3-8. Descripción del símbolo *registrar*

Verbo: Corroborar

Noción

Acción realizada por el banco para asegurar la identidad de la persona que quiere realizar una operación con una cuenta.

Impactos

El banco verifica la identidad de la persona.

Figura 3-9. Descripción del símbolo *corroborar*

Los verbos *depositar*, *extraer* y *consultar el saldo* hacen referencia a través de los impactos a los verbos *corroborar* y *registrar*. Estas referencias fueron indicadas en la Tabla 3-3, la cual se vuelve a replicar y se resalta en gris las referencias en cuestión (Tabla 3-4).

Tabla 3-4. Referencias a los símbolos *registrar* y *corroborar* en la aplicación bancaria

	Banco	Cliente	Cuenta	Log	Abrir una cuenta	Activada	Registrar	Corroborar	Operar	Depositar	Extraer	Consultar el saldo	Solicitada	Bloqueada	Activada
Banco		1	1	1	1		1	1	1						
Cliente			3							1	1	1			
Cuenta		3								1	1	1			
Log	2						1		2			1			
Abrir una cuenta	1				1										
Activar	1		1												
Registrar	1			1			1		1						
Corroborar	1	1													
Operar		1	1						1						
Depositar	3		2	1			1	1	2						
Extraer	4		2	1			1	1	2						
Consulta el saldo	3		1	1			1	1	2						
Solicitada	1	1	2		1									1	
Bloqueada	1	1	2			1			1						1
Activada		1	1						1						

Estas referencias significan que además de la funcionalidad básica que se realiza en las operaciones *depositar*, *extraer* y *consultar el saldo*, deben completar su funcionalidad con las operaciones de *corroborar* y *registrar*. El “uso” de funcionalidad adicional, queda de manifiesto en forma explícita en la Figura 3-10, la cual muestra un grafo dirigido con links desde los símbolos *depositar*, *extraer* y *consultar el saldo*, hacia los símbolos *corroborar* y *registrar*.

La esencia de nuestro enfoque radica en identificar como características transversales a los símbolos más referenciados por los impactos de otros símbolos. En este ejemplo, dado que los símbolos *corroborar* y *registrar* son referenciados por los otros tres símbolos, tanto *corroborar* como *registrar* son considerados características transversales candidatas. Es importante remarcar que lo expuesto es la base y esencia de la estrategia, sin embargo, la estrategia utiliza los símbolos de categoría estados para agrupar los otros símbolos y el análisis de referencias se realiza a partir de los grupos determinados por los estados.

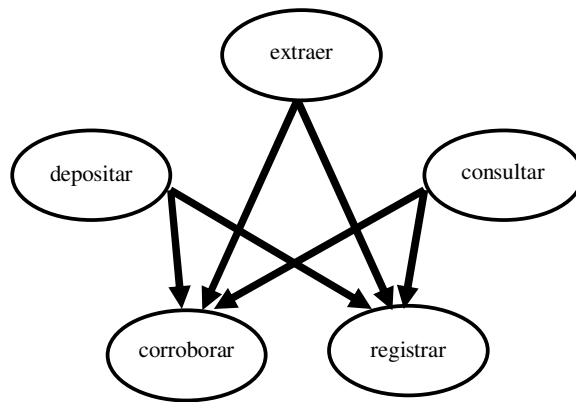


Figura 3-10. Referencias a los símbolos *corroborar* y *registrar*

En el ejemplo previo, los símbolos *depositar*, *extraer* y *consultar el saldo* poseen referencias a los símbolos *corroborar* y *registrar*. Como se pudo apreciar, el sentido de las referencias es crucial para determinar que símbolos son considerados características transversales. El mismo ejemplo se podría haber realizado con referencias en el sentido opuesto, es decir, los símbolos *corroborar* y *registrar* podrían tener referencias a *depositar*, *extraer* y *consultar el saldo* y aunque las referencias estuvieran invertidas, el LEL podría modelar la misma situación (Figura 3-11).

Verbo: Corroborar

Noción

Acción realizada por el banco para asegurar la identidad de la persona que quiere realizar una operación con una cuenta.

Impactos

El banco debe corroborar que la persona quien deposita es el propietario de la cuenta.

El banco debe corroborar que la persona quien extrae es el propietario de la cuenta.

El banco debe corroborar que la persona quien opera es el propietario de la cuenta.

Verb: Registrar

Noción

Acción realizada por el banco para agregar la operación al log.

Impactos

El banco registra el depósito en un log.

El banco registra la extracción en un log.

El banco registra la consulta de saldo en un log.

Figura 3-11. Definición alternativa para los símbolos *corroborar* y *registrar*

Dado que el sentido de las referencias es inverso, es decir, son los símbolos *corroborar* y *registrar* los que referencias a las tres operaciones, la estrategia propuesta no identificaría como características transversales a las operaciones *corroborar* y *registrar*.

Esta forma de estructurar los símbolos *corroborar* y *registrar* se corresponde con una manera factorizada de estructurar la información y es similar a como se definen y organizan los aspectos en el código fuente. Claramente es muy sencillo el identificar características transversales si el LEL se escribiera de esta forma, puesto que se simplifica la estrategia al buscar solamente los símbolos que más referencias salientes poseen. Sin embargo, el esfuerzo de escribir el LEL de esta última forma recae en el

ingeniero de requerimientos ya que no es una forma natural de hacerlo, puesto que el conocimiento no es capturado de forma así organizada.

Por otra parte, es necesario aclarar el porque se consideran solamente las referencias de los impactos y no se consideran también las referencias de la noción. Sucede que los impactos tienen cierta reminiscencia o conexión con la invocación de procedimientos en el código fuente. Los impactos de los verbos en particular se corresponden con un desglose de los pasos o etapas necesarias para lograr el cometido que el verbo determina, lo cual es una analogía directa con la invocación a submódulos. Para el caso de los sujetos y los objetos podemos considerar que los mismos son objetos de un ambiente orientado a objetos, por lo cual, los impactos al hacer referencia a las acciones que realizan o que les realizan también es comportamiento, es decir, métodos que implementan o implementa otro objeto pero que refiere a acciones relacionadas. Finalmente, los estados describen en los impactos las transiciones, que también son las acciones que permiten pasar de un estado a otro. Por lo cual, para todas las categorías, hay una vinculación con la ejecución de comportamiento en los impactos, mientras que en la noción no es sucede.

Otro punto importante para aclarar es el motivo por el cual el análisis no se realiza a nivel de símbolo, sino que en realidad se realiza agrupando símbolos. En el LEL no hay bloques de construcción como puede ser en los requerimientos, diseño o código. En el LEL la única unidad es el símbolo por lo cual definimos crear grupos de símbolos (clusters) a partir de los símbolos estados. Definimos utilizar los estados para agrupar símbolos, porque los símbolos estados modelan en cierta forma los nodos de una máquina de estado y Mahoney [Mahoney 2005] [Mahoney 2007] clama que el comportamiento *core* y transversal es a menudo modelado usando escenarios o máquinas de estados. La razón radica en que las aplicaciones son sistemas complejos reactivos con dos niveles: las máquinas de estados describen el comportamiento intra-objetos, mientras que los escenarios describen el comportamiento inter-objetos.

Un ejemplo de características transversales modeladas como una máquina de estados es el control de acceso a una cuenta bancaria. El mecanismo es el siguiente. El sistema pide un *login* con el fin de brindar el acceso para *depositar, extraer y consultar el saldo*. El proceso de *login* tiene varios pasos, puesto que se ingresan los caracteres de a uno hasta lograr la cantidad necesaria, y luego, el banco debe verificar si el *login* ingresado es correcto. Luego de ello despliega el menú con las tres opciones para que el usuario opere la cuenta. Todos los descriptos son estados posibles en los cuales se puede encontrar la interacción del cliente con el cajero y en todos esos estados es posible cancelar la interacción con el mismo. Como la cancelación puede ser realizada en muchos estados diferentes, la cancelación es considerada una características transversales.

Una descripción detallada de la operatoria es la siguiente. El cajero automático (ATM, por sus siglas en inglés *automated teller machine*) espera a que el cliente ingrese cuatro caracteres. Luego de ingresados los cuatro caracteres, el cajero verifica si el *password* ingresado es correcto. Si el *password* no lo es, el cajero espera un nuevo *password*. Pero si es correcto muestra al cliente el menú con las tres operaciones básicas y el cajero espera a que el cliente elija alguna de las operaciones (Figura 3-12).

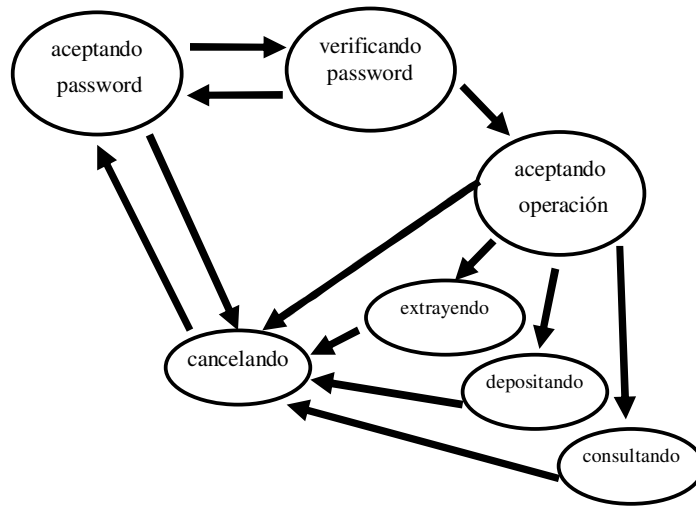


Figura 3-12. Máquina de estados de la interacción con un cajero automático

Como refleja la máquina de estados, tanto el estado *aceptando password*, como *aceptando operación*, puede llevar al estado *cancelando*. Es decir, en esos estados además de verificar si se pulsa una tecla de ingreso de *password* o de opción de menú, se verifica si se pulsa la tecla de cancelar. Esa verificación de que tecla se pulsa, puede verse en el siguiente código de la Figura 3-13, el cual muestra una posible implementación de los estados *aceptando password* y *aceptando operación* con el fin de mostrar el código repetido que debería estar factorizado y aislado en un aspecto.

```

public class Authorizer {

    private Application application;
    private Password password;

    public void acceptPassword (Character character) {
        if (character.isEsc())
            application.cancel
        else {
            password.add (character);
            if (password.size() = 4)
                this.passwordCompleted();
        }
    }

    public void acceptOperation (Character character) {
        if (character.isEsc())
            application.cancel
        else {
            application.perform
            (this.actionsFor (character));
        }
    }
}

```

Figura 3-13. Definición de los métodos *acceptPassword* y *acceptOperation*

El ejemplo muestra que dos diferentes estados poseen código repetido el cual se debería factorizar en aspectos. Consideramos que si organizamos los símbolos del LEL a partir de los estados es posible realizar un análisis similar e identificar las características transversales. Por lo tanto, el enfoque propuesto consiste en construir grupos de símbolos en función de los símbolos estados y luego se deben contar las referencias desde los impactos de símbolos de otros estados. De esta forma se identifica los estados que son candidatas a características transversales. El detalle de los pasos es el siguiente:

- (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.

La Figura 3-14 ilustra el proceso. Las siguientes secciones describen cada uno de los pasos.

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. Dado que las aplicaciones de software pueden ser reducidas (entiéndase por reducidas el ser descriptas y no necesariamente simplificadas) a máquinas de estado [Mahoney 2005] [Mahoney 2007], y por otro lado, dado que el LEL provee una categoría específica de estados y estos símbolos son naturalmente identificados y descriptos, la estrategia determina el agrupar los símbolos en estados. Por lo tanto, el proceso de identificación y descripción de símbolos en el marco de la estrategia para identificar características transversales debe atravesar las siguientes etapas:

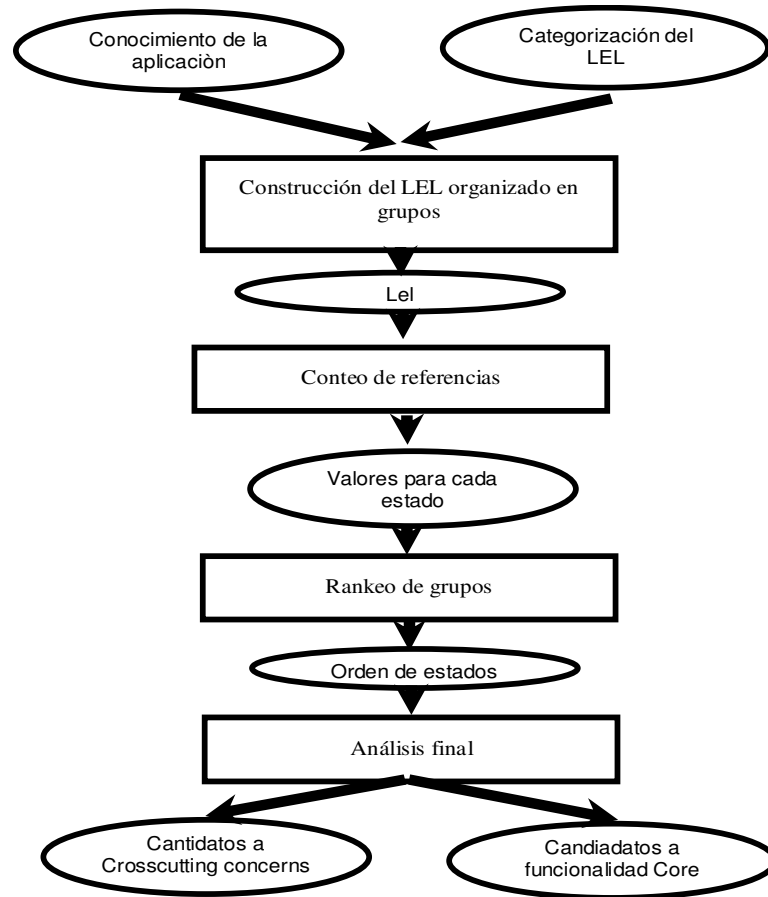


Figura 3-14. Resumen de la estrategia

- (i) 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. Con el fin de identificar símbolos, si se analiza material escrito deben considerarse las palabras más frecuentes. Si en cambio es analizada comunicación oral se deben considerar las palabras que se pronuncian con mayor énfasis. La categorización y descripción de símbolos se debe realizar de acuerdo a la categorización básica la cual determina que información debe poseer la descripción de la noción y de los impactos. Aunque nuestra estrategia solo trabaje con los impactos, es muy importante describir las nociones cuando se describen los símbolos. Las nociones son significativas porque estas descripciones ayudan a identificar sinónimos. Dos símbolos con descripciones de noción similares son candidatos a ser sinónimos. Las nociones también son importantes porque pueden incluir palabras que deben ser consideradas y descriptas en el LEL, por lo cual, también constituyen una fuente de símbolos.
- (ii) El Segundo paso consiste en identificar al símbolo más representativo y significativo de la aplicación. Debe ser un sujeto o un objeto el cual es el

elemento más relevante en el contexto de la aplicación y que además posee una máquina de estados asociada. Los símbolos que fueron identificados para la aplicación bancaria son: *banco*, *cliente*, *cuenta*, *log*, *abrir una cuenta*, *activar una cuenta*, *registrar*, *corroborar*, *operar*, *extraer*, *consultar saldo*, *solicitada*, *bloqueada* y *activada*. Sin embargo, de todos los símbolos, sólo se debe prestar atención a *banco*, *cliente*, *cuenta* y *log*, puesto que los dos primeros son sujetos y los dos últimos son objetos. Y de estos cuatro, el símbolo principal que tiene una máquina de estados asociada es *cuenta*. 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. Los estados vinculados con *cuenta* para la aplicación bancaria son: *solicitada*, *bloqueada* y *activada*. Inicialmente el cliente se dirige al banco y solicita una cuenta, por lo cual, el primer estado es *solicitada*. Luego, el banco abrirá la misma, sin embargo, hasta que el banco no concluya ciertas verificaciones y chequeos la cuenta permanece *bloqueada*. Finalmente, cuando el banco habilita la cuenta para que el cliente la utilice, la misma pasa al estado *activada*. La siguiente figura muestra la transición de estados.

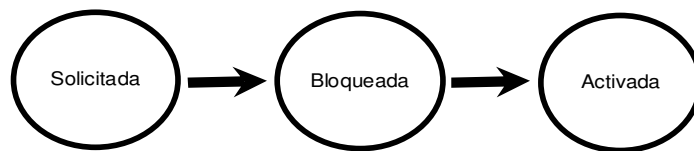


Figura 3-15. Máquina de estados correspondiente a la apertura de una cuenta bancaria

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. Siguiendo con el ejemplo de la aplicación bancaria, se muestra una tabla que relaciona a cada uno de los símbolos (agrupados por categoría de símbolo) con los estados correspondientes. En la tabla puede observarse que todos los símbolos se ubican en el estado donde aparecen por primera vez. En el estado *solicitada* se muestran los símbolos *banco*, *cliente*, *cuenta* y *abrir una cuenta*, puesto que todos esos elementos intervienen para solicitar una cuenta. Luego, de solicitar una cuenta se obtiene una cuenta *bloqueada*, en donde no se pueden realizar operaciones mas que *activar*, sin embargo, se puede intentar operar con la misma y es por ello que tienen sentido las operaciones *registrar* y *corroborar*, como así también el *log*. Luego, en el estado *activado*, se ubican las 3 operaciones que tienen sentido realizar con la cuenta: *depositar*, *extraer* y *consultar el saldo*. También se ubica el símbolos *operar* para generalizar los otros 3.

Es importante mencionar que la estrategia propone un análisis a nivel de grupos ya que un análisis a nivel de símbolo podría ser problemático ya que los símbolos no necesariamente se corresponderían con módulos en una futura implementación, mientras que los estados en cambio describen entidades conceptuales las cuales de

alguna forma estarán plasmadas en la aplicación final. Para este ejemplo, *Registrar* y *Corroborar* serían aspectos, sin embargo, es cuestión de granularidad. No podemos asegurar que cada símbolo identificado como tal sea un aspecto, porque el nivel de granularidad es muy bajo y no sabemos como terminará siendo implementado.

(iii)

Tabla 3-5. Relación entre los símbolos y los estados para la aplicación bancaria

Estado	Sujeto	Objeto	Verbo
solicitada	banco cliente	cuenta	abrir una cuenta
bloqueada		log	activar registrar corroborar
activada			operar depositar extraer consultar el saldo

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 relacionados con 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 nuestra intención es medir el acoplamiento entre distintos grupos, es por ello que sólo nos concentramos en las referencias que trascienden desde un grupo a otro. Por otro lado, es importante notar que no medimos 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 mas 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 esta 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. La Figura 3-16 presenta un algoritmo el cual resume la estrategia de conteo.

```

For each group G in LEL
  totalGroup := 0
  For each category C in (subject, objet, verb)
    totalCategory := 0
    For each symbol S in G where S belongs to C
      Count in referencesToSymbol references from
symbol S2 to symbol S where S2 do not belong to G
      totalCategory := totalCategory +
referencesToSymbol
    totalgroup := totalgroup + totalCategory
  Count in differentGroups different group G2 where G2
has a symbol S2 where S2 has references to S

```

Figura 3-16. Estrategia de conteo

La siguiente tabla muestra las referencias entre los símbolos agrupadas en estados. Sólo se indican las referencias entre distintos estados dado que es lo que interesa a la estrategia propuesta.

Tabla 3-6. Conteo de referencias entre los distintos grupos de la aplicación bancaria

		Solicitada					Bloqueada					Activada				
		Solicitada	Banco	Cliente	Cuenta	Abrir una cuenta	Bloqueada	Log	Activar	Registrar	Corroborar	Activar	Operar	Depositar	Extraer	Consultar el saldo
Solicitada	Solicitada					1										
	Banco						1		1	1		1				
	Cliente												1	1	1	
	Cuenta												1	1	1	
	Abrir una cuenta															
Bloqueada	Bloqueada		1	1	2						1	1				
	Log															
	Activar		1		1											
	Registrar		1									1				
	Corroborar		1	1												
Activada	Activada			1	1											
	Operar			1	1											
	Depositar		3		2			1		1	1					
	Extraer		4		2			1		1	1					
	Consultar saldo		3		1			1		1	1					

Tabla 3-7 resume las referencias que cada uno de los grupos posee. Muestra tanto el promedio general de referencias, como los promedios por cada una de las categorías de símbolos, como así también información de la cantidad de grupos desde los cuales salen las referencias.

Consideremos el primer grupo, *solicitado*, para ejemplificar los distintos valores que se muestran en la tabla. La primera columna, *grupos que lo referencian*, posee un valor 2, puesto que los símbolos del grupo *solicitada*, son referenciados por símbolos que pertenecen a dos grupos distintos: *bloqueada* y *activada* (los otros dos grupos del ejemplo). Luego, la columna *promedio general de referencias* da una medida de la cantidad de referencias que el grupo posee dividido la cantidad de símbolos del grupo. Y las tres columnas siguientes, también indican promedios, pero acotados a una categoría específica de símbolo: sujeto, objeto o verbo. La última columna, *promedio de referencias a verbos*, posee un valor 0, puesto que el grupo *solicitada*, posee un sólo verbo, *abrir una cuenta*, y este símbolo no posee referencia alguna, es por ello que el promedio es 0. La columna *promedio de referencias a objetos* posee un valor 10, porque el grupo *solicitada* tiene un sólo objeto, *cuenta*, el cual posee 3 referencias desde el grupo *bloqueada* y 7 referencias desde el grupo *activada*, esto da un total de 10 referencias, y dado que es un sólo símbolo, el promedio es 10. Por su parte, la columna *promedio de referencias a sujetos* posee un valor 9, porque *solicitada* tiene 2 sujetos, *banco* y *cliente*; *banco* posee 4 referencias desde *bloqueada* y 10 referencias desde *activada*; mientras que *cliente* posee 2 referencias desde *bloqueada* y 2 referencias desde *activada*. El total de referencias para *banco* es 14 y el total de referencias para *cliente* es 4, y la suma de ambos es 18, por lo cual, el promedio de referencias para los dos símbolos es de 9. Finalmente, el promedio general de referencias para *solicitada* es de 5,6, puesto que la cantidad total de referencias del grupo es de 28: 10 referencias a objetos y 18 referencias a sujetos; y el grupo posee 5 símbolos: *solicitada*, *banco*, *cliente*, *cuenta* y *abrir cuenta*, por lo cual, el cociente 28 sobre 5 da como resultado 5,6.

Tabla 3-7. Conteo de referencias detallado

	Grupos que los referencian	Promedio general de referencias	Promedio de referencias a sujetos	Promedio de referencias a objetos	Promedio de referencias a verbos	
1	Solicitada	2	5.6	9.0	10.0	0
2	Bloqueada	2	2.4	0	4.0	2.7
3	Activada	2	2.0	0	0	2.3

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 nuestro enfoque, los elementos (ya sean origen o destino) son los grupos de símbolos. Y usamos 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.

Pueden existir grupos que maximizan (i) el número de grupos que los referencian pero poseen bajo (ii) promedio de referencias. Si se presenta esta situación, lo que puede ocurrir es que un subconjunto de los símbolos del grupo son los que reciben todas las referencias, entonces al recibirlas sólo un subconjunto de los símbolos y no todos los símbolos, el promedio de referencias es bajo, a pesar de que se encuentra disperso. En este caso, es necesario analizar la posibilidad de que el subconjunto de símbolos y no todo el grupo, sea considerado característica transversal.

También puede ocurrir que haya grupos que poseen un bajo valor en grupos que lo referencian (i) mientras que poseen un alto promedio general de referencias (ii). Esto muestra que el grupo en cuestión si bien está altamente acoplado con unos pocos grupos, pero no lo disperso necesario como para que sea considerado característica transversal.

Finalmente, puede ocurrir que un grupo minimice la cantidad de grupos de que lo referencian (i) y también minimice el promedio general de referencias (ii). En este caso, no se debe considerar en absoluto el grupo, puesto que no posee acoplamiento ni esparcimiento, por lo cual, sencillamente debe ser ignorado.

La Figura 3-17 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 mas oscura indica la ubicación de las características transversales candidatas.

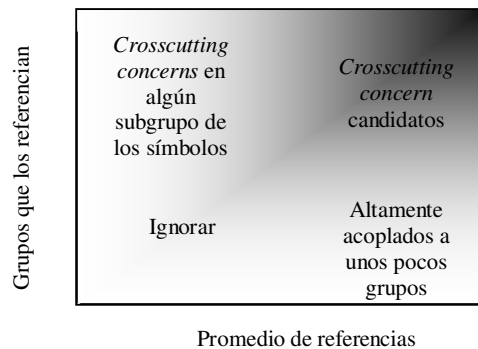


Figura 3-17. Nivel de dispersión (*scattering*) y acoplamiento (*coupling*)

Figura 3-18 muestra el ranking de los grupos del ejemplo de la aplicación bancaria. Dado que la aplicación bancaria posee solo tres grupos y que cada uno de los grupos posee referencias desde los otros dos grupos, todos los grupos se encuentran en el mismo valor de y, pero el promedio general de referencias es el que varía, es por ello que el grupo 1 *solicitada*, se encuentra en el extremo derecho, el grupo 2 *bloqueada* se encuentra en el centro de los otros dos grupos y finalmente el grupo 3 *activada* es que se encuentra más a la izquierda.

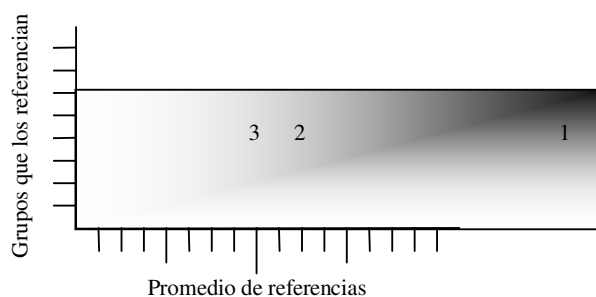


Figura 3-18. Ranking de los grupos del ejemplo bancario

En esta situación es fácil concluir un orden lineal ya que todos los grupos tienen el mismo número de *grupos que los referencian*, por lo cual, sólo resta ordenarlos de acuerdo al *promedio de referencias*. Por lo tanto, el primer candidato a ser considerado característica transversal es el grupo con el valor más alto en *promedio general de referencias* que es el grupo 1 *solicitada*. Luego, se encuentra el segundo valor más alto (es el valor del medio), el grupo 2 *bloqueada*. Y finalmente se ubica el grupo 3 *activada*. Tabla 3-8 muestra el orden final.

Tabla 3-8. Orden final

		Grupos que lo referencian	Promedio general de referencias
1	Solicitada	2	5.6
2	Bloqueada	2	2.4
3	Activada	2	2.0

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. Tabla 3-9 presenta el análisis detallado de los grupos de la aplicación bancaria.

Resumiendo, el grupo *bloqueada* es el único que debe ser considerado característica transversal. Los tres grupos poseen el mismo número de grupos que los referencian. Aunque el grupo *solicitada* tiene un alto valor en promedio general de referencias, las logra por medio de símbolos *core*, por lo cual no debe ser considerado característica transversal. El grupo *activada* que se encuentra en tercer lugar en función del *promedio*

general de referencias, también posee símbolos *core* y no debe ser considerado. Por lo cual, el grupo *bloqueada* es el único a considerarse característica transversal.

Tabla 3-9. Análisis detallado

Id	Estado	Descripción
1	Solicitada	Los símbolos <i>banco</i> y <i>cuenta</i> son los principales sujetos y objetos de la aplicación y alteran las referencias, es por ello que estos símbolos no deben considerarse. Por lo cual, este grupo no debe ser considerado candidato.
2	Bloqueada	Este grupo posee sólo objetos y verbos, y las referencias están balanceadas entre todos ellos. Por lo cual, este grupo debe ser considerado candidato.
3	Activada	Este grupo posee sólo verbos y las referencias están bastante balanceadas entre todos ellos. Sin embargo, estos verbos son considerados símbolos principales en el lenguaje de la aplicación. Por lo cual, este grupo no debe ser considerado candidato.

Haciendo un análisis intuitivo, como lo haría un experto en el dominio, el también concluiría que el grupo *bloqueada* debe ser considerada característica transversal, dado que el grupo posee términos relacionados con atributos de calidad como ser *logging* y *autorización*, los cuales son reconocidos como característica transversal.

3.3 Casos de estudio

Esta sección muestra la aplicación de la estrategia 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.

Ambas aplicaciones ya se han implementado y se encuentran en período de mantenimiento. Ninguna de las dos se ha diseñado o desarrollado con tecnología de aspectos por diferentes razones. El sistema antievasión de impuestos fue desarrollado hace varios años y el equipo de desarrollo no tenía experiencia con la tecnología de aspectos. Más aún, las herramientas utilizadas no proveían soporte para esta tecnología. Con respecto al portal web de noticias, cuando comenzó el desarrollo, los requerimientos eran muy pobres y la funcionalidad aspectos no fue detectada. Luego, la aplicación fue extendida de una forma desordenada y si bien los aspectos fueron identificados, los mismos nunca fueron factorizados como tales.

El autor de esta tesis ha trabajado en ambas aplicaciones por varios años, por lo cual, posee un amplio conocimiento de las mismas. Él ha trabajado por más de 10 años con la primera aplicación y ha trabajado por más de 3 años con la segunda. Aunque él ha construido algún LEL parcial durante el desarrollo de cada una de las aplicaciones, los LELs construidos necesitaban ser completados y revisados. Utilizando herramientas básicas y de propósito general tales como procesadores de texto y planillas de cálculo, se construyó el LEL y se realizaron los cálculos para ambas aplicaciones.

Si bien el objetivo de esta sección es simplemente 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 en otra sección, de todas formas, en esta sección

se van a mostrar cada uno de los casos de estudio y al final de cada caso de estudio se van a comparar los resultados obtenidos por la estrategia con la opinión de diferentes expertos que estuvieron involucrados con ambos sistemas. Uno de los expertos que da su opinión es el autor de este trabajo, el cual se desarrolló como diseñador y desarrollador de la primera aplicación y se desarrolló como analista funcional de la segunda. Sin embargo, a pesar de la opinión calificada del autor, se buscó la opinión de otros participantes de la construcción del sistema, con el objetivo de presentar opiniones objetivas de otros expertos involucrados en la construcción. Por lo cual, se desarrollaron entrevistas estructuradas en las cuales se realizaron las siguientes preguntas: (i) ¿Qué rol desempeño durante el desarrollo? (ii) ¿Cuántos años de experiencia posee? (iii) ¿Qué estudios académicos posee? (iv) ¿Qué funcionalidad considera identificarse como características transversales? (v) Justifique la respuesta anterior.

Las siguientes secciones describen cada uno de los casos de estudio. En primero lugar se describe el análisis realizado a la aplicación de antievasión de impuestos, mientras que en segundo lugar se describe el análisis realizado al portal web. Cabe señalar que la primera aplicación es rica en situaciones y acciones que son llevadas a cabo durante la fase de prosecución. La segunda aplicación en cambio se respalda en la tecnología ya que muchas de las acciones que se realizan solo tienen sentido dentro del mundo de la tecnología. Por lo cual, a pesar de que la estrategia se ideó con el fin de identificar características transversales en las fases de análisis, con el segundo caso de estudio se muestra que la estrategia se puede utilizar de la misma forma al código fuente e identificar características transversales a partir de ellos.

3.3.1 Caso de estudio antievasión de impuestos

Cuando un ciudadano no cumple con sus deberes impositivos el organismo demandante del gobierno realiza las acciones necesarias para obligar al deudor a cumplir con sus obligaciones. El curso normal de acciones implica: (i) establecimiento de la deuda, (ii) procedimientos a nivel administrativo, (iii) procedimientos a nivel judicial, (iv) procedimientos a nivel cautelar y (v) cancelación de deuda.

Los cinco pasos indicados previamente son actividades más complejas las cuales incluyen las siguientes tareas:

- (i) Establecimiento de la deuda: situación en la cual el contribuyente ha incurrido en una deuda y él ha rechazado o desaprovechado todas las oportunidades posteriores que se les brindaron para ponerse al día.
- (ii) Procedimientos a nivel administrativo: en esta etapa, se realiza un análisis costo / beneficio para determinar si el esfuerzo de reclamar la deuda es provechoso o no. Si la relación costo / beneficio resulta positiva, también es necesario determinar si es una deuda crítica o no (ya sea por el monto, por las características del deudor, etc.) con el fin de tomar las medidas preventivas.
- (iii) Procedimientos a nivel judicial: consiste básicamente en 4 etapas: hacer la presentación en un juzgado, exhortarlo a pagar, tomar las medidas preventivas y finalmente lograr la sentencia. Dado que las medidas preventivas si pudieron haber llevado a cabo en el paso anterior, la secuencia de acciones no es única.

- (iv) Procedimientos a nivel cautelar: básicamente consiste en esperar a que el deudor voluntariamente y espontáneamente se presente a pagar su deuda. Se espera a que el deudor lo realice, puesto que previo a este paso, se tomaron todas las medidas legales para cerrarle las puertas a cualquier acto legal o administrativo, por lo cual, si el mismo se ve en esta necesidad, no le quedará otra opción mas que saldar su deuda. Sin embargo, también hay alternativas, por ejemplo, si el deudor poseyera alguna propiedad (un bien mueble o inmueble) el organismo demandante podría hacer uso de los mismos para cubrir el monto adeudado.
- (v) Cancelación de la deuda: es la etapa final en la cual, la deuda es recuperada y las medidas preventivas son anuladas.

Si bien el flujo normal y esperado se corresponde con los cinco pasos enunciados y descriptos previamente, pueden presentarse situaciones las cuales pueden alterar el flujo normal de eventos. Las situaciones son las siguientes:

- (vi) Deuda reclamada por error: ocurre cuando se reclama una deuda, pero en realidad es un error porque la misma no existe, es decir, el deudor realmente no falló en pagar sus obligaciones. Es un simple error administrativo. Sin embargo, aunque la deuda no exista, dado que el deudor tuvo oportunidades previas para aclarar la situación de todas formas deberá abonar los gastos administrativos que se ocasionaron.
- (vii) Plan de facilidades de pago: es un mecanismo que permite al deudor pagar su deuda en pagos mensuales. Es un compromiso que asume y puede desencadenar en el pago completo de la deuda, o también podría suceder que el deudor abandone el plan de facilidades por lo cual, incurriría nuevamente en una deuda.
- (viii) Subasta: situación en la cual el deudor posee bienes (muebles o inmuebles) por lo cual, se subastan los mismos con el fin de recuperar la deuda.
- (ix) Medidas preventivas administrativas: situación en la cual no es beneficioso el llevar un acto judicial con el fin de reclamar la deuda y obtener su pago. Por lo cual, solo se realizan medidas preventivas a nivel administrativo con el fin de restringir las acciones del deudor para forzarlo a pagar su deuda.

Si bien la descripción provista del contexto de la aplicación es breve, puede observarse de la misma que las nueve situaciones planteadas se corresponden con estados en los que pueden encontrarse el reclamo de la deuda. Y este elemento constituye el primer paso en la aplicación de la estrategia, el determinar el sujeto u objeto principal, junto con los estados asociados al mismo. Por lo cual, el objeto principal es la deuda y los estados por los que pasa son los nueve estados enunciados. Luego de ello, se relacionan a cada uno de los restantes símbolos con alguno de los estados y se calculan las referencias entre los símbolos que trascienden los límites del estado al que pertenece. Hecho este análisis, se muestra en la siguiente tabla, los valores correspondientes a las mediciones necesarias para identificar a los características transversales candidatas.

Tabla 3-10. Conteo de referencias para la aplicación de antievasión de impuestos

		Números de grupos que lo referencian	Promedio general de referencias	Promedio de referencias a sujetos	Promedio de referencias a objetos	Promedio de referencias a verbos
1	Establecimiento de deuda	7	3.7	10	2.3	1.2
2	Procedimiento a nivel administrativo	7	4.3	6	4.7	0
3	Procedimiento a nivel judicial	8	2.0	2	3.3	0.5
4	Procedimiento a nivel cautelar	1	0.7	0	0	0
5	Cancelación de deuda	4	2.0	2	0	1.3
6	Deuda reclamada por error	1	3.0	0	0	2.0
7	Plan de facilidades de pago	4	2.3	0	0	1.7
8	Subasta	2	1.0	0	0	2.0
9	Medidas preventivas administrativas	4	2.0	0	0	2.2

La siguiente figura muestra la dispersión de los grupos de acuerdo al número de grupos que los referencian y al promedio general de las referencias.

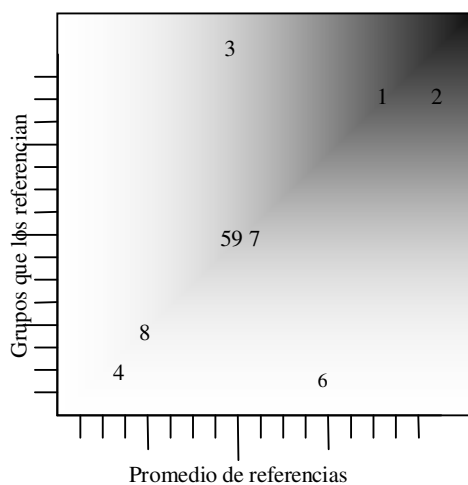


Figura 3-19. Ranking de los grupos del sistema antievasión

En general, los grupos se encuentran ubicados en la diagonal principal de la Figura 3-19, por lo cual, es clara la condición de característica transversal de cada uno de los grupos. Los grupos 2 y 1 se encuentran respectivamente en primer y segundo lugar, al encontrarse en la zona más oscura de la figura (y lo demuestran los valores de promedio de referencias y grupos que lo referencian). Luego debe ser considerado el grupo 3,

porque tiene aproximadamente los mismos valores que 5, 9 y 7, sin embargo, posee un número mucho mayor (el mas grande de entre todos los grupos) respecto a los grupos que lo referencian. Luego se deben ubicar los grupos 7, 5 y 9. En particular, 5 y 9 poseen exactamente los mismos valores de promedio de referencias y grupos que los referencian, sin embargo, el grupo 5 es ubicado antes del grupo 9 puesto que el grupo 9 sólo está relacionado con verbos, mientras que el grupo 5 es relacionado con sujetos y verbos, y esta variedad hace que el grupo se ubique en primer lugar. Luego se ubica el grupo 8, porque posee dos grupos que lo referencian, mientras que los dos grupos restantes, poseen sólo un grupo que los referencian. Por lo tanto, finalmente se ubican los grupos 6 y 4, ambos poseen un sólo grupo que los referencian. La siguiente tabla muestra el orden final de los grupos.

Tabla 3-11. Orden final de los grupos del sistema antievasión

		Número de grupos que los referencian	Promedio general de referencias
2	Procedimiento a nivel administrativo	7	4.3
1	Establecimiento de la deuda	7	3.7
3	Procedimiento a nivel judicial	8	2.0
7	Plan de facilidades de pago	4	2.3
5	Cancelación de la deuda	4	2.0
9	Medidas preventivas administrativas	4	2.0
8	Subasta	2	1.0
6	Deuda reclamada por error	1	3.0
4	Procedimiento a nivel cautelar	1	0.7

Aunque la tabla Tabla 3-11 muestra el orden de probabilidad de que los distintos grupos sean considerados características transversales, es necesario realizar un análisis detallado de la situación de cada uno de los grupos, ya que un grupo podría contener un símbolo principal el cual podría alterar el ranking. La tabla Tabla 3-12 describe el análisis detallado realizado con el fin de identificar las características transversales candidatas.

Resumiendo, los grupos que deben ser considerados características transversales (ordenados desde el más probable al menos probable) son: *procedimiento a nivel administrativo*, *plan de facilidades de pago*, *cancelación de la deuda* y *medidas preventivas*.

Tabla 3-12. Análisis detallado del sistema antievasión de impuestos

Id	Estado	Descripción
2	Procedimiento a nivel administrativo	El símbolo <i>demandante</i> es un sujeto principal en el lenguaje de la aplicación y las referencias del mismo elevan los promedios. Sin embargo, el <i>demandante</i> no es el único que produce un alto promedio de referencias. El <i>demandante</i> es un sujeto y el promedio de referencias de los sujetos es medio. Sin embargo, el grupo también posee valores altos por un objeto <i>bienes</i> el cual logra que el grupo tenga el valor más alto de promedio de referencias a objetos. Por lo tanto, el grupo debe ser considerado como candidato.
1	Establecimiento de la deuda	Este grupo posee el símbolo <i>deudor</i> el cual es un sujeto principal en el lenguaje de la aplicación y este símbolo es el que provoca que el grupo posea valores altos. El símbolo <i>deudor</i> posee 32 referencias mientras que el promedio general es 3,7 y el promedio elevado se debe exclusivamente a este símbolo. Por lo cual, este grupo no debe ser considerado como candidato.
3	Procedimiento a nivel judicial	El símbolo <i>paso judicial</i> es un símbolo principal del sistema. Este símbolo posee 10 referencias mientras que el promedio general del grupo es 2. Por lo tanto, el grupo posee valores altos por este símbolo, de modo que este grupo no debe ser considerado como candidato.
7	Plan de facilidades de pago	El número de grupos que los referencian y el promedio general de referencias son valores medios de entre los valores de todos los grupos, por lo cual, este grupo es algo disperso y algo acoplado. Además, posee altos valores de referencia a verbos. De modo que si bien en términos generales posee valores medios, los valores específicos de las referencias a verbos determinan que debe ser considerado como candidato.
5	Cancelación de la deuda	El número de grupos que los referencian y el promedio general de referencias son valores medios de entre los valores de todos los grupos, por lo cual, este grupo es algo disperso y algo acoplado. Además, posee valores medios de referencia a verbos. De modo que si bien en términos generales posee valores medios, los valores específicos de las referencias a verbos determinan que debe ser considerado como candidato.
9	Medidas preventivas	El número de grupos que los referencian y el promedio general de referencias son valores medios de entre los valores de todos los grupos, por lo cual, este grupo es algo disperso y algo acoplado. Además, posee valores medios de referencia a verbos. De modo que si bien en términos generales posee valores medios, los valores específicos de las referencias a verbos determinan que debe ser considerado como candidato.
8	Subasta	El número de grupos que los referencian y el promedio general de referencias son valores bajos, por lo cual este grupo no debe ser considerado como candidato.
6	Deuda reclamada por error	Aunque este grupo posee un alto valor en promedio general de referencias, el número de grupos que lo referencian es bajo. Por lo cual, este grupo es acoplado con muy pocos grupos, en realidad con un sólo grupo. Por lo tanto, este grupo no debe ser considerado como candidato.
4	Procedimiento a nivel cautelar	El número de grupos que los referencian y el promedio general de referencias son valores bajos, por lo cual este grupo no debe ser considerado como candidato.

El autor de este trabajo, quien estuvo involucrado en el desarrollo del sistema de este caso de estudio, esta de acuerdo con los cuatro candidatos que han sido identificados por la estrategia. En particular, el autor identifica en primer lugar a los grupos *plan de*

facilidades de pago y cancelación de la deuda como los candidatos más firmes. La justificación es que no importa en que estado se encuentre el juicio, este podría continuar con el ingreso a un *plan de facilidades de pago* si la deuda es real y legítima, o bien podría continuar con la *cancelación de la deuda* si la deuda no es real. Por lo tanto, la aplicación necesita funcionalidad para permitir que un juicio pueda cambiar desde cualquier estado a alguno de estos dos. Luego, con un análisis mas detallado, se concluye que *procedimiento a nivel administrativo y medidas preventivas* también son candidatos a ser características transversales. *Procedimiento a nivel administrativo* logra este status porque el grupo posee un objeto el cual es referenciado frecuentemente por otros grupos. Y esto significa que la aplicación de software contendrá un recurso (archivo, estructura de datos u objeto) el cual será utilizado desde muchos lugares diferentes. *Medidas preventivas* presenta una situación similar porque posee verbos los cuales son frecuentemente referenciados, y traducido a una aplicación de software, implica que habrá varias invocaciones de métodos o funciones desde diferentes módulos.

Por lo tanto, de la misma forma que ocurre en la aplicación bancaria, en la cual cada operación debe ser enriquecida con funcionalidad para *registrar* las operaciones y para *corroborar* la identidad de los usuarios, en la aplicación de antievasión, en muchos estados en los cuales un juicio se puede encontrar, deben ser enriquecidos con funcionalidad para mover a un juicio a los estados de *plan de facilidades de pago y cancelación de la deuda*. Hay que tener presente que mover no sólo significa cambiar el estado, sino que también significa ajustar algunas cosas relacionadas con el cambio de estado. Luego, desde que muchos grupos referencia a objetos o verbos de *procedimiento a nivel administrativo y medidas preventivas*, muchos estados deben ser enriquecidos con está funcionalidad también.

Además de la opinión del autor de este trabajo, se entrevistaron a dos personas que estuvieron involucradas con el desarrollo de la aplicación y están familiarizadas con la tecnología de aspectos. Una de ellas posee 28 años de experiencia y desempeñó los roles de analista funcional, diseñador y arquitecto en el desarrollo de la aplicación. La otra persona posee 25 años de experiencia, era el líder del proyecto y también desarrolló el rol de analista funcional en el desarrollo de la aplicación. Ambas personas poseen título universitario de grado.

El primer entrevistado consideró a la siguiente funcionalidad para ser características transversales: *plan de facilidades de pago, cancelación de la deuda y medidas preventivas*. Él basó su opinión en el hecho de que los juicios siguen una secuencia de pasos lineal en donde un paso es seguido de otro, sin embargo, hay algunos pasos que son transversales a todos los demás. *Plan de facilidades de pago* es uno de ellos. Al comienzo del desarrollo de la aplicación antievasión había un solo punto o instante de tiempo en el cual un juicio podía ingresar al *plan de facilidades de pago*, sin embargo, la gestión de los juicios fue sufriendo cambios y por sobre le final del desarrollo el ingreso al *plan de facilidades de pago* era una opción a la cual se podía acceder desde cualquier estado en que se encuentre el juicio. *Medidas preventivas* tiene ciertas similitudes con *plan de facilidades de pago*. Luego, *cancelación de la deuda* era el estado final de entre los eventos posibles en la realización de un juicio y podía ser alcanzado solamente desde el estado previo, sin embargo, al igual que *plan de facilidades de pago y medidas preventivas* fueron siendo posible acceder desde cualquier otro estado del juicio, sucedió lo mismo con *liquidación de la deuda*, y más

aún, *cancelación de la deuda* puede ser alcanzado incluso desde *plan de facilidades de pago y medidas preventivas*.

El segundo entrevistado consideró solamente a *plan de facilidades de pago* como característica transversal. Él clama que *plan de facilidades de pago* es un privilegio el cual puede ser accedido desde cualquier situación en la cual el juicio se puede encontrar. Por lo tanto, él considera que *plan de facilidades de pago* es una característica transversal ya que constituye funcionalidad extra que debe ser agregada a cualquier situación en la que se encuentre el juicio para permitir el paso a *plan de facilidades de pago*.

Aunque sólo se entrevistaron a dos personas para corroborar los resultados de este caso de estudio, estas dos personas tienen una gran experiencia y su opinión es muy valiosa y significativa. Por lo cual, es determinante el hecho de que ellos hayan identificado como características transversales elementos que también lo identificó la estrategia propuesta. En realidad, la estrategia propuesta identificó a cuatro candidatos a ser características transversales; uno de los entrevistados identificó también a 3 de ellos, y el otro identificó sólo a 1 de ellos. Es importante destacar que la característica transversal que fue identificada por los dos entrevistados, la estrategia propuesta lo ubicó en segundo lugar como candidato, por lo cual, en cierta forma hay una coherencia entre la posición en que lo ubicó la estrategia propuesta y la coincidencia de los dos entrevistados.

A ambos entrevistados se les preguntó el porque no eligieron a *procedimiento a nivel administrativo* como candidato a característica transversal. Ambos indicaron que en el momento en que fueron consultados no habían considerado el que fuera característica transversal por ello no lo eligieron, sin embargo, ante esta pregunta realizada luego de la entrevista, meditaron la situación de grupo *procedimiento a nivel administrativo* y admiten que probablemente pueda ser considerada característica transversal, sin embargo, deben analizar con más detalle para terminar de definirlo.

La tabla Tabla 3-13 resume los grupos identificados como características transversales por la estrategia propuesta y por los dos miembros del equipo de desarrollo. La tabla muestra que la estrategia propuesta concuerda en tres de las 4 características transversales identificadas con el primer miembro y sólo en 1 con el segundo. Y el hecho más sobresaliente es que nuestra estrategia tiene por objetivo identificar características transversales candidatas y como los miembros identificaron características transversales que también lo identificó nuestra estrategia y no fue nada más allá de ello, de alguna forma muestra la efectividad de la estrategia.

3.3.2 Caso de estudio del portal web

La estrategia de identificación de características transversales también fue aplicada a un sistema web que se encarga de publicar noticias diariamente. Las mismas son desarrolladas por redactores quienes luego van a ser supervisados por editores, los que son encargados de revisar las notas y decidir si se publican o no. La publicación de las noticias no es trivial, puesto que se prepara un formato HTML para un navegar web, pero también se preparan otros formatos para teléfonos celulares y dispositivos móviles. Además de construir los distintos formatos para cada noticia es necesario realizar algún tipo de categorización sobre las mismas. Por lo tanto, la publicación de las notas desde que el redactor las crea, hasta que el usuario final puede leerlas desde la web pasa por distintos estados. Los mismos son: *redactada, aceptada, HTML construido, versiones auxiliares construidas, categorizada, taggeada, incluida en el calendario y publicada*.

Tabla 3-13. Características transversales identificadas por la estrategia y por el equipo de desarrollo

Id	Estado	Identificado por la estrategia	Identificado por el miembro #1	Identificado por el miembro #2
2	Procedimiento a nivel administrativo	Si		
1	Establecimiento de la deuda			
3	Procedimiento a nivel judicial			
7	Plan de facilidades de pago	Si	Si	Si
5	Cancelación de la deuda	Si	Si	
9	Medidas preventivas	Si	Si	
8	Subasta			
6	Deuda reclamada por error			
4	Procedimiento a nivel cautelar			

El primer estado es *redactada* y representa la situación en donde el redactor preparó su nota y la dejó lista para que el editor haga su revisión. Luego de la revisión, el editor puede descartar la nota, y si eso sucede la nota pasa al estado *rechazada*, el cual es un estado final, es decir, una vez que la nota fue calificada como rechazada la misma es descartada y nunca se la considerará para su publicación. Sin embargo, si la noticia pasa la revisión del editor y es considerada para su publicación, pasa al estado *aceptada* y desde allí es necesario crear las versiones para publicarla. El próximo estado es *HTML construido* y luego de construir el HTML se construyen las versiones auxiliares, por lo cual, pasa al estado *versiones auxiliares construidas*. Los próximos pasos para la publicación de la noticia son opcionales, es por ello que se pueden obviar, realizar alguno de ellos, o realizar todos. Los siguientes estados son: *categorizada*, *taggeada* e *incluida en el calendario*. Luego de todos estos pasos se alcanza el estado *publicado*.

Para el dominio descripto, se construyó el LEL correspondiente, agrupando los símbolos en función de los estados y a partir de ellos se contaron las referencias necesarias para la estrategia propuesta. La tabla Tabla 3-13 muestra los nueve grupos con los valores de referencia para cada uno de los estados.

La figura Figura 3-20 muestra la dispersión de los grupos en función del número de grupos que los referencian y en función del promedio general de referencias. Hay un grupo el cual posee un valor muy alto de promedio general de referencias y es por él que el diagrama se extiende tanto en el eje x. Luego, el resto de los grupos están distribuidos por la misma área, sólo que la dispersión no es buena, ya que 6 de los 8 grupos restantes poseen sólo 1 grupo que los referencian, por lo cual, se encuentran todos sobre una misma línea.

Tabla 3-14. Conteo de referencias para el caso de estudio del portal web de noticias

		Números de grupos que lo referencian	Promedio general de referencias	Promedio de referencias a sujetos	Promedio de referencias a objetos	Promedio de referencias a verbos
1	Redactado	5	2.5	2	6.3	0
2	Rechazado	1	0.5	0	0	0
3	Aceptado	5	13.4	32	0	1.0
4	HTML construido	3	0.9	0	0.3	1.4
5	Versiones auxiliares construidas	1	0.2	0	0	0.2
6	Categorizado	1	0.6	0	0	0.2
7	Incluido en calendario	1	1.3	0	0	1.0
8	Taggeado	1	1.0	0	0	0.5
9	Publicado	1	2.5	0	0	1.0

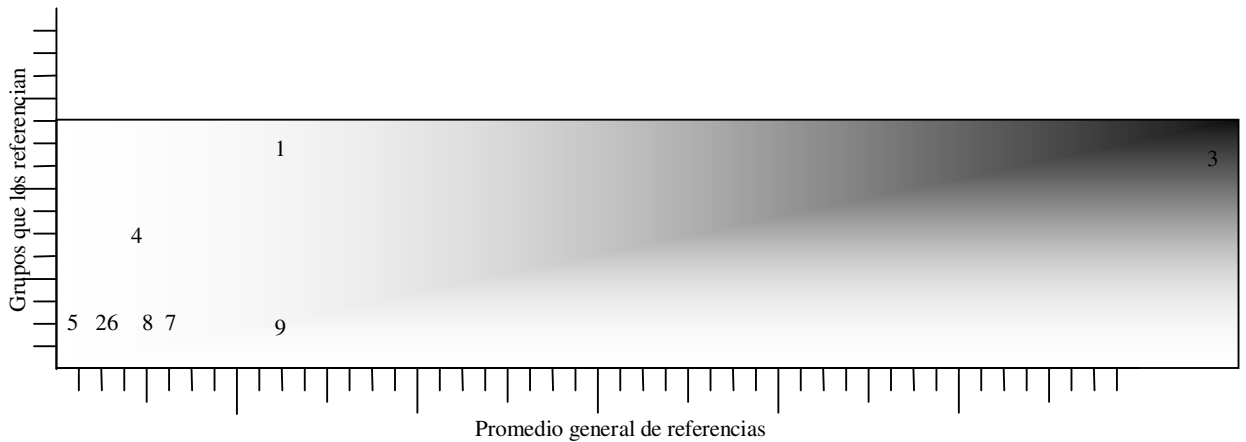


Figura 3-20. Ranking de grupos del portal web

El grupo 3 definitivamente ocupa el primer lugar de los candidatos desde que posee el valor más alto en promedio de referencias y en grupos que los referencian. Luego, tanto los grupos 1 como el 9 poseen el mismo valor en promedio de referencias, pero en función de grupos que los referencian, el grupo 1 tiene un valor más alto, por lo cual, el segundo grupo es el número 1. Luego, el grupo 9 posee sólo un grupo que lo referencian y así sucede con 5 grupos más. Sólo el grupo 4 es el que posee un valor distinto de 1 en grupos que lo referencian, de modo que él ocupa la tercera posición. Los restantes grupos, todos con valor 1 en grupos que los referencian, se ocupan a continuación, ordenados por promedio general de las referencias. La tabla Tabla 3-14 resume esta información.

Aunque la tabla Tabla 3-15 muestra el orden de probabilidad de que los distintos grupos sean consideradas características transversales, es necesario realizar un análisis detallado de la situación de cada uno de los grupos, ya que un grupo podría contener un

símbolo principal el cual podría alterar el ranking. La siguiente tabla describe el análisis detallado realizado con el fin de identificar las características transversales candidatas.

Tabla 3-15. Orden final de los grupos del portal web

		Grupos que los referencian	Promedio general de referencias
3	Aceptado	5	13.4
1	Redactado	5	2.5
4	HTML construido	3	0.9
9	Publicado	1	2.5
7	Incluido en calendario	1	1.3
8	Taggeado	1	1.0
6	Categorizado	1	0.6
2	Rechazado	1	0.5
5	Versiones auxiliares construidas	1	0.2

En resumen, de acuerdo con el análisis descripto en la tabla Tabla 3-16, el único grupo que debe ser considerado como característica transversal candidata es *HTML construido*.

Utilizando nuestra estrategia es fácil identificar el grupo *HTML construido* como característica transversal candidata ya que es uno de los tres grupos que poseen más de un grupo que lo referencian. Sin embargo, el autor de esta tesis estuvo involucrado en el desarrollo del sistema en cuestión, y durante el desarrollo del mismo fue difícil para él identificarlo intuitivamente como característica transversal, básicamente porque durante el desarrollo de la aplicación no se contaron con todos los requerimientos desde el inicio y el desarrollo fue un tanto desordenado. Sin embargo, analizando en retrospectiva y teniendo una visión completa del sistema es claro que el grupo merece ser identificado como característica transversal ya que define verbos (que en definitiva serán funciones) que implementan funcionalidad que está dispersa por gran parte del sistema.

Además de esta opinión subjetiva, se realizó una encuesta a 7 miembros del equipo de desarrollo con el fin de validar esta opinión. Los miembros del equipo se pueden organizar en 3 grupos en función de su experiencia. El grupo más experimentado lo conforma solo un miembro: el líder de proyecto, arquitecto, diseñador y analista del equipo. Un graduado en ciencias de la computación con 12 años de experiencia. Luego, el segundo grupo, es un grupo de experiencia media, que lo integran 4 miembros que poseen de entre 3 a 5 años de experiencia, 1 de ellos graduado en ciencias de la computación y los otros 3 estudiantes avanzados. Estos miembros se desarrollaron como diseñadores y desarrolladores. Finalmente, el último grupo, el menos experimentado, lo constituyen 2 miembros del equipo, desarrolladores con menos de 2 años de experiencia. Si bien las respuestas de todos los miembros se consideran por igual, el agrupamiento se hace con el fin de caracterizar a las personas entrevistadas.

Tabla 3-16. Análisis detallado de los grupos del portal web

Id	Estado	Descripción
3	Aceptado	Este grupo posee el símbolo <i>editor</i> el cual es un sujeto principal en el lenguaje de la aplicación, por lo que posee muchas referencias y altera los cálculos. <i>Editor</i> posee 63 referencias mientras que el promedio general del grupo es 13,4. Si <i>editor</i> no se consideraría en el grupo, el promedio general no sería tan alto. Por lo cual, este grupo no debe ser considerado como candidato.
1	Redactado	Este grupo posee el símbolo <i>contenido</i> el cual es un objeto principal en el lenguaje de la aplicación. El símbolo posee 12 referencias mientras que el promedio general es de 2,5. Si <i>contenido</i> no se consideraría en el grupo, el promedio general no sería tan alto. Por lo cual, este grupo no debe ser considerado como candidato.
4	HTML construido	Este grupo posee varios verbos los cuales son referenciados en una medida similar por otros tres grupos. Por lo tanto, este grupo debe ser considerado como candidato.
9	Publicado	Aunque este grupo posee un alto promedio general de referencias, este grupo posee solo 1 grupo que lo referencia. Por lo tanto, este grupo no debe ser considerado como candidato.
7	Incluido en calendario	Aunque este grupo posee un alto promedio general de referencias, este grupo posee solo 1 grupo que lo referencia. Por lo tanto, este grupo no debe ser considerado como candidato.
8	Taggeado	Aunque este grupo posee un alto promedio general de referencias, este grupo posee solo 1 grupo que lo referencia. Por lo tanto, este grupo no debe ser considerado como candidato.
6	Categorizado	Este grupo posee un bajo promedio general de referencias, además posee solo 1 grupo que lo referencia. Por lo tanto, este grupo no debe ser considerado como candidato.
2	Rechazado	Este grupo posee un bajo promedio general de referencias, además posee solo 1 grupo que lo referencia. Por lo tanto, este grupo no debe ser considerado como candidato.
5	Versiones auxiliares construidas	Este grupo posee un bajo promedio general de referencias, además posee solo 1 grupo que lo referencia. Por lo tanto, este grupo no debe ser considerado como candidato.

El único grupo que fue identificado como característica transversal fue *HTML construido*. El mismo fue identificado por la mayoría de los miembros (4 de 7). El más experimentado de los miembros del equipo estuvo de acuerdo con esta decisión. El indicó: “Aunque pensé que los aspectos eran: *versions auxiliares construidas*, *categorizados*, *incluidos en calendario* y *taggeado*, después de analizar los símbolos del grupo *HTML construido*, estuve completamente seguro de que este grupo debe ser considerado característica transversal porque este grupo contiene la funcionalidad que fue incorporada gradualmente y en forma desordenada en la aplicación. Y sólo cuando hubimos avanzado en el desarrollo de la aplicación nos dimos cuenta de que hubiera sido conveniente diseñar esta funcionalidad como característica transversal”. Luego, del grupo de miembros con experiencia media, 2 de las 4 personas concordaron en que *HTML construido* debe ser característica transversal. Una de las argumentaciones a favor indican que: “Los métodos que se encuentran agrupados en *HTML construido* son claramente métodos compartidos por toda la aplicación, y por sobre todo, son métodos que sufrieron muchos cambios durante el desarrollo”. Hay otra argumentación a favor

del grupo *HTML construido*, sin embargo la argumentación no es completamente a favor sino que presenta algunas dudas: “Aunque es complejo entender el diseño de la aplicación a través de grupos de estados, es posible realizarlo analizando los métodos que están ubicados en cada uno de los grupos. Algunos de los métodos definidos en *HTML construido* pueden estar localizados en otros grupos, y no estoy convencido de si es necesario factorizarlos como aspectos”. Finalmente, hay una argumentación en contra de la identificación de *HTML construido* como característica transversal, sin embargo, reconoce que algún tipo de factorización es necesaria, aunque la duda es si debe realizarse utilizando la tecnología de aspectos. Por lo cual, es importante mencionar que la estrategia identifica características transversales candidatas que los expertos necesitan analizar en detalle para determinar si son características transversales o no. Lo importante no es si *HTML construido* es característica transversal o no, lo importante es que aunque los miembros del equipo no están completamente de acuerdo en si debe ser o no, si están de acuerdo en que si algún estado debe serlo, ese debe ser *HTML construido*. La siguiente tabla resume las respuestas de cada uno de los miembros del equipo.

Tabla 3-17. Características transversales candidatas identificados por la propuesta y por el equipo de desarrollo.

Id	Estado	Identificado por la estrategia	Identificado por el miembro #1	Identificado por el miembro #2	Identificado por el miembro #3	Identificado por el miembro #4	Identificado por el miembro #5	Identificado por el miembro #6	Identificado por el miembro #7
3	Aceptado								
1	Redactado								
4	HTML construido	Si	Si	Si	Si			Si	
9	Publicado								
7	Incluido en calendario								
8	Taggeado								
6	Categorizado								
2	Rechazado								
5	Versiones auxiliares construidas								

4 Experimentación

Este capítulo presenta un experimento que prueba la efectividad de la estrategia de identificación de características transversales, para la cual se muestra su aplicabilidad en un par de casos de estudios en capítulo previo.

La efectividad la probamos comparando la estrategia del LEL contra la estrategia Theme/Doc [Baniassad 2004]. Con este fin, definimos una misma aplicación para la cual aplicamos ambas estrategias. Luego, contrastamos los resultados en términos de las características transversales identificadas por cada una de las estrategias. Nuestra hipótesis es que si las características transversales identificadas por la estrategia del LEL incluyen a las características transversales identificadas por Theme/Doc podemos decir que la estrategia del LEL es más efectiva.

Hemos seleccionado Theme/Doc para hacer la comparación ya que es una estrategia que presenta gran trabajo y experiencia detrás de ella [Clarke 2005]. Más aún, ambas estrategias Theme/Doc y LEL poseen varias similitudes en su esencia. Ambas estrategias se basan en identificar términos los cuales interactúan con muchos otros términos. Theme/Doc analiza verbos mientras que la estrategia del LEL también considera sustantivos. La diferencia de ambas estrategias radica en los productos con los cuales trabaja cada uno y en la granularidad del análisis. Theme/Doc trabaja sobre requerimientos, mientras que la estrategia del LEL trabaja sobre los símbolos que lo componen, que se organizan a partir de símbolos de una de las categorías posibles: estados. Esta diferencia en los productos con los cuales trabaja, condiciona la granularidad de análisis de cada una de las estrategias. Theme/Doc identifica términos (acciones) que se encuentran dispersos entre los requerimientos, mientras que el LEL identifica los símbolos estados (los cuales agrupa a símbolos de las otras categorías). Sin embargo, la estrategia del LEL permite hacer un análisis más profundo, analizando dentro del estado. Por lo cual, si bien en un primer nivel se identifican estados que tienen un nivel de granularidad diferente al de los términos de Theme/Doc, en un nivel más detallado se puede analizar dentro del estado para identificar símbolos que podrían ser comparables con los términos de Theme/Doc.

Para demostrar la efectividad de la estrategia del LEL se diseñó un experimento de tipo aleatorio simple entre sujetos (*simple between-subjects randomized*) con características realísticas. El experimento consistió en presentar la descripción de una aplicación a los participantes del experimento para pedirles que produzcan los materiales con los cuales aplicar la estrategia Theme/Doc y del LEL. Estos materiales son: requerimientos y símbolos del LEL respectivamente. Para ello conformamos dos grupos de participantes, en donde cada uno trabajó con un producto distinto. Luego de que los participantes del experimento produjeron los materiales necesarios para aplicar las estrategias, ambas estrategias se pueden aplicar sistemáticamente y sin ningún tipo de subjetividad. Por lo cual, el experimento consistió en obtener de un grupo variado de participantes un conjunto variado de materiales fuentes para aplicar las estrategias, las cuales, dado que

sólo pueden aplicarse de una sola forma y sin subjetividad, fueron realizadas por el autor de esta tesis.

A continuación describimos el objetivo del experimento a través del método Objetivo/Duda/Métrica (*Goal/Question/Metric, GQM*) propuesto por Basili et al. [Basili 1994].

*Analizar las técnicas de LEL y Theme/Doc
para el propósito de evaluación
con respecto a la efectividad
desde el punto de vista de potenciales usuarios
en el contexto de profesionales del área*

De un grupo numeroso de candidatos, fueron seleccionados 20 participantes. Las personas seleccionadas poseen un promedio de experiencia de 4 años, con un rango que va desde los 6 meses a los 18 años de experiencia en la disciplina. Todas las personas poseen educación universitaria (algunos son estudiantes, otros graduados y algunos estudiantes de posgrado). Los participantes recibieron una descripción de un sistema administrativo, una especie de gestor de incidencias. El mismo les resultaba familiar, ya que ellos utilizan uno similar. Sin embargo, la especificación no era exactamente la aplicación que comúnmente usan. Los participantes recibieron una breve presentación sobre como escribir los requerimientos y el LEL. La tarea de producir el LEL podría considerarse un tanto más exigente que la de producir los requerimientos. En primer lugar, porque los participantes tienen algún tipo de conocimiento sobre como especificar requerimientos, mientras que ninguno de los participantes tuvo contacto alguno con el LEL. Además, construir un LEL implica identificar los símbolos, describirlos y organizarlos en grupo. Sin embargo, a pesar de la diferencia en la curva de aprendizaje para producir los requerimientos y el LEL, ambos productos poseen la misma esencia: el definir los links adecuados entre sus elementos. Links que en definitiva son los que determinaran las características transversales, y no importa si los requerimientos pueden ser descritos más rápido que los símbolos del LEL, lo importante es que el participante del experimento haya reparado en los mismos y los haya dejado explícitos en el material que tuvo que producir.

El resto del capítulo está organizado de acuerdo a Shull [Shull 2008] de la siguiente manera. En primer lugar se presenta la planificación del experimento con el objetivo del mismo, los participantes, los materiales, las tareas, la hipótesis y las variables, el diseño, el procedimiento de ejecución y el procedimiento de análisis. Luego se describe como se llevó a cabo la ejecución para luego hacer un análisis y finalmente presentar las conclusiones.

4.1 Planificación

Esta sección describe los siguientes elementos respecto de la planificación del experimento: el objetivo, los participantes, los materiales, las tareas, la hipótesis y las variables, el diseño, el procedimiento de ejecución y el procedimiento de análisis.

4.1.1 Objetivo

El objetivo de este experimento es probar que la estrategia de identificación de características transversales es más efectiva que la estrategia Theme/Doc. Por lo cual, utilizamos Theme/Doc como referencia y verificamos que las características

transversales que identifica Theme/Doc también son identificadas por nuestra estrategia. Y no sólo verificamos que cierta característica transversal sea identificada por ambas estrategias, sino que también verificamos la frecuencia de identificación en las distintas estrategias.

Es importante mencionar que la comparación de identificación de características transversales entre ambas estrategias no puede realizarse de forma directa, ya que Theme/Doc identifica acciones mientras que la estrategia propuesta identifica estados en un primer nivel. Sin embargo, luego de identificar los estados es posible hacer un análisis más detallado para encontrar los verbos que se corresponden con los identificados por Theme/Doc.

El objetivo del experimento se especifica utilizando la plantilla (*template*) GQM:

Analizar las técnicas Theme/Doc y LEL
por el propósito de contrastar la identificación de características transversales
entre las estrategias de LEL y Theme/Doc
con respecto a la efectividad de la estrategia LEL.

4.1.2 Participantes

Con el fin de conducir el experimento, seleccionamos a 20 personas que constituyen una muestra al azar de un grupo de profesionales más grande que participa en proyectos de desarrollo de software en una organización de la Argentina. Las personas seleccionadas poseen un promedio de 4 años de experiencia, con un rango que va desde los 6 meses a los 18 años. Este grupo, si bien lo comprenden personas que actualmente se desarrollan en diferentes roles, todos poseen experiencia en requerimientos de alguna y otra forma, es decir, tienen experiencia relevando, especificando, validando o al menos consumiendo los requerimientos. Todos los participantes poseen educación universitaria. Alguno de ellos son estudiantes, otros graduados en ciencias de la computación y algunos otros posgraduados.

Todos los participantes tienen algún tipo de conocimiento y experiencia en el dominio de la aplicación que se utiliza en el experimento, dado que los participantes utilizan una aplicación de características similares en el trabajo diario, sin embargo, los participantes no han visto antes del experimento la especificación utilizada en el experimento.

Los participantes están todos familiarizados con la tecnología de aspectos en el código fuente, sin embargo, ellos no poseen experiencia en la identificación de características transversales en las etapas tempranas del desarrollo de software. Por lo cual, no poseen conocimiento alguno de las técnicas Theme/Doc ni de LEL.

El experimento consiste en identificar características transversales con Theme/Doc y con LEL. Para hacerlo con Theme/Doc es suficiente con analizar la especificación provista y escribir requerimientos a partir de ella, mientras que para identificar los aspectos con la estrategia de LEL, es necesario construir el LEL, para poder identificar las características transversales. Si bien del grupo del cual seleccionamos los 20 participantes para experimento había personas que conocían como construir un LEL, preferimos no involucrar a estas personas en el experimento, así que tomamos sólo personas que nunca lo habían construido y le dimos una breve introducción sobre como hacerlo.

Es importante aclarar que los participantes se ofrecieron como voluntarios para participar del experimento. Ellos decidieron formar parte del mismo porque habían escuchado sobre la estrategia de LEL para identificar características transversales y quisieron involucrarse en el experimento para testear si la estrategia es efectiva o no.

4.1.3 Materiales del experimento

Los siguientes materiales fueron necesarios para realizar el experimento. En primer lugar se preparó una presentación con una breve descripción de la aplicación (presentación de aproximadamente 20 minutos de duración). Luego, se preparó una presentación sobre como escribir requerimientos para que puedan ser utilizados por la estrategia Theme/Doc. Dado que la plantilla (*template*) de requerimientos que necesita Theme/Doc es muy simple, se preparó un sólo *slide* para tal fin. Después, se prepararon *slides* sobre como construir el LEL. Finalmente, se preparó una especificación de requerimientos de software (*software requirements specification*, SRS) con el detalle de las características de la aplicación para la cual se deben identificar las características transversales.

Es importante destacar que la SRS poseía un glosario y una lista de requerimientos funcionales, sin embargo, los mismos no estaban descriptos de la forma necesaria para que puedan ser utilizados por las dos estrategias de identificación de características transversales que se comparan en el experimento. El glosario era muy simple y no estaba descripto como la estrategia de identificación de características transversales de LEL lo requiere. Y la lista de requerimientos funcionales era confusa y alguna información estaba mezclada. Por lo cual, la SRS proveía información detallada sobre la aplicación, pero esta información debía ser analizada y no consumida directamente. Cabe señalar que la SRS también incluía un prototipo funcional de la aplicación.

4.1.4 Tarea

Todos los participantes tuvieron que leer en detalle la especificación de requerimientos provista. Luego, se armaron dos grupos, y uno de ellos debió escribir requerimientos para poder aplicar la estrategia Theme/Doc mientras que el otro grupo debió construir el LEL con el fin de aplicar la estrategia de identificación de características transversales del LEL.

El esfuerzo necesario para producir cada uno de los productos es muy distinto, puesto que la escritura de los requerimientos podría considerarse una síntesis de la SRS, mientras que la construcción del LEL podría considerarse un desglose y descripción detallada de la SRS. Sin embargo, a pesar de la diferencia de esfuerzo en generar los productos para cada una de las estrategias, el esfuerzo necesario para analizar la SRS y entenderla y poder generar cada uno de los productos podría considerarse similar. Por lo cual, la diferencia radica solamente en la generación del material final, en donde uno de ellos (los requerimientos) son mas simples que el otro (el LEL). De modo que para que una actividad no condicione a la otra, dividimos los grupos de participantes y cada uno de ellos trabajó con una sola estrategia.

Luego de que la materia prima para aplicar las estrategias Theme/Doc y LEL fue generada por los participantes, el autor de esta tesis aplicó ambas estrategias. Esta tarea no se encomendó a los participantes ya que ambas estrategias pueden aplicarse sistemáticamente y sin ningún tipo de subjetividad ya que ambas estrategias son bien determinísticas y radican básicamente en realizar conteos de ciertos elementos.

Theme/Doc consiste básicamente de dos etapas. Primero se deben organizar los requerimientos de acuerdo a las acciones, es decir, se deben relacionar a las acciones con los requerimientos en los cuales aparece. Luego, la estrategia consiste en identificar a las acciones que son relacionadas con el mayor número de requerimientos. La estrategia de LEL también es sistemática y también se puede realizar sin ningún tipo de subjetividad. Los participantes debían proveer el LEL construido y organizado según lo necesita la estrategia. Luego, la estrategia de identificación de características transversales de LEL consiste en contar las referencias entre los distintos símbolos. Y se debe identificar como características transversales a los grupos de símbolos que maximizan dos variables: promedio de referencias y grupos de símbolos que los referencian.

Por lo tanto, la estrategia Theme/Doc y la del LEL pueden ser aplicadas sistemáticamente sin ningún tipo de subjetividad una vez que los materiales fuentes son producidos. Este es el motivo por el cual decidimos pedirles a los participantes del experimento que sólo produzcan el material fuente.

4.1.5 Hipótesis y variables

El experimento posee un sólo objetivo: contrastar la identificación de características transversales por la estrategia del LEL contra la identificación de características transversales por la estrategia Theme/Doc. En verdad, nuestro interés es probar que la estrategia de LEL identifica al menos las mismas características transversales identificadas por Theme/Doc. Más aún, estamos interesados en probar que las características transversales identificadas por la estrategia LEL son identificados más frecuentemente que en Theme/Doc. Por lo tanto, nuestra hipótesis es:

H011: La estrategia de LEL debe identificar todos las características transversales identificados por la estrategia Theme/Doc con mayor frecuencia en que son identificados por esta última.

Existe dos variables: la técnica (estrategia de LEL o estrategia Theme/Doc) y las características transversales identificadas por las estrategias. La estrategia es una variable independiente porque es lo que varía para testear los resultados, mientras que el conjunto de características transversales es una variable dependiente porque queremos medir como varía de acuerdo a la técnica. La siguiente tabla describe ambas variables.

Tabla 4-1. Variables del experimento

Nombre de la variable	Tipo de variable	Clase	Entidad	Tipo de escala	Rango
Tipo de técnica de identificación de características transversales	Independiente	Método	Estrategia de identificación	Nominal	LEL, Theme/Doc
Características transversales identificadas	Dependiente	Producto	Característica transversal	Nominal	Cada característica transversal en el dominio de la aplicación

4.1.6 Diseño

Dado que el experimento posee una sola variable independiente, decidimos realizar un experimento simple. En particular, elegimos un diseño aleatorio simple entre sujetos (*simple between-subjects randomized*) donde conformamos dos grupos de sujetos y cada grupo fue asignado a solo una técnica (*treatment*). La selección (*selection*) es la principal amenaza (*threat*) a la validez interna (*internal validity*) debido el efecto de la variación natural en la performance de los individuos. De modo que los sujetos fueron asignados a cada técnica en forma al azar para evitar este threat.

Si bien pudimos haber solicitado a los participantes que preparen ambos materiales: requerimientos y LEL, quisimos evitar que la preparación de un material condicionara la preparación del otro. Así que preferimos que cada sujeto se concentrara en un sólo material de forma que la calidad de la misma sea la mejor posible.

4.1.7 Procedimiento

El procedimiento del experimento consiste de las siguientes etapas. En primer lugar se realizó una presentación donde se instruyó a los participantes sobre el dominio de la aplicación, sobre como escribir requerimientos y como construir el LEL. Luego de ello, los participantes tuvieron una semana de tiempo para producir los materiales que le fueron solicitados: requerimientos y LEL. Finalmente, el autor de esta tesis aplicó las estrategias Theme/Doc y LEL a los materiales necesarios para cada una de ellas: requerimientos y LEL.

La presentación del dominio de la aplicación fue realizada en la oficina de los participantes luego del horario laboral. Además de la presentación los participantes recibieron una descripción sobre el dominio, de modo que si bien luego de la presentación los participantes se llevaron una introducción del dominio, también poseían información adicional para repasar o profundizar sobre la aplicación.

Luego de la presentación inicial los participantes fueron divididos en dos grupos. Un grupo fue instruido sobre como escribir requerimientos los cuales serían utilizados a posteriori con Theme/Doc y el otro grupo fue instruido en como construir el LEL para ser utilizados en la estrategia de LEL.

Los participantes contaron con 5 días para terminar el trabajo. Luego, ellos entregaron las especificaciones de requerimientos y el LEL al autor, quien aplicó las estrategias correspondientes.

4.1.8 Procedimiento de análisis

El procedimiento de análisis consiste en contrastar las características transversales identificadas por la estrategia Theme/Doc con las características transversales identificadas por la estrategia de LEL, a la vez de verificar que la estrategia de LEL identifica las características transversales en forma más frecuente que Theme/Doc.

En primer lugar tenemos que establecer el conjunto de características transversales definidas por Theme/Doc, ya que hay 10 sujetos que escribieron distintos conjuntos de requerimientos para la misma aplicación y los diferentes conjuntos de requerimientos originan diferentes características transversales. De modo que consideramos a la unión de todas las características transversales identificadas por cada conjunto de requerimientos como el conjunto final de características transversales. Luego, tenemos que unificar las características transversales identificadas con Theme/Doc y los

identificados por la estrategia del LEL, ya que ambas estrategias poseen diferente nivel de abstracción, desde que Theme/Doc identifica acciones mientras que la estrategia del LEL identifica estados. Esta diferencia radica en un estado de LEL es un conjunto de verbos (además de sujetos y objetos). De modo que cuando Theme/Doc detecta una acción como característica transversal, la estrategia del LEL debe detectar el estado que incluye a dicha acción. Una posibilidad es que Theme/Doc detecte dos acciones, las cuales en el LEL están dentro del mismo estado, por lo cual, mientras que Theme/Doc detecta dos características transversales, en LEL se detecta uno. Así, con el fin de contrastar las características transversales de ambos enfoques, se deben analizar las acciones identificadas por Theme/Doc. Luego, se deben verificar que cada acción pertenezca a un estado el cual haya sido identificado como característica transversal por la estrategia del LEL. Luego, se debe verificar que la acción dentro de LEL tenga referencias suficientes para que se considere a la misma como característica transversal. Finalmente, se deben calcular las frecuencias de comparación de entre Theme/Doc y la estrategia del LEL.

4.2 Ejecución

En general, el experimento se ejecutó sin ajustes significativos respecto del procedimiento diseñado. La única diferencia es que los participantes necesitaron más asistencia para producir el material solicitado. El experimento puede ser dividido en dos fases: (i) la fase en donde los participantes produjeron el material fuente, y (ii) la fase donde el autor de esta tesis usó el material fuente para aplicar las estrategias e identificar características transversales. La primera fase, en donde los participantes produjeron los requerimientos y el LEL, comenzó una presentación del dominio de la aplicación la cual fue seguida por una presentación de las técnicas que debían utilizarse para hacer las especificaciones. Concluida la presentación, se les pidió a los participantes que se tomen 5 días para producir los materiales solicitados. La segunda fase fue ejecutada por el autor de esta tesis, quien aplicó la estrategia Theme/Doc a los requerimientos y aplicó la técnica de LEL al LEL. El resto de la sección describe en detalle cada una de las dos etapas indicadas.

La primera etapa duró un poco más de tiempo y requirió algo más de esfuerzo de lo planeado. En primer lugar se demoró la sesión de presentación dado que algunos participantes no pudieron llegar a tiempo debido a compromisos laborales de último momento, por lo tanto, el comienzo de la presentación fue demorado algunos minutos para esperar a todos los participantes. Mientras se llevó a cabo la espera, algunos participantes se mostraron muy bien dispuestos y estaban interesados en conocer el dominio de la aplicación y las técnicas. Una vez que el resto de los participantes se unieron al grupo, comenzó la sesión de presentación. Los participantes mostraron mucho interés en el dominio de la aplicación y aunque el objetivo fue sólo el dar una introducción básica a la aplicación, los participantes hicieron muchas preguntas, incluso, siguieron consultando luego de que la sesión terminó. Luego, armamos dos grupos de personas, uno que trabajaría con los requerimientos y otro que trabajaría con el LEL. De modo que presentamos la plantilla (*template*) de los requerimientos a un grupo y la plantilla (*template*) de construcción del LEL al otro grupo. El grupo que debió escribir requerimientos debió utilizar la plantilla (*template*) de la figura Figura 4-1.

Esta plantilla (*template*) es la clásica y bien conocida plantilla (*template*) usada para escribir requerimientos. La primera oración es la más simple, mientras que la segunda,

es algo más compleja ya que relaciona dos acciones. Este tipo de relaciones es la que origina las características transversales cuando la estrategia Theme/Doc es aplicada. Es importante mencionar que esto último no fue mencionado a los participantes. A continuación proveemos dos ejemplos de requerimientos, uno por cada tipo de oración, sobre la aplicación de seguimiento de pedidos.

El sistema debe permitir <a alguien> <hacer algo>
El sistema debe permitir <a alguien> <hacer algo> <en relación a algo más>

Figura 4-1. Plantilla de requerimientos

R.1.- El sistema debe permitir al especialista crear un pedido.
R.2.- El sistema debe permitir al jefe de sección asignar un pedido en función de su visibilidad de personas.

Figura 4-2. Requerimientos de ejemplo para cada plantilla

En el ejemplo, R.1 tiene una sola acción: *crear*, por lo cual, esta acción está sólo relacionada con el requerimiento R1. Luego, las acciones *asignar* y *visibilidad* están relacionadas entre ellas a través del requerimiento R.2. Esta relación entre *asignar* y *visibilidad* es la cual origina las características transversales. Si bien esta explicación no fue dada a los participantes, ellos sí recibieron un gran conjunto de requerimientos los que ejemplificaron el trabajo que debían hacer. A continuación se presenta un ejemplo más completo de requerimientos.

R.3.- El sistema debe permitir al jefe de área crear un pedido.
R.4.- El sistema debe permitir al jefe de sección crear un pedido.
R.5.- El sistema debe permitir al especialista crear un pedido.
R.6.- El sistema debe permitir al jefe de área asignar un pedido en función de su visibilidad de personas.
R.7.- El sistema debe permitir al jefe de sección asignar un pedido en función de su visibilidad de personas..
R.8.- El sistema debe permitir al especialista asignar un pedido en función de su visibilidad de personas.
R.9.- El sistema debe permitir al jefe de área ver a todas las personas.
R.10.- El sistema debe permitir al jefe de sección ver a todas las personas en su equipo.
R.11.- El sistema debe permitir al especialista verse solo a sí mismo.
R.12.- El sistema debe permitir al especialista marcar un pedido como finalizado.
R.13.- El sistema debe permitir al jefe de área calcular las estadísticas en función de su visibilidad de personas.
R.14.- El sistema debe permitir al jefe de sección calcular las estadísticas en función de su visibilidad de personas.
R.15.- El sistema debe permitir al especialista calcular las estadísticas en función de su visibilidad de personas.
R.16.- El sistema debe permitir al jefe de área ver todos los pedidos.
R.17.- El sistema debe permitir al jefe de sección ver los pedidos de su equipo.
R.18.- El sistema debe permitir al especialista ver sus propios pedidos.
R.19.- El sistema debe permitir al jefe de área redistribuir a los especialistas.

Figura 4-3. Requerimientos de la aplicación de pedidos

El segundo grupo de participantes tuvo que construir el LEL. A continuación se proveen algunos ejemplos de símbolos agrupados por categoría para la aplicación de pedidos utilizada en el experimento.

Tabla 4-2. Símbolos de ejemplo con su categoría.

Categoría	Símbolos
Sujeto	jefe de área, jefe de sección, especialista
Objeto	Pedido
Verbo	Crear, asignar, ver, marcar como finalizado, calcular, redistribuir
Estado	Nuevo, asignado, finalizado

A continuación se describen los impactos de algunos símbolos (uno por categoría). Figura 4-4 describe un símbolo de categoría objeto: *pedido*; Figura 4-5 describe un símbolo de categoría estado: *nuevo*; Figura 4-6 describe un símbolo de categoría sujeto: *jefe de área*; y Figura 4-7 y Figura 4-8 describen símbolos de categoría verbo: *crear* y *asignar*.

Pedido

Objeto

Noción

Solicitud realizada al área de sistemas para que ellos la resuelvan.

Impactos

El jefe de área puede crear un pedido

El jefe de sección puede crear un pedido

El especialista puede crear un pedido

El jefe de área puede ver pedidos en función de su visibilidad

El jefe de sección puede ver pedidos en función de su visibilidad

El especialista puede ver pedidos en función de su visibilidad

El jefe de área puede asignar pedidos en función de su visibilidad

El jefe de sección puede asignar pedidos en función de su visibilidad

El especialista puede asignar pedidos en función de su visibilidad

El jefe de área puede calcular estadísticas en función de su visibilidad

El jefe de sección puede calcular estadísticas en función de su visibilidad

El jefe de área puede redistribuir especialistas

Figura 4-4. Definición del símbolo *pedido*

Nuevo

Estado

Noción

Situación en la cual el pedido fue creado recientemente.

Impactos

El jefe de área asigna un pedido y el pedido cambia a estado asignado

El jefe de sección asigna un pedido y el pedido cambia a estado asignado

Figura 4-5. Definición del símbolo *nuevo*

Jefe de área

Sujeto

Noción

Persona a cargo de toda el área de sistemas.

Impactos

El jefe de área puede crear un pedido

El jefe de área puede ver pedidos de acuerdo a su visibilidad

El jefe de área puede asignar pedidos de acuerdo a su visibilidad

El jefe de área puede calcular estadísticas de acuerdo a su visibilidad

El jefe de área puede redistribuir especialistas

Figura 4-6. Definición del símbolo *jefe de área*

Crear

Verbo

Noción

Acción por la cual se reporta un nuevo pedido al área de sistemas.

Impactos

El jefe de área puede crear un pedido

El jefe de sección puede crear un pedido

El especialista puede crear un pedido

Figura 4-7. Definición del símbolo *crear*

Asignar

Verbo

Noción

Acción por la cual se determina un responsable para la realización del pedido.

Impactos

El jefe de área puede asignar pedidos de acuerdo a su visibilidad

El jefe de sección puede asignar pedidos de acuerdo a su visibilidad

El especialista puede asignar pedidos de acuerdo a su visibilidad

Figura 4-8. Definición del símbolo *asignar*

Luego de que los símbolos fueron definidos, los participantes tuvieron que agrupar los símbolos en estados. La siguiente es una guía sobre como identificar el grupo al que pertenece un símbolo:

- (i) El símbolo es originado en el estado
- (ii) La acción crea el estado o permite la transición hacia el estado
- (iii) La acción sólo puede ser realizada en el estado
- (iv) Nociones básicas del dominio de la aplicación deben ser relacionadas con el estado inicial

A continuación se muestra un ejemplo de organización de símbolos en estados para la aplicación de pedidos usada en el experimento.

Tabla 4-3. Símbolos de LEL agrupados por estado

Estado	Símbolos
Nuevo	Pedido, jefe de área, jefe de sección, especialista, crear, asignar
Asignado	ver, marcar como finalizado
Finalizado	Calcular estadísticas, redistribuir

Este ejemplo no fue brindado a los participantes del experimento, el mismo sólo se incluye en esta tesis para mostrar la clase de trabajo que los participantes tuvieron que realizar.

El ejemplo muestra al símbolo *asignar* como una acción del grupo *nuevo*, y la misma refiere al símbolo *pedido* que forma parte del mismo estado. Para la estrategia de identificación de características transversales, estas referencias no deben ser consideradas. Sin embargo, el símbolo *asignar* referencia al símbolo *ver* el cual está relacionado con el grupo *asignado*, por lo cual, esta referencia sí debe ser considerada. Siguiendo con este razonamiento, se puede concluir que el término *ver* debe ser

considerado como característica transversal puesto que es muy referenciado por símbolos de otros grupos.

Aunque habíamos planificado terminar la presentación con la descripción de las plantillas (*templates*) y con pedirles a los participantes que entreguen sus especificaciones en un tiempo de 5 días, los participantes comenzaron a realizar las especificaciones en ese mismo momento al final de la sesión de presentación y solicitaron ayuda para realizar la tarea. Nos limitamos a simplemente responder dudas respecto de la plantilla (*template*) y a como escribir requerimientos y describir símbolos del LEL, y pusimos especial atención a no responder ninguna pregunta sobre que escribir. Luego de esa sesión, los participantes solicitaron dos reuniones adicionales en la semana para realizar consultas.

La aplicación de las estrategias fue realizada por el autor de esta tesis. La especificación de requerimientos como así también la descripción de los símbolos del LEL fue entregada por los participantes y las estrategias fueron aplicadas manualmente. La estrategia Theme/Doc es relativamente sencilla de aplicar dado que solo es necesario realizar un diagrama, el cual se llevó a cabo sin ayuda de la computadora. Sin embargo, para aplicar la estrategia de LEL, fue necesario el uso de planilla de cálculo para organizar los cálculos.

4.3 Análisis

El análisis consiste de dos etapas. La primera etapa consiste en verificar que la estrategia del LEL identifica las mismas características transversales que Theme/Doc. En la segunda etapa, se verifica que las características transversales se identifican a través de la estrategia del LEL con mayor frecuencia que son identificados por Theme/Doc.

Después de aplicar la estrategia Theme/Doc a las 10 especificaciones de requerimientos producidas por los participantes, llegamos a obtener 4 diferentes características transversales en la suma de todas las características transversales identificadas por cada una de las especificaciones. Sólo un par de especificaciones de requerimientos nos permitió obtener las 4 características transversales a la vez. Las demás especificaciones sólo permitieron obtener algunos de esas 4 características transversales. De todas formas, tomamos las 4 características transversales como la base para comparar ambas estrategias. Las 4 características transversales son: (i) *visualización de pedidos*, (ii) *autorización para ejecutar comandos*, (iii) *logging de comandos* y (iv) *notificación de acciones*. En términos generales, *logging* fue la característica transversal más fácil de identificar al igual que *autorización para ejecutar comandos*. Es importante mencionar que en algunas especificaciones de requerimientos el nivel de detalle no fue lo suficientemente preciso, por lo cual, *visualización de pedidos* estaba incluido en *autorización para ejecutar comandos*, ya que la acción de visualizar era considerado un comando. Finalmente, *notificación de acciones* fue muy difícil de identificar. La tabla Tabla 4-4 muestra las características transversales identificadas por cada especificación de requerimientos.

La estrategia del LEL consiste en identificar principalmente a los símbolos estados como características transversales. El dominio de la aplicación utilizado en este experimento posee tres estados en los cuales un pedido puede encontrarse: *nuevo*, *asignado* y *finalizado*. En general, luego de aplicar la estrategia de LEL al LEL

construido por los participantes, el estado que fue identificado como característica transversal fue *asignado*. Por lo cual, con el fin de comparar las estrategias Theme/Doc y LEL, se tuvo que analizar las referencias en el interior del grupo *asignado*, esto es, se tuvieron que analizar las referencias entre los símbolos del grupo considerado característica transversal. Hay dos símbolos con un alto número de referencias: *visualización de pedidos* y *logueo de comandos*. Estas acciones fueron muy referenciadas ya que constituyen funcionalidad elemental y dispersa por toda la aplicación. *Visualización* esta relacionada con cada acción que la aplicación brinda al igual que *logueo de comandos*. Debido al gran número de referencias que estos símbolos poseen en todas las especificaciones, podemos considerar que ambos fueron identificados por todas las especificaciones. Luego, *notificación de acciones* fueron identificados por la mayoría de las especificaciones de LEL (6 de las 10 especificaciones lo permiten identificar). Finalmente, *autorización para ejecutar comandos* se ubica en cuarto lugar. La siguiente tabla resume las características transversales identificadas por cada especificación del LEL.

Tabla 4-4. Características transversales identificadas a través de las especificaciones de requerimientos

Especificación de requerimientos #	visualización de tickets	autorización para ejecutar comandos	logueo de comandos	notificación de acciones
1	X	X	X	
2		X	X	
3	X	X	X	X
4		X	X	
5		X		
6	X	X	X	
7	X	X	X	X
8		X	X	
9	X	X	X	
10			X	

Tabla 4-5. Características transversales identificadas a través de la especificación de LEL

Especificación de requerimientos #	visualización de tickets	autorización para ejecutar comandos	logueo de comandos	notificación of acciones
1	X		X	X
2	X	X	X	X
3	X		X	X
4	X	X	X	X
5	X		X	
6	X		X	
7	X	X	X	X
8	X		X	
9	X	X	X	
10	X	X	X	X

Realizamos dos análisis para comparar la frecuencia de características transversales identificadas. El primero consistió en comparar cuantas características transversales fueron identificadas por cada experimento. Mientras que la segunda comparación implicó analizar la frecuencia de identificación de cada característica transversal.

Con el fin de analizar el número de características transversales identificadas por cada especificación, calculamos el promedio de características transversales identificadas por las estrategias Theme/Doc y por la de LEL. La Tabla 4-6 muestra un promedio de 2,5 características transversales por especificación en Theme/Doc, mientras que la tabla Tabla 4-7 muestra un promedio de 3,1 características transversales por LEL en la estrategia de LEL.

Tabla 4-6. Promedio de características transversales identificadas por la estrategia Theme/Doc

Especificación de requerimientos #	Número de características transversales identificadas
1	3
2	2
3	4
4	2
5	1
6	3
7	4
8	2
9	3
10	1
Total	25
Promedio	2.5

Tabla 4-7. Promedio de características transversales identificadas por la estrategia de LEL

Especificación de requerimientos #	Número de características transversales identificadas
1	3
2	4
3	3
4	4
5	2
6	2
7	4
8	2
9	3
10	4
Total	31
Promedio	3.1

Sobre la comparación de frecuencia, debemos distinguir tres situaciones. En primer lugar, existe una característica transversal con una alta certeza, la cual fue identificada casi por igual por ambas estrategias: *logging*. Luego, hay dos características transversales que están relacionadas: *visualización* y *autorización*. Indicamos que están relacionadas porque se pueden considerar como dos formas de referirse a la misma funcionalidad. *Visualización* puede considerarse como una acción en particular que necesita *autorización*, pero también puede considerarse como la única acción que necesita *autorización* y todas las otras acciones dependen de ella. Creemos que esta es la razón por la cual en la estrategia Theme/Doc *autorización* ha sido identificada 9 veces y *visualización* 5 veces, mientras que en la estrategia de LEL ocurrió lo opuesto: *visualización* ha sido identificada 10 veces y *autorización* ha sido identificada 5 veces. La última situación a analizar involucra a la característica transversal *notificación*. La misma fue identificada sólo 2 veces por Theme/Doc. Sin embargo, la estrategia de LEL identificó a *notificación* 6 veces. Por lo cual, concluimos que la frecuencia de identificación de la estrategia de LEL es superior que la de Theme/Doc, siendo significativa la estrategia de LEL para identificar a *notificación* como característica transversal.

Tabla 4-8. Comparación de características transversales identificadas por cada estrategia

Característica transversal	Theme/Doc	LEL
Visualización	5	10
Autorización	9	5
Logging	9	10
Notificación	2	6

4.4 Conclusiones

El experimento muestra que la estrategia del LEL es más efectiva que Theme/Doc. En primer lugar tenemos que aclarar que ninguna de las dos técnicas fue 100% efectiva, esto se evidencia con el hecho de cada una de las técnicas fue aplicada con 10 sujetos distintos y para ninguna de las dos técnicas, los resultados obtenidos por los 10 sujetos fueron exactamente iguales. Es decir, no sucedió que Theme/Doc o que la estrategia de LEL hayan encontrado exactamente las mismas características transversales en todas las réplicas del experimento.

A pesar de que ninguna de las dos técnicas fue 100% efectiva, al menos ambas técnicas fueron consistentes en cuanto a las características transversales identificadas. Es decir, tanto Theme/Doc como la estrategia de LEL encontraron en conjunto y por separado cada una de ellas al mismo conjunto de características transversales. Es decir, las estrategias encontraron 4 características transversales que fueron las mismas para cada una de las dos estrategias.

Ahora bien, si bien las dos estrategias identificaron al mismo conjunto de 4 características transversales, el experimento mostró que la estrategia de LEL es más efectiva que Theme/Doc por diferentes motivos. Por un lado, si bien cada estrategia identificó al mismo conjunto de 4 características transversales, cada instancia del

experimento no encontró todas las características transversales a la vez. En el caso de Theme/Doc sólo 2 instancias encontraron las 4 características transversales, mientras que para la estrategia de LEL 4 estrategias encontraron las 4 características transversales. Luego, con 3 y 2 identificaciones de características transversales, tanto la estrategia del LEL como Theme/Doc encontraron la misma cantidad de veces, 3 para cada una de ellas. Y finalmente, en 2 instancias de Theme/Doc se encontraron sólo una características transversales, mientras que en la estrategia de LEL ninguna instancia encontró solo 1 característica transversal. Esto demuestra la mayor efectividad de la estrategia de LEL sobre Theme/Doc, que en promedio tuvo una identificación de 3.1 características transversales sobre 2.5 de Theme/Doc.

Esta superioridad de la efectividad también puede apreciarse si analizamos en cuantas instancias del experimento se identificó cada característica transversal. Hubo dos características las cuales fueron identificadas por las 10 instancias de la estrategia del LEL. Mientras que la estrategia de Theme/Doc no encontró ningún característica en las 10 instancias (aunque si encontró a 2 características en 9 de las 10 instancias).

Si bien el análisis cuantitativo muestra la superior efectividad de la estrategia de LEL por sobre Theme/Doc, intuitivamente se puede entender este hecho dado que la estrategia de LEL incluye el patrón de reconocimiento que es usado en Theme/Doc que consiste en analizar las relaciones de las acciones entre los requerimientos, e identifica la acción mas dispersa como característica transversal. La estrategia de LEL también considera a las acciones, sin embargo, utiliza los estados en lugar de analizar los requerimientos. Es decir, la estrategia de LEL identifica verbos los cuales están dispersos entre los diferentes estados. La organización entre estados es diferente de la organización entre requerimientos. Un estado puede relacionar a muchos más símbolos que las acciones que puede incluir un requerimiento. Esta organización favorece a Theme/Doc para identificar a más características transversales que la estrategia de LEL. A pesar de todo, hay cierta relación entre Theme/Doc y la estrategia de LEL. Los estados que la estrategia de LEL identifica como características transversales se corresponden gramáticamente con el participio pasado de los verbos que Theme/Doc identifica como características transversales. Por lo cual, mientras que Theme/doc organiza la información en requerimientos para identificar verbos como características transversales, la estrategia de LEL organiza la información en estados e identifica estos estados como características transversales. Si bien hay cierta relación entre ambas estrategias, la estrategia del LEL es más completa que Theme/Doc dado que la estrategia del LEL utiliza otras clases de símbolos como ser sujetos y objetos para realizar la identificación. Por lo tanto, si bien la estrategia de LEL puede requerir más esfuerzo que Theme/Doc, ya sea para producir el material a procesar o para el proceso mismo de conteo, la estrategia del LEL posee la ventaja de ser más precisa, por lo cual, el esfuerzo extra es aprovechado.

4.4.1 Amenazas a la validez (Treaths to validity)

Nos centramos en la validez interna y externa según la categorización de Cook [Cook 1979]. Selección es la principal amenaza a la validez interna. Selección se refiere a como alteran los resultados el efecto de la variación natural en la performance de los seres humanos. Con el fin de acatar esta amenaza hemos seleccionado un grupo de personas con habilidades muy diferentes, experiencia muy variada y las hemos seleccionado tanto de la industria como de la academia. En la sección 4.1.2 puede verse

el detalle de las características de los mismos. Más aún, hemos llevado a cabo una asignación de sujetos a técnicas (*treatments*: requerimientos o LEL) en forma azarosa.

La instrumentación es otra amenaza a la validez interna que nos preocupamos de atacar. Con este fin, le dedicamos especial atención a la preparación de los artefactos para el experimento, de modo de producir una descripción realista de una aplicación, la cual, está basada en una aplicación real.

Sjøberg et al. [Sjøberg 2002] sostienen que muchas de las amenazas a la validez externa son causadas por planteos artificiales en los experimentos. Ellos mencionan la importancia de tareas y sujetos realistas. Tareas realistas tienen que ver con el tamaño, la complejidad y la duración de las mismas. Nosotros, hemos planteado un experimento de mediana complejidad, pero sobre una aplicación real (ya que la aplicación se corresponde con un desarrollo real vinculada con el autor de esta tesis). Sujetos realistas tienen que ver con la selección de los sujetos que realizarán las tareas. Con el fin de atacar este problema, hemos seleccionado profesionales con experiencia real en el desarrollo de software. Ellos poseen un gran rango de experiencia y también diferentes habilidades (la mayoría son desarrolladores, pero también participaron *testers*, analistas, arquitectos y líderes de proyecto).

5 Soporte automatizado a la estrategia

El presente capítulo presenta una herramienta la cual implementa la estrategia de identificación de características transversales a través de la estrategia del LEL. 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.

El presente capítulo se organiza de la siguiente forma. En primer lugar se describe la motivación en la cual radica la necesidad de contar con una herramienta automatizada y también justifica el porque de la decisión de implementar una herramienta que solo realice los cálculos y se apoye en otra herramienta ya existente para la edición. Luego, se describe las características básicas de C&L, con el fin de dar un marco a TICCWL, la herramienta desarrollada para los cálculos. El capítulo termina con 3 secciones que describen la nueva herramienta. Uno de ellos describe la arquitectura, otro las tecnologías utilizadas y el último como usarlo.

5.1 Motivación

La necesidad y utilidad de un soporte automatizado radica en diversos motivos. Por un lado, el contar con un soporte automatizado permite realizar ajustes a la estrategia mientras se desarrolla. Si bien la estrategia tiene un fundamento sólido que determina su esencia, existen ciertos ajustes finos que un soporte automatizado permite realizar a la estrategia. Por otro lado, el soporte automatizado permite que la aplicación de la estrategia sea más efectiva, puesto que una aplicación manual de la estrategia puede ocasionar ciertos errores involuntarios, sin embargo, la aplicación automática y sistematizada reduce a cero los errores de cálculo que se pueden presentar. Finalmente, el tiempo de aplicación de la estrategia se ve reducido notablemente.

La estrategia consta de dos partes fundamentales. Por un lado es necesario construir el LEL y luego de ello, se deben realizar los cálculos. La realización de los cálculos es una tarea que requiere mucha dedicación y la herramienta lo realiza inmediatamente y el único esfuerzo que se dejaría al profesional es el de construir el LEL. Sin embargo, aquellas personas que construían el LEL para otros fines y no solamente para identificar características transversales con la estrategia propuesta en esta tesis, se ven más favorecidas. Puesto que ellos sólo tienen el costo extra de armar los grupos de símbolos y con sólo un pequeño esfuerzo que les demando armar los grupos, pueden lograr una propuesta de características transversales candidatas. Esta es una ventaja del hecho de que la herramienta se apoye en otra herramienta. En TICCWL se importa el LEL

construido desde C&L, por lo tanto la nueva herramienta solamente se ocupa de todo lo necesario para configurar el LEL, realizar los cálculos y mostrar los resultados, sin preocuparse de la edición del LEL en sí. Por lo tanto, el esfuerzo de construcción de TICCWL se concentró en desarrollar funcionalidad que no está construida, por lo cual, se pudo aprovechar ese esfuerzo para lograrlo de la mejor forma, es decir buscando performance y aspectos de usabilidad entre otras cosas. Además, el desarrollar solo la parte de los cálculos, permitirá que la herramienta se integre con C&L y que los usuarios, puedan utilizar una herramienta ya existente y que saben usar, por lo cual, no se les genera carga adicional para aprender una nueva herramienta para editar un LEL, cosa que ya sabían hacer con otra herramienta.

5.2 C&L

Esta herramienta permite realizar una edición colaborativa donde diferentes usuarios pueden editar simultáneamente el mismo símbolo y un administrador controla la edición. La herramienta está basada en la filosofía de software libre, por lo cual, su código fuente está disponible para que cualquier persona interesada puede hacer contribuciones a la misma. La arquitectura modular de C&L permite el agregado de nuevos *plug-ins* al ambiente los cuales proveerán funcionalidad de una forma transparente como si fuera funcionalidad nativa del sistema.

C&L provee un ambiente colaborativo para crear, editar y gestionar tanto símbolos del LEL como Escenarios. Para esta tesis sólo es de interés el LEL, el cual está naturalmente organizado en un hipertexto. C&L implementa los referencias (*links*) entre los símbolos del LEL y también hacia los Escenarios. Estas referencias (*links*) se crean automáticamente cuando el usuario describe un símbolo e ingresa una palabra la cual ya se encuentra definida dentro del LEL.

C&L implementa dos niveles de acceso al sistema: nivel de usuario y nivel de administrador. Ambos roles están habilitados para ver, crear y eliminar símbolos. Sin embargo, la edición de los usuarios necesita ser aprobadas por el administrador para que pueda estar disponible para el resto de los usuarios. La siguiente figura muestra una pantalla de edición de un símbolo.



Figura 5-1. Pantalla de edición de un símbolo de LEL en C&L.

Aunque C&L provee la posibilidad de adicionar *plug-ins* para extender su funcionalidad, C&L también provee otra opción para compartir proyectos. Permite exportar toda la información de un proyecto a un archivo XML. Este archivo puede ser

construido con un formato específico definido por un archivo XSL, o también puede ser construido sin ningún tipo de formato. La siguiente tabla, resume la funcionalidad de C&L.

Tabla 5-1. Funcionalidad de C&L.

Funcionalidad general
Creación de proyectos con sus administradores
Asociación de usuarios a proyectos
Verificar y aprobar / denegar solicitudes de edición de símbolos
Verificar y aprobar / denegar solicitudes de edición de Escenarios
Verificar y aprobar / denegar solicitudes de edición en links
Funcionalidad de edición de LEL
Crear, editar y eliminar
Definición automática de links
Verificar consistencia luego de eliminar símbolos
Funcionalidad de edición de Escenarios
Crear, editar y eliminar
Definición automática de links
Verificar consistencia luego de eliminar símbolos
Funcionalidad de Input / Output
Exportar e importar XML de proyectos
Crear y ver historial de Ontologías de proyectos en DAML+OIL

5.3 Arquitectura

TICCWL sólo provee soporte para identificar características transversales, esto es, la herramienta realiza los cálculos necesarios y muestra los resultados con el fin de hacer ajustes de configuración de símbolos y grupos para volver a realizar los cálculos. Sin embargo, la herramienta no provee ningún tipo de funcionalidad para crear o editar el LEL. El LEL debe ser creado usando C&L y luego de que el LEL es completamente creado, debe ser exportado a un archivo XML. La Figura 5-2 muestra un ejemplo de un archivo XML generado por C&L, el cual debe ser provisto a TICCWL.

El ejemplo es muy sencillo. Sólo posee 2 símbolos: *client* y *account*. Cada símbolo posee un *id*, noción (*notion*) e impactos (*behavioural responses*). Las descripciones incluyen referencias al otro símbolo.

La herramienta de identificación importa el XML y vuelve a construir el LEL para permitir realizar las 3 actividades necesarias para la identificación de las características transversales. En primer, lugar el usuario debe asociar cada símbolo a un estado, y tiene la posibilidad de habilitar o deshabilitar cada símbolo para que sea considerado en los cálculos. Luego de esta configuración, se desencadena el cálculo para la identificación de características transversales. Una vez que los cálculos fueron realizados, el usuario puede navegar los resultados, esto es, puede ver los valores de referencias y como se distribuyen. Luego, puede hacer modificaciones de la configuración que realizó inicialmente de vinculación de símbolos con estados, como así también de habilitar o deshabilitar símbolos. Luego de ello, es posible lanzar nuevamente los cálculos para identificar las características transversales. La Figura 5-3 ilustra estas etapas sin embargo, las mismas se describen con más detalle en una sección posterior.


```

<project>

<lexicon>
<id_lexicon>1</id_lexicon>
<name>client</name>
<notion>
    Person that opens an <lexicon_link target= 2> account
    </lexicon_link> with the bank.
</notion>
<behavioral_response>
    The client can deposit money into his <lexicon_link target= 2>
    account </lexicon_link>.
    The client can withdraw money from his <lexicon_link target= 2>
    account </lexicon_link>.
    The client can consult balance from his <lexicon_link target= 2>
    account t</lexicon_link>.
</behavioral_response>
<classification>subject</classification>
</lexicon>

<lexicon>
<id_lexicon>2</id_lexicon>
<name>account</name>
<notion>
    The account has a balance and the <lexicon_link target= 1> client
    </lexicon_link> can operate on it.
</notion>
<behavioral_response>
    The <lexicon_link target= 1> client </lexicon_link> can deposit
    money into his account.
    The <lexicon_link target= 1> client </lexicon_link> can withdraw
    money from his account.
    The <lexicon_link target= 1> client </lexicon_link> can consult
    balance from his account.
</behavioral_response>
<classification>object</classification>
</lexicon>

</project>

```

Figura 5-2. Ejemplo de archive XML generado por C&L

5.4 Tecnologías utilizadas

TICCWL fue construida con todas tecnologías libres. Se utilizó Lua [Lua 2011] para desarrollar el núcleo (*core*) de la aplicación y se ha utilizado el *framework* Spring MVC [Spring 2011] para desarrollar la vista, la cual fue enriquecida con controles provistos por JQuery [JQuery 2011]. Luego, con el fin de integrar la herramienta con C&L, se utilizó XML [XML 2011] y se utilizó Maven [Maven 2011] para integrar todos los módulos de la aplicación, la cual se ejecuta en un servidor Apache / Tomcat [Apache 2011].

El núcleo (*core*) de la aplicación está implementado en Lua dado que se pretende integrar a la herramienta con C&L. Lua es un lenguaje de *scripting* el cual combina sintaxis simple del paradigma procedural con poderosos constructores de descripción de datos basados en arrays asociativos y semántica extendible. Lua es de tipos dinámicos,

corre a través de interpretación de *bytecodes* y posee gestión automática de memoria con un *garbage collector* incremental.

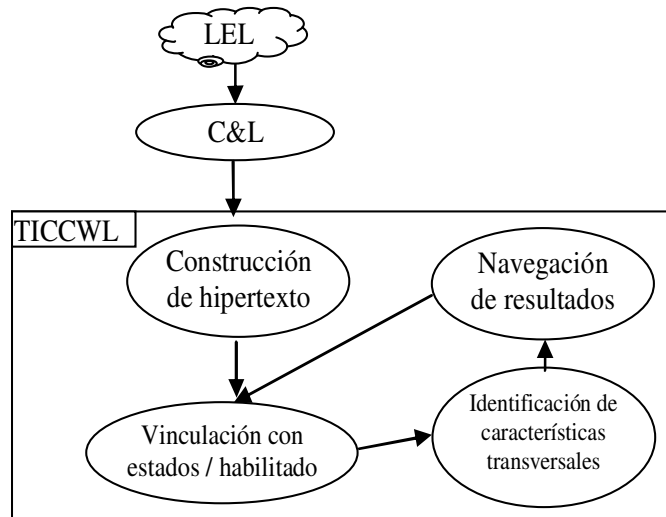


Figura 5-3. Etapas de ejecución.

La vista de la aplicación está implementada con el Sprint MVC, un Framework basado en *request*. El Framework define interfaces para todas las responsabilidades que deben ser manejadas. Luego, los usuarios de Spring MVC deben escribir sus propias implementaciones si lo necesitan. Todas las interfaces están altamente relacionadas con la API de *servlet*, lo cual, asegura que las mismas se mantengan disponibles, mientras que ofrece un *framework* de un nivel abstracto para facilitar el uso de la API. La facilidad para *testear* las implementaciones de las interfaces es una importante ventaja del nivel de abstracción ofrecido por Spring MVC.

Para interactuar con la aplicación se utiliza controles provistos por JQuery. En particular se encontró que el control *droppable* es muy útil para la aplicación. JQuery es una librería de javascript la cual simplifica la forma de interactuar con documentos HTML, manipular árboles DOM, gestionar eventos, desarrollar animaciones e incorporar interacciones AJAX.

Con el fin de comunicarse con C&L la herramienta utiliza XML para transportar la información. *Extensible Markup Language* (XML) comprende un conjunto de reglas para codificar documentos de una forma entendible por la computadora. La W3C produjo una especificación XML 1.0 la cual sienta las bases para la construcción de documentos XML. Los mismos son archivos de texto con un fuerte soporte de *Unicode* para facilitar el uso por los distintos lenguajes. Aunque el diseño de XML se centra en documentos, es ampliamente utilizado para la representación de estructuras de datos, por ejemplo, servicios web.

Maven es una herramienta para la gestión de proyectos y la construcción automática de los mismos. *Maven* es apadrinado por la *Apache Software Foundation*, donde inicialmente era una parte del proyecto *Jakarta*. *Maven* utiliza construcciones conocidas como *Project Object Model* (POM) para describir el proyecto de software que se debe construir, sus dependencias en módulos externos y componentes, y el orden de construcción. Contiene tareas predefinidas para realizar compilaciones y

empaquetamiento de módulos. *Maven* dinámicamente obtiene de Internet librerías Java y *plug-ins* desde uno o varios repositorios, y también permite subir artefactos a repositorios específicos luego de construcciones exitosas.

La aplicación se ejecuta en un servidor *Apache Tomcat* (también denominado *Jakarta Tomcat* o simplemente *Tomcat*), un contenedor de *servlet open source*, desarrollado por *Apache Software Foundation (ASF)*. *Tomcat* implementa las especificaciones de *Java Servlets* y *JavaServer Pages (JSP)* de *Oracle Corporation*, y provee un *Server HTTP* puro de Java para correr la aplicación. *Tomcat* no debe ser confundido con *Apache Web Server*, el cual es una implementación en C del servidor web de HTTP. Estos dos servidores no se encuentran ofrecidos en forma conjunto, sin embargo, se los suele montar en conjunto

5.5 Uso

La herramienta sólo provee soporte para calcular las referencias con el fin de identificar características transversales, sin embargo, la herramienta no provee asistencia para la edición del LEL, la cual se debe realizar con C&L. Por lo cual, una vez construido el LEL, se debe exportar a un archivo XML con el fin de importarlo en TICCW (Figura 5-4).

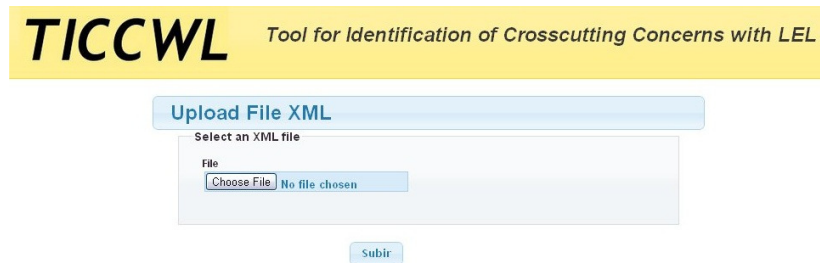
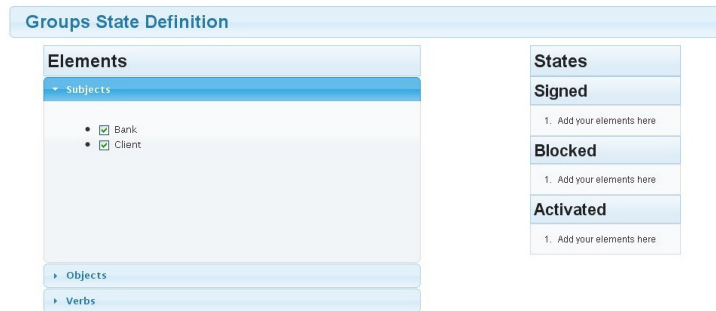


Figura 5-4 Pantalla para cargar un archivo XML

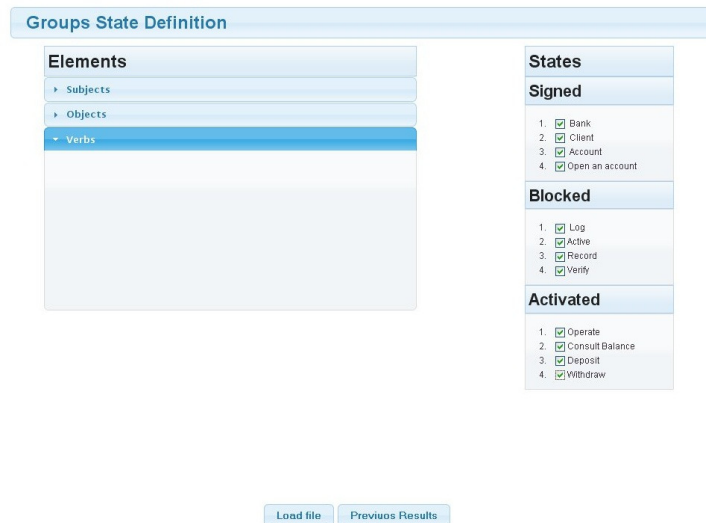
Hay dos tareas que el usuario debe realizar antes de hacer los cálculos que permiten identificar las características transversales. El usuario debe relacionar cada símbolo con uno y sólo un símbolo estado. Luego, el usuario puede identificar los símbolos principales dentro del dominio de la aplicación, los cuales no deben ser considerados para contar las referencias que ellos reciben desde otros símbolos, pero si deben ser considerados para contar las referencias hacía otros símbolos.

Con el fin de relacionar cada símbolo con un estado, el usuario debe arrastrar el símbolo desde la columna de la izquierda y lo debe soltar en un estado de la columna de la derecha. Figura 5-5 muestra la situación inicial en donde ningún símbolo fue arrojado en ningún estado. Se debe apreciar también, que cada símbolo posee un *checkbox* para habilitar o deshabilitar los símbolos respecto del conteo de sus referencias. Figura 5-6 muestra todos los símbolos relacionados con sus estados.



Load file Previous Results

Figura 5-5. Pantalla inicial para agrupar símbolos en estados.



Load file Previous Results

Figura 5-6. Algunos símbolos agrupados en estados.

Una vez que los símbolos fueron configurados, esto es, una vez que cada símbolo fue relacionado con un estado y fueron deshabilitados los símbolos necesarios, es posible desencadenar el cálculo de referencias para poder identificar características transversales. La primera pantalla luego de realizar los cálculos muestra las características transversales ordenadas por posibilidad de ser consideradas como tales, desde la más posible hasta la menos. El orden es determinado a través del producto entre el número de grupos que los referencias y los promedios generales (Figura 5-7).

Final Order		
State Name	Number of groups that have references	General average of references
Signed	2	5.6
Blocked	2	2.4
Activated	2	2.0

[Previous results](#) [State Definition](#) [View Chart](#)

Symbol name	Signed	Blocked	Activated

Figura 5-7 Características transversales candidatas ordenadas por posibilidad de ser consideradas como tales. Además, es posible mostrar la información de un diagrama XY, el cual muestra la dispersión de los grupos en función de las dos variables (Figura 5-8).

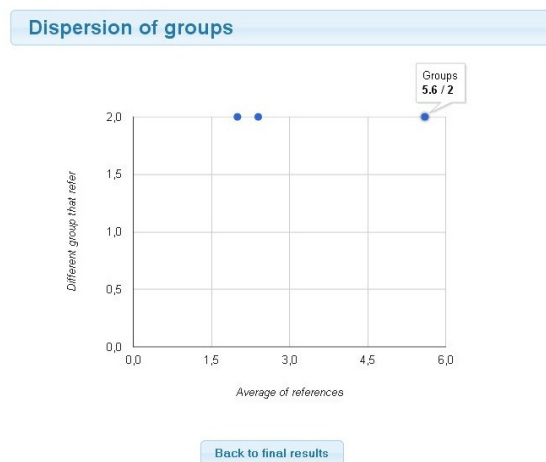


Figura 5-8. Diagrama XY que muestra la dispersión de los grupos.

Luego, es posible obtener más detalle, respecto de los promedios por cada categoría de símbolo (Figura 5-9).

References Counting					
State Name	Number of groups that have references	General average of references	Average of references to subjects	Average of references to objects	Average of references to verbs
Signed	2	5.6	9.0	10.0	0.0
Blocked	2	2.4	0.0	4.0	2.7
Activated	2	2.0	0.0	0.0	2.3

[State Definition](#) [Final Results](#)

Figura 5-9. Reporte detallado de referencias por grupos

También es posible analizar en detalle el número de referencias por símbolo (Figura 5-10) con el fin de determinar si hay algún símbolo que tiene valores altos y es central (*core*), por lo cual, debería ser deshabilitado y realizar los cálculos nuevamente.



Figura 5-10 Reporte de referencias por símbolos.

6 Conclusiones

El presente capítulo ofrece un resumen de las contribuciones de esta tesis, como así también de los beneficios de aplicar la estrategia y también las limitaciones que posee. Finalmente, el capítulo describe los trabajos futuros.

6.1 Contribuciones

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. 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. Por un lado se describe una pequeña aplicación bancaria que se utiliza para ejemplificar la estrategia y su aplicación. Por otro lado, se describen dos casos de estudios referentes aplicaciones que se podrían categorizar como entre mediana y gran magnitud. Ambos casos de estudios están basados en aplicaciones reales. Por un lado una aplicación de control antievasión de impuestos y por otro lado un portal web que publica noticias. Además, la presente tesis también describe un experimento que se llevó a cabo con el fin de mostrar la efectividad de la estrategia. El experimento fue planteado suficientemente realista y con una población de 20 sujetos como para obtener un muestreo de resultados suficientes. Por último, la tesis describe una herramienta que permite asistir en la aplicación de la estrategia propuesta.

6.2 Beneficios

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 (Léxico Extendido del Lenguaje). 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] 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”.

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.

La herramienta 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 para la edición del LEL. Es decir, la herramienta debe recibir el LEL construido a través de C&L [C&L 2009] y una vez incorporado, 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.

En este punto, pueden apreciarse dos beneficios más de la estrategia. Por un lado, el esquema de cálculos se basa principalmente en la construcción del LEL. LEL que es una herramienta propuesta desde hace varios años y en uso como se mostró con las referencias previas. Por lo cual, aquellos profesionales que utilizaban el LEL para sus tareas de análisis se benefician con esta estrategia porque sólo deben configurar el mismo para que luego la herramienta realice los cálculos automáticamente. 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.

Por otro lado, la herramienta implementada para realizar los cálculos, sólo se ocupa de realizar los mismos y el LEL se debe construir en otra herramienta ya construida y disponible hace tiempo [C&L 2009]. Por lo cual, los profesionales que quieran utilizar la estrategia, ni siquiera poseen la necesidad de aprender a utilizar una nueva herramienta para construir el LEL, ya que lo construyen con la herramienta que utilizaban.

La aplicación de la estrategia requiere gran precisión en dos etapas: la identificación de links en la descripción de los símbolos y el conteo de referencias que esos links determinan. La estrategia se basa en contar las referencias de los links de los distintos símbolos que conforman el LEL. Por lo cual, durante la construcción del LEL y en la descripción de los símbolos es necesario no omitir ninguna referencia, y también es crítico hacer los cálculos correctamente. Sin embargo, ambas tareas están automatizadas, ya que la construcción del LEL recae en la herramienta C&L [C&L 2009] mientras que los cálculos están implementadas en otra aplicación [Aramayo 2012]. De esta forma, estas dos tareas tan críticas están soportadas por herramientas y el profesional se puede concentrar en entender el lenguaje del contexto y describirlo a través del LEL.

La descripción producida por el LEL, a pesar de ser simple, es muy rica y permite lograr un nivel de detalle tal que incluso podría no estar en productos posteriores del desarrollo del software. Durante la construcción de los distintos productos se suele abstraer la información. Esto ocurre en productos de requerimientos, arquitectura y diseño por ejemplo. Sin embargo, al llegar a la codificación es necesario plasmar en forma explícita todo el conocimiento que implícitamente está representado por los requerimientos, modelos de arquitectura y diseños de alto nivel. Sin embargo, el LEL permite construir descripciones por demás detalladas y precisas. Por su naturaleza, la de capturar el lenguaje y por los dos principios que lo rigen: el de circularidad y el de vocabulario mínimo, el LEL termina siendo una descripción completa y autocontenida del lenguaje del contexto de la aplicación que no debiera omitir detalle. Sin embargo, estos detalles pasan a ser implícitos cuando se pasan a requerimientos, reglas de negocio, diseños, etc... Entonces, el análisis que se pueda realizar en etapas posteriores estará basado en modelos que poseen información parcial por un lado y por otro en el conocimiento del profesional que realiza el análisis. Sin embargo, en el caso de la estrategia del LEL, la estrategia se basa en un producto explícito y concreto que puede ser revisado y validado para asegurar la correctitud y completitud.

Además, considerando que el LEL es un modelo del lenguaje del contexto de la aplicación, la estrategia propuesta lleva la identificación de características transversales

a una etapa muy temprana del desarrollo del software. Esta información estratégica en una etapa tan temprana permite tomar decisiones arquitecturales bien críticas, ya que introducir cambios en el medio del desarrollo del software puede ser difícil y algunas veces, los cambios necesarios pueden ser tan grandes que se consideran imposibles de aplicar. 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. Más aún, Mizuno desarrolló un modelo de “*catarata de errores*” [Mizuno 1983] en el cual, establece que en cada etapa, la posibilidad de que ocurran errores es mayor que en la etapa previa. Ya que cada etapa se basa en los productos de la etapa previa, la descripción producida en una etapa podría contener un error y al trabajarse con el mismo en la siguiente etapa, dado que se está trabajando a partir de una descripción con errores, aunque el trabajo y el proceso realizado sea correcto, no lo será la descripción, ya que estará basado en descripciones incorrectas y la descripción resultante también lo será. Por estos motivos, es necesario tomar las decisiones lo antes posible en el desarrollo del software y no introducir cambios en el medio. Dado que la estrategia es aplicada antes de la etapa de especificación de requerimientos, el ingeniero de requerimientos puede usar esta información para identificar características transversales antes de escribir los requerimientos y así poder guiar el proceso de elicitación y especificación.

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. Los dos casos de estudio mostrados ejemplifican esta virtud de la estrategia. El caso de estudio de la aplicación antievasión de impuestos ilustra como a través del contexto de la aplicación se puede construir un LEL y aplicar la estrategia. Mientras que el caso de estudio de la aplicación del portal web muestra como a partir de una aplicación (codificada o al menos diseñada) se puede construir el LEL y luego aplicar la estrategia.

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. Los verbos son los que más ayudan para identificar características transversales; sin embargo, sujetos, objetos y estados ayudan para confirmar las características transversales identificadas por los verbos, o incluso permiten identificarlos aunque los verbos no lo permitan. En una descripción coloquial, podría suceder que una misma acción se describa a través de distintos verbos (sinónimos) pero durante el análisis no se lo detecta. Sin embargo, si la acción recae en el mismo objeto, este hecho puede brindar la información necesaria para identificar la característica transversal.

6.3 Limitaciones

La estrategia posee varias limitaciones, algunas propias de la aplicación que se pretende analizar y otras propias de la construcción del LEL.

En primer lugar, existe una dependencia de la estrategia con la estructura de la aplicación a analizar. La estrategia considera que la aplicación posee una y sola una

máquina de estados que sintetiza el comportamiento principal. Es decir, luego de definido el LEL y antes de realizar los cálculos de las referencias, la estrategia demanda identificar los estados principales de la aplicación y agrupar los símbolos en esos estados. De esta forma, logramos un modelo de alto nivel de la aplicación, en donde los estados representan las distintas partes del mismo y los distintos símbolos se agrupan entre los estados con el fin de caracterizarlos. Como los símbolos solamente se deben vincular con un solo estado para la estrategia, puede suceder que los símbolos de alguna forma tengan relación con más de uno para la aplicación. Esta situación es la que determina que ciertos estados son características transversales. Por lo cual, la estrategia se basa en un conjunto de estados que describen una única máquina de estados y no hemos analizado si es posible que una aplicación tenga más de una máquina de estados que la sintetice o si pudiera no tener ninguna. Si esto fuera posible, tendríamos que analizar como ajustar la estrategia para que funcione en estos casos.

Luego, 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.

6.4 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 estamos trabajando en construir guías bien precisas sobre como construir el LEL. Con el fin de construir estas guías estamos realizando un experimento en donde 20 grupos de participantes construyen un LEL para una aplicación única. A partir del proceso de construcción de LEL y de los resultados producidos estamos generando guías sobre como deben realizarse la descripción. Guías que sirven tanto para enriquecer la construcción de un LEL que se utilice o no para identificar características transversales. Luego, no sólo estamos trabajando en escribir las guías que prescriben como construir el LEL, sino que también estamos trabajando en desarrollar una métrica que una vez construido el LEL permite dar una medida de tamaño del LEL producido como así también una medida de calidad del mismo.

Por otro lado, la identificación de características transversales a partir del LEL es una de las patas 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 es enriquecida con las métricas que mencionamos y que esperamos encontrar una relación con otras métricas como *use case points* por ejemplo.

Ya se han indicado de otros trabajos en donde se han mostrado lo efectivo que resulta el hecho de comprender el lenguaje de la aplicación a partir del LEL. Si a partir del LEL logramos producir todos los productos indicados con la misma cantidad de información, precisión y calidad que el LEL mostró poseer, estaríamos brindando información muy valiosa para desarrollar software. Entiéndase por desarrollar información sobre el producto que se desea construir, pero también información que permitirá planificar la construcción del mismo. Y toda esta información, será obtenida a partir del LEL, el cual ha mostrado su efectividad en dominios complejos y con usuarios no expertos en tales técnicas. Por lo cual, creemos que este desarrollado guiado a partir del LEL será de mucha ayuda para la ingeniería de software.

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.
- [Apache 2011] Apache/Tomcat available at <http://tomcat.apache.org/> accessed in October, 2011.
- [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.
- [AspectJ 2011] The AspectJ project, available at <http://www.eclipse.org/aspectj/>, accessed in February 2011.
- [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.
- [Basili 1994] Basili V., Gianluigi C., Dieter Rombach H., "The Goal Question Metric Approach", 1994.
- [Boehm 1997] Boehm B. W., Software Engineering, Computer society Press, IEEE, 1997.
- [Bounour 2006] Bounour N., Ghouil 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.
- [Brooks 1995] Brooks F., The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley Professional, 2 edition, 1995.
- [C&L 2009] C&L Tool, available at <http://pes.inf.puc-rio.br/cel/>, accessed in October, 2009.
- [Clarke 2005] Clarke S., Baniassad E., “Aspect-Oriented Analysis and Design The Theme Approach” Addison Wesley, ISBN: 0321246748, 2005.
- [Cook 1979] Cook T. D., Campbell D. T., “Quasi-experimentation: Design and analysis for field settings”, Chicago: Rand McNally, 1979.

- [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.
- [jQuery 2011] JQuery available at <http://jquery.com/> accessed in October, 2011.
- [Kaplan 2000] Kaplan G., Hadad G., Doorn J., Leite J.C.S.P., “Inspeccion del Lexico Extendido del Lenguaje”, In: proceedings of the Workshops de Engenharia de Requisitos, Wer’00, Rio de Janeiro, Brazil, 2000.
- [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.
- [Leite 2005] Leite J.C.S.P., Silva L.F., Breitman K.K., “C&L: Uma Ferramenta de Apoio à Engenharia de Requisitos”, In: RITA 122.: ISSN 0103-4308, Revista de Informática Teórica e Aplicada (RITA), Vol. XII, Número 1, Junho, pp 23-46, 2005.
- [Lua 2011] Lua, available at <http://www.lua.org/portugues.html> accessed in October, 2011.
- [Mahoney 2005] Mahoney M., “Modeling Crosscutting Concerns in Reactive Systems with Aspect-Oriented”, 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.
- [Maven 2011] Maven, available at <http://maven.apache.org/> accessed in October, 2011.
- [Mizuno 1983] Mizuno Y., "Software Quality Improvement," IEEE Computer, Vol. 16, No. 3, March, pp. 66 – 72, 1983.
- [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.
- [Rashid 2008] Rashid A., Chitchyan R., “Aspect-oriented requirements engineering: a roadmap”, In: proceeding of the international Conference on Software Engineering, Proceedings of the 13th international workshop on Early Aspects, ISBN:978-1-60558-032-6, Leipzig, Germany, pp 35-41, 2008.
- [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.
- [Sampaio 2005b] Sampaio A., Chitchyan R., Rashid A., Rayson P., “EA-Miner: A Tool for Automating Aspect-Oriented Requirements Identification” Proc. Int’l Conf. Automated Software Eng. (ASE 05), ACM Press, pp. 352–355, 2005.
- [Silva 2005] Silva, L., Leite, J.C.S.P., “Uma linguagem de modelagem de requisitos orientada a aspectos”, Proceedings of the Requirement Engineering Workshop at CAiSE 2005, Porto-Portugal, 2005.
- [Shull 2008] Shull F., Singer J., Sjøberg D.I.K., Guide to Advanced Empirical Software Engineering, ISBN-13: 978-1-84800-043-8, 2008.
- [Sjøberg 2002] Sjøberg D.I.K., Anda B., Arisholm E., Dybå T., Jørgensen M., Karahasanovic A., Koren E. F., Vokác M., “Conducting Realistic Experiments in Software Engineering”, In Proc. 1st Int. Symposium on Empirical Software Engineering. 2002.
- [Sousa 2004] Sousa G., Soares S., Borba P., Castro J., “Separation of Crosscutting Concerns from Requirements to Design: Adapting an Use Case Driven Approach”, Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design, workshop of the 3rd International Conference on Aspect-Oriented Software Development, Lancaster, UK, 22-26, March 2004.
- [Spring 2011] SpringMVC, available at <http://www.springsource.org/> accessed in October, 2011.
- [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.
- [XML 2011] XML, available at <http://www.w3.org/TR/REC-xml/> accessed in October, 2011.
- [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.

8 Anexos

El presente anexo incluye información más detallada sobre los casos de estudio y del experimento de validación realizado. La primera sección incluye información del caso de estudio antievasión de impuestos, la siguiente sección incluye información del caso de estudio del portal web y finalmente la última sección incluye información del experimento.

8.1 Descripción caso de estudio antievasión de impuestos

A continuación se enumera la lista completa de símbolos que fueron identificados y descriptos para el caso de estudio antievasión de impuestos. Para la misma se identificaron 63 símbolos en total, mientras que por categoría las cantidades son las siguientes: 9 estados, 14 sujetos, 12 objetos y 28 verbos.

Tabla 8-1. Lista completa de símbolos definidos para el caso de estudio antievasión de impuestos

Estado	Sujetos	Objetos	Verbos
Establecimiento de la deuda	Contribuyente Repartición Deudor	Impuesto Deuda Título Comprobante	Pagar Contraer deuda Intimar prejudicial Crear título Cobrar deuda
Procedimiento a nivel administrativo	Demandante Registro Registro propiedad Registro automotor	Bien Bien mueble Bien inmueble	Evaluar costo beneficio
Procedimiento a nivel judicial	Juez Abogado Apoderado demandante Delegado demandante	Expediente judicial Carta documento Juicio	Paso Judicial Iniciar Intimar Obtener sentencia
Procedimiento a nivel cautelar			Renovar embargo Realizar remate
Cancelación de la deuda	Banco	Dinero	Calcular monto a abonar Pagar en efectivo Verificar que se haya efectuado el pago Emitir comprobante de pago de plan de facilidades de pago
Medidas preventivas			Trabar medida preventiva administrativa Trabar medida preventiva Reinscribir medida preventiva administrativa Levantar medida preventiva administrativa Levantar medida preventiva

Plan de facilidades de pago			Entrar Salir Pagar cuota
Subasta	Martillero Participante del remante	Precio base	Designar martillero Publicar subasta Rematar
Deuda reclamada por error			Presentar comprobantes de pago

Luego, se muestra una tabla para cada uno de los grupos, en la cual se detallan los símbolos de ese grupo, separados por categorías y para cada uno de ellos se detallan las referencias que poseen desde otros grupos.

Tabla 8-2. Detalle de símbolos y sus referencias para el grupo de establecimiento de deuda

	Establecimiento de deuda	Procedimiento a nivel administrativo	Procedimiento a nivel judicial	Procedimiento a nivel cautelar	Cancelación de la deuda	Medidas preventivas	Plan de facilidades de pago	Subasta	Deuda reclamada por error
Estados									
Establecimiento de deuda					1				
Sujetos									
Contribuyente									
Repartición									
Deudor		1	4	2	9	8	6		2
Objetos									
Impuesto									
Deuda			1	2					
Título			2						
Comprobante							1		3
Verbos									
Pagar									
Contraer deuda									
Intimar prejudicial			1	1					
Crear título									
Cobrar deuda		1		3					

Tabla 8-3. Detalle de símbolos y sus referencias para el grupo de procedimiento a nivel administrativo

	Establecimiento de deuda	Procedimiento a nivel administrativo	Procedimiento a nivel judicial	Procedimiento a nivel cautelar	Cancelación de la deuda	Medidas preventivas	Plan de facilidades de pago	Subasta	Deuda reclamada por error
Estados									
Procedimiento a nivel	1								

administrativo									
Sujetos									
Demandante	1			2	3	6	1	4	2
Registro				1		3			
Registro propiedad				1					
Registro automotor									
Objetos									
Bien				4		2		8	
Bien mueble									
Bien inmueble									
Verbos									
Evaluar costo - beneficio									

Tabla 8-4. Detalle de símbolos y sus referencias para el grupo de procedimiento a nivel judicial

	Establecimiento de deuda	Procedimiento a nivel administrativo	Procedimiento a nivel judicial	Procedimiento a nivel cautelar	Cancelación de la deuda	Medidas preventivas	Plan de facilidades de pago	Subasta	Deuda reclamada por error
Estados									
Procedimiento a nivel judicial		1			1		2		
Sujetos									
Juez									
Abogado	1			3	2		2		
Apoderado demandante									
Delegado demandante									
Objetos									
Expediente judicial									
Carta documento									
Juicio	1	1			1	2	1	3	1
Verbos									
Paso judicial									
Iniciar		2							
Intimar									
Obtener sentencia									

Tabla 8-5. Detalle de símbolos y sus referencias para el grupo procedimiento a nivel cautelar

	Establecimiento de deuda	Procedimiento a nivel administrativo	Procedimiento a nivel judicial	Procedimiento a nivel cautelar	Cancelación de la deuda	Medidas preventivas	Plan de facilidades de pago	Subasta	Deuda reclamada por error
Estados									
Procedimiento a nivel cautelar			1					1	
Sujetos									
Ninguno									
Objetos									
Ninguno									
Verbos									
Renovar embargo Realizar remate									

Tabla 8-6. Detalle de símbolos y sus referencias para el grupo cancelación de la deuda

	Establecimiento de deuda	Procedimiento a nivel administrativo	Procedimiento a nivel judicial	Procedimiento a nivel cautelar	Cancelación de la deuda	Medidas preventivas	Plan de facilidades de pago	Subasta	Deuda reclamada por error
Estados									
Cancelación de la deuda									
Sujetos									
Banco									
Objetos									
Dinero									
Verbos									
Calcular monto a abonar							1		
Pagar en efectivo	2			1		1			
Verificar que se haya efectuado el pago									
Emitir comprobante de pago del plan de facilidades de pago									

Tabla 8-7. Detalle de símbolos y sus referencias para el grupo de Medidas preventivas

	Establecimiento de deuda	Procedimiento a nivel administrativo	Procedimiento a nivel judicial	Procedimiento a nivel cautelar	Cancelación de la deuda	Medidas preventivas	Plan de facilidades de pago	Subasta	Deuda reclamada por error
Estados									
Medidas preventivas									
Sujetos									
Ninguno									
Objetos									
Ninguno									
Verbos									
Trabar medida preventiva administrativa		2					1		
Trabar medida preventiva		4	1	1					
Reinscribir medida preventiva administrativa							1		
Levantar medida preventiva administrativa									
Levantar medida preventiva							1		

Tabla 8-8. Detalle de símbolos y sus referencias para el grupo Plan de facilidades de pago

	Establecimiento de deuda	Procedimiento a nivel administrativo	Procedimiento a nivel judicial	Procedimiento a nivel cautelar	Cancelación de la deuda	Medidas preventivas	Plan de facilidades de pago	Subasta	Deuda reclamada por error
Estados									
Plan de facilidades de pago									
Sujetos									
Ninguno									
Objetos									
Ninguno									
Verbos									
Entrar									
Salir									
Pagar cuota	2		1	1		1			

Tabla 8-9. Detalle de símbolos y sus referencias para el grupo de Subasta

	Establecimiento de deuda	Procedimiento a nivel administrativo	Procedimiento a nivel judicial	Procedimiento a nivel cautelar	Cancelación de la deuda	Medidas preventivas	Plan de facilidades de pago	Subasta	Deuda reclamada por error
Estados									
Subasta				1					
Sujetos									
Martillero									
Participante de la subasta									
Objetos									
Precio base									
Verbos									
Designar martillero									
Publicar subasta									
Subastar		3		3					

Tabla 8-10. Detalle de símbolos y sus referencias para el grupo Deuda reclamada por error

	Establecimiento de deuda	Procedimiento a nivel administrativo	Procedimiento a nivel judicial	Procedimiento a nivel cautelar	Cancelación de la deuda	Medidas preventivas	Plan de facilidades de pago	Subasta	Deuda reclamada por error
Estados									
Deuda reclamada por error	1		1	1	1				
Sujetos									
Ninguno									
Objetos									
Ninguno									
Verbos									
Presentar comprobantes de pago	2								

8.2 Descripción caso de estudio portal web

A continuación se enumera la lista completa de símbolos que fueron identificados y descriptos para el caso de estudio portal web. Para la misma se identificaron 58 símbolos en total, mientras que por categoría las cantidades son las siguientes: 9 estados, 5 sujetos, 16 objetos y 28 verbos.

Tabla 8-11. Lista completa de símbolos definidos para el caso de estudio portal web

Estado	Sujetos	Objetos	Verbos
--------	---------	---------	--------

Redactado	Redactor Editor Gestor	Contenido Titulo Desarrollo	Crear contenido Aceptar contenido Rechazar contenido
Rechazado			Limpiar historia
Aceptado	Usuario final Publicador		Publicar contenido Construir versiones publicables del contenido
HTML construido		HTML Plantilla Black Word list White Word list	Crear HTML Definir nombre Formatear contenido Quitar símbolos Quitar artículos Quitar Black Priorizar White Recortar desarrollo
Versiones auxiliares construidas		PDF Imprimible Mobile SMS Formato básico Formato auxiliar	Construir auxiliar Construir PDF Construir imprimible Construir Mobile Construir SMS Aplicar formato básico
Indexado		BD índices	Indexar Buscar candidatos Buscar candidatos título Buscar candidatos desarrollo Buscar texto enfatizado
Calendarizado		Calendario	Agregar al calendario
Taggeado		Tags	Taggear
Publicado			Hacer visible al usuario

Luego, se muestra una tabla para cada uno de los grupos, en la cual se detallan los símbolos de ese grupo, separados por categorías y para cada uno de ellos se detallan las referencias que poseen desde otros grupos.

Tabla 8-12. Detalle de símbolos y sus referencias para el grupo de redactado

	Redactado	Rechazado	Aceptado	HTML construido	Versiones auxiliares	Indexado	Calendarizado	Taggeado	Publicado
Estados									
Redactado									
Sujetos									
Redactor									
Editor		2	1			1	2	1	
Gestor		1	1				1	1	
Objetos									
Contenido		1	5	3	3		1	5	3
Titulo				5		1			5
Desarrollo						1			

Verbos									
Crear contenido									
Aceptar contenido									
Rechazar contenido									

Tabla 8-13. Detalle de símbolos y sus referencias para el grupo de Rechazado

	Redactado	Rechazado	Aceptado	HTML construido	Versiones auxiliares	Indexado	Calendarizado	Taggeado	Publicado
Estados									
Rechazado	1								
Sujetos									
Ninguno									
Objetos									
Ninguno									
Verbos									
Limpiar historia									

Tabla 8-14. Detalle de símbolos y sus referencias para el grupo de Aceptado

	Redactado	Rechazado	Aceptado	HTML construido	Versiones auxiliares	Indexado	Calendarizado	Taggeado	Publicado
Estados									
Aceptado	1								
Sujetos									
Usuario final				1					
Publicador				32	21	10			
Objetos									
Ninguno									
Verbos									
Publicar contenido	1		1						
Construir versiones publicables del contenido									

Tabla 8-15. Detalle de símbolos y sus referencias para el grupo de HTML construido

	Redactado	Rechazado	Aceptado	HTML construido	Versiones auxiliares	Indexado	Calendarizado	Taggeado	Publicado
Estados									
HTML construido			1						
Sujetos									
Ninguno									
Objetos									
HTML					1				
Plantilla									
Black word list									
White word list									
Verbos									
Crear HTML			1						
Definir nombre					4				
Formatear contenido									
Quitar símbolos					2				
Quitar artículos									
Quitar black						2			
Priorizar White						1			
Recortar desarrollo					2				

Tabla 8-16. Detalle de símbolos y sus referencias para el grupo de Versiones auxiliares

	Redactado	Rechazado	Aceptado	HTML construido	Versiones auxiliares	Indexado	Calendarizado	Taggeado	Publicado
Estados									
Versiones auxiliares construidas									
Sujetos									
Ninguno									
Objetos									
PDF									
Imprimible									
Mobile									
SMS									
Formato básico									

Formato auxiliar									
Verbos									
Construir auxiliar			1						
Construir PDF									
Construir imprimible									
Construir Mobile									
Construir SMS									
Aplicar formato básico									

Tabla 8-17. Detalle de símbolos y sus referencias para el grupo de Indexado

	Redactado	Rechazado	Aceptado	HTML construido	Versiones auxiliares	Indexado	Calendarizado	Taggeado	Publicado
Estados									
Indexado					1		1	1	
Sujetos									
Ninguno									
Objetos									
BD índices									
Verbos									
Indexar			1						
Buscar candidatos									
Buscar candidatos título									
Buscar candidatos desarrollo									
Buscar texto enfatizado									

Tabla 8-18. Detalle de símbolos y sus referencias para el grupo de Calendarizado

	Redactado	Rechazado	Aceptado	HTML construido	Versiones auxiliares	Indexado	Calendarizado	Taggeado	Publicado
Estados									
Calendarizado					1	1		1	
Sujetos									

Ninguno									
Objetos									
Calendario									
Verbos									
Agregar al calendario			1						

Tabla 8-19. Detalle de símbolos y sus referencias para el grupo de Taggeado

	Redactado	Rechazado	Aceptado	HTML construido	Versiones auxiliares	Indexado	Calendarizado	Taggeado	Publicado
Estados									
Taggeado					1	1	1		
Sujetos									
Ninguno									
Objetos									
Tags									
Verbos									
Taggear			1						

Tabla 8-20. Detalle de símbolos y sus referencias para el grupo de Publicado

	Redactado	Rechazado	Aceptado	HTML construido	Versiones auxiliares	Indexado	Calendarizado	Taggeado	Publicado
Estados									
Publicado					1	1	1	1	
Sujetos									
Ninguno									
Objetos									
Ninguno									
Verbos									
Hacer visible al usuarios			1						

8.3 Material del experimento

En esta sección se describen 3 elementos relativos al experimento. En primer lugar se brinda una descripción de la aplicación. Luego se presentan diagramas que ilustran las características transversales identificadas utilizando la estrategia Theme/Doc. Por último, se presentan los símbolos ordenados por grupos, los cuales permiten identificar las características transversales utilizando la estrategia del LEL.

8.3.1 Descripción de la aplicación

La descripción de la aplicación es la siguiente. Sea una aplicación para administrar los pedidos que se realizan a un área de informática. El área está dirigida por un jefe de área, que tiene a su cargo a 3 secciones. Cada sección tiene un jefe de sección quien está a cargo de un conjunto disjuncto de especialistas.

Cada rol (jefe de área, jefe de sección y especialista) tiene distinta visibilidad del conjunto total de pedidos. El jefe de área puede ver todos los pedidos (es decir, los pedidos sin asignar, los asignados a él y los asignados a cualquier otro). El jefe de sección puede ver todos los pedidos de su sección (es decir, los pedidos asignados a él y a cualquier especialista de su sección). Finalmente, los especialistas solo pueden ver los pedidos asignados a él.

Los pedidos son creados por cualquier persona (jefes de área, jefe de sección o especialista). Luego de que un pedido es creado, se lo debe asignar de acuerdo a la visibilidad indicada previamente. Finalmente, los especialistas son los únicos autorizados para comenzar a trabajar con los pedidos y terminarlos.

Es posible realizar estadísticas sobre los pedidos terminados. De acuerdo a la visibilidad que le da el rol, las estadísticas son a distintos niveles: área, sección o persona. Apoyado en estas estadísticas, el jefe de área (y solo él) es el único que está autorizado a mover especialistas de una sección a otra.

Es necesario dejar registro de auditoria de ciertas operaciones sensibles. Básicamente, las operaciones que requieren de autorización. Por lo cual, el sistema loggnea las siguientes acciones: comenzar a trabajar en una tarea, terminar una tarea y redistribuir especialistas.

Luego, es necesario que el sistema informe cuando alguna persona realiza una operación y esa operación tiene impacto sobre otra persona. Es por ello que el sistema notifica cuando se asigna una tarea a una persona y también informa cuando un especialista es trasladado (redistribuido) de una sección a otra.

8.3.2 Vistas de acciones

A continuación se presentan cada uno de las vistas de acciones (*action view*) que corresponden a los requerimientos definidos por cada uno de los 10 sujetos. En las vistas de acciones pueden verse las themes transversales (*crosscutting themes*) haciendo referencia a los themes bases (*base themes*). Por lo cual, gráficamente queda en evidencia los themes transversales (*crosscutting themes*) identificados por cada uno de los sujetos del experimento.

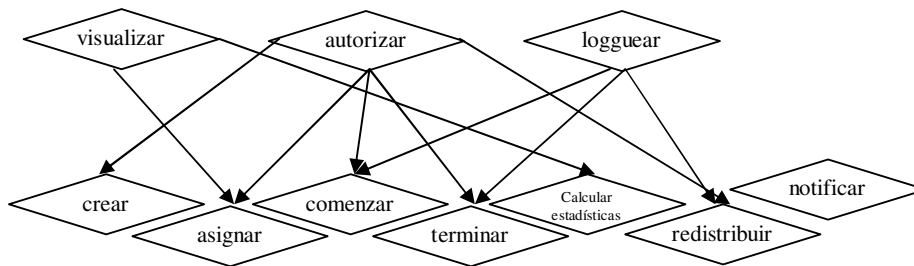


Figura 8-1. Vista de acción (*action view*) del sujeto #1

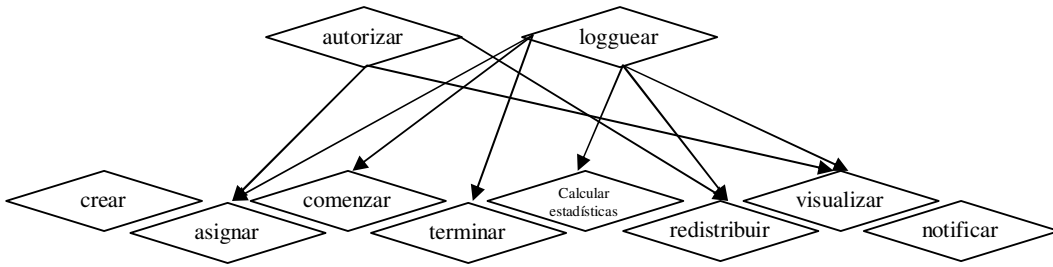


Figura 8-2. Vista de acción (action view) del sujeto #2

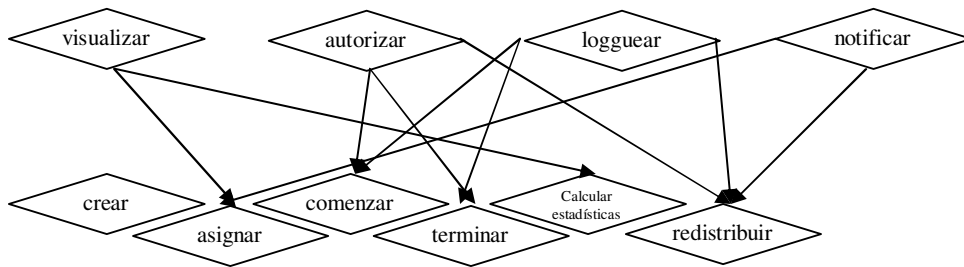


Figura 8-3. Vista de acción (action view) del sujeto #3

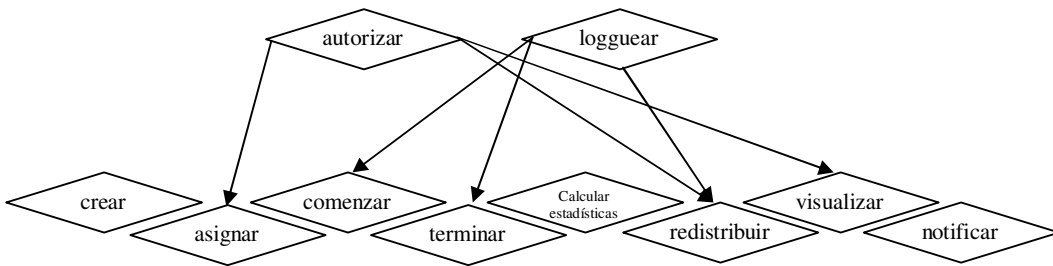


Figura 8-4. Vista de acción (action view) del sujeto #4

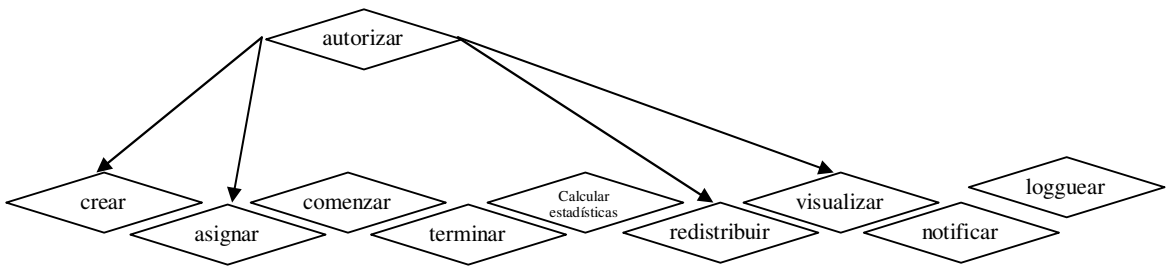


Figura 8-5. Vista de acción (action view) del sujeto #5

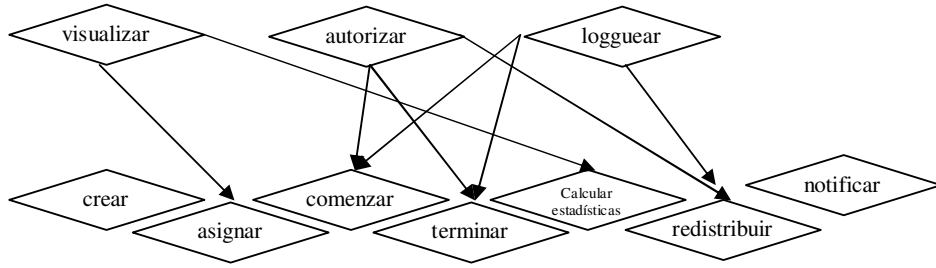


Figura 8-6. Vista de acción (action view) del sujeto #6

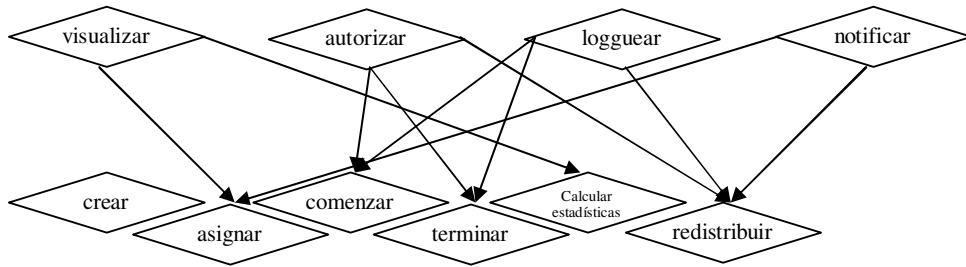


Figura 8-7. Vista de acción (action view) del sujeto #7

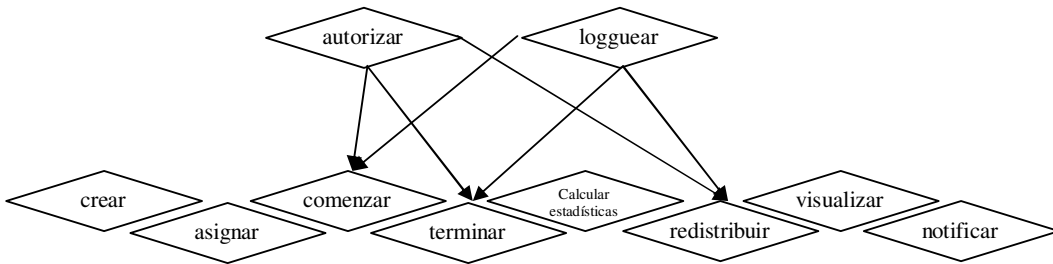


Figura 8-8. Vista de acción (action view) del sujeto #8

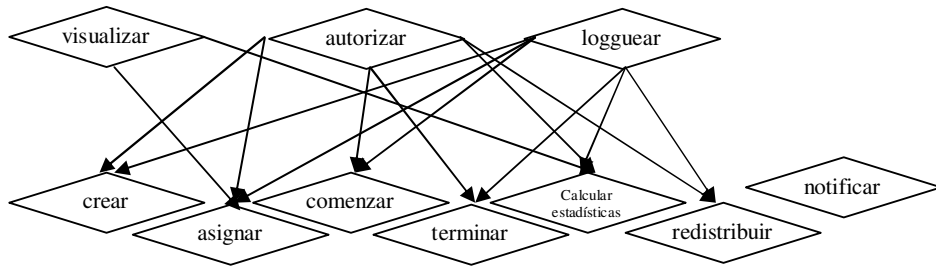


Figura 8-9. Vista de acción (action view) del sujeto #9

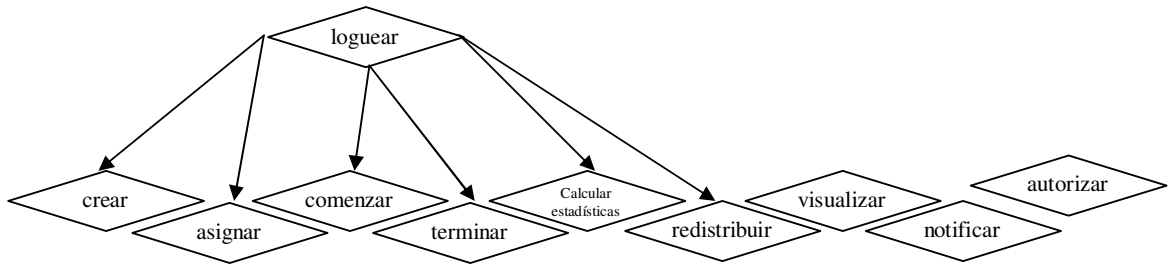


Figura 8-10. Vista de acción (action view) del sujeto #10

8.3.3 Símbolos de LEL agrupados por estados

A continuación se presentan cada uno de los símbolos identificados por cada uno de los sujetos del experimento. Se presenta una tabla por cada uno de los sujetos, en donde, en dicha tabla además de los símbolos se muestran los grupos a los cuales pertenecen, como así también se detallan las referencias que proceden desde otro grupo. Las características transversales son los símbolos que pertenecen a los grupos con mayor cantidad de referencias en donde las mismas están distribuidas desde los otros grupos. Los mismos se resaltan en negrita para hacerlos claros.

Tabla 8-21. Detalle de símbolos y sus referencias del sujeto #1

	Nuevo	Asignado	Terminado
Nuevo			
Ticket		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas
Jefe de área		Desde ver	Desde calcular estadísticas Desde redistribuir
Jefe de Sección		Desde ver	Desde calcular estadísticas
Especialista		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas Desde redistribuir
Crear			
Asignar			
Asignado	Desde Nuevo Desde Ticket Desde los 3 roles		
Ver	Desde ticket Desde los 3 roles Desde asignar		Desde calcular estadísticas
Comenzar	Desde ticket Desde especialista		
Terminar	Desde ticket Desde especialista		
Loguear	Desde asignar		Desde redistribuir
Notificar	Desde asignar		Desde redistribuir

Terminado		Desde asignado	
Calcular estadísticas	Desde ticket Desde 3 roles		
Redistribuir	Desde jefe de área		

Tabla 8-22. Detalle de símbolos y sus referencias del sujeto #2

	Nuevo	Asignado	Terminado
Nuevo			
Ticket		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas
Jefe de área		Desde ver	Desde calcular estadísticas Desde redistribuir
Jefe de Sección		Desde ver	Desde calcular estadísticas
Especialista		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas Desde redistribuir
Crear			
Asignar			
Asignado	Desde Nuevo Desde Ticket Desde los 3 roles		
Ver	Desde ticket Desde los 3 roles Desde asignar		Desde calcular estadísticas
Comenzar	Desde ticket Desde especialista		
Terminar	Desde ticket Desde especialista		
Autorizar	Desde asignar		Desde redistribuir
Loguear	Desde asignar		Desde redistribuir
Notificar	Desde asignar		Desde redistribuir
Terminado		Desde asignado	
Calcular estadísticas	Desde ticket Desde 3 roles		
Redistribuir	Desde jefe de área		

Tabla 8-23. Detalle de símbolos y sus referencias del sujeto #3

	Nuevo	Asignado	Terminado
Nuevo			
Ticket		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas

Jefe de área		Desde ver	Desde calcular estadísticas Desde redistribuir
Jefe de Sección		Desde ver	Desde calcular estadísticas
Especialista		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas Desde redistribuir
Crear			
Asignar			
Autorizar			Desde redistribuir
Asignado	Desde Nuevo Desde Ticket Desde los 3 roles		
Ver	Desde ticket Desde los 3 roles Desde asignar		Desde calcular estadísticas
Comenzar	Desde ticket Desde especialista		
Terminar	Desde ticket Desde especialista		
Loguear	Desde asignar		Desde redistribuir
Notificar	Desde asignar		Desde redistribuir
Terminado		Desde asignado	
Calcular estadísticas	Desde ticket Desde 3 roles		
Redistribuir	Desde jefe de área		

Tabla 8-24. Detalle de símbolos y sus referencias del sujeto #4

	Nuevo	Asignado	Terminado
Nuevo			
Ticket		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas
Jefe de área		Desde ver	Desde calcular estadísticas Desde redistribuir
Jefe de Sección		Desde ver	Desde calcular estadísticas
Especialista		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas Desde redistribuir
Crear			
Asignar			
Asignado	Desde Nuevo Desde Ticket Desde los 3 roles		
Ver	Desde ticket Desde los 3 roles Desde asignar		Desde calcular estadísticas
Comenzar	Desde ticket		

	Desde especialista		
Terminar	Desde ticket Desde especialista		
Autorizar	Desde asignar		Desde redistribuir
Loguear	Desde asignar		Desde redistribuir
Notificar	Desde asignar		Desde redistribuir
Terminado		Desde asignado	
Calcular estadísticas	Desde ticket Desde 3 roles		
Redistribuir	Desde jefe de área		

Tabla 8-25. Detalle de símbolos y sus referencias del sujeto #5

	Nuevo	Asignado	Terminado
Nuevo			
Ticket		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas
Jefe de área		Desde ver	Desde calcular estadísticas Desde redistribuir
Jefe de Sección		Desde ver	Desde calcular estadísticas
Especialista		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas Desde redistribuir
Crear			
Asignar			
Autorizar			Desde redistribuir
Notificar			Desde redistribuir
Asignado	Desde Nuevo Desde Ticket Desde los 3 roles		
Ver	Desde ticket Desde los 3 roles Desde asignar		Desde calcular estadísticas
Comenzar	Desde ticket Desde especialista		
Terminar	Desde ticket Desde especialista		
Loguear	Desde asignar		Desde redistribuir
Terminado		Desde asignado	
Calcular estadísticas	Desde ticket Desde 3 roles		
Redistribuir	Desde jefe de área		

Tabla 8-26. Detalle de símbolos y sus referencias del sujeto #6

	Nuevo	Asignado	Terminado
Nuevo			
Ticket		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas
Jefe de área		Desde ver	Desde calcular estadísticas Desde redistribuir
Jefe de Sección		Desde ver	Desde calcular estadísticas
Especialista		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas Desde redistribuir
Crear			
Asignar			
Autorizar			Desde redistribuir
Notificar			Desde redistribuir
Asignado	Desde Nuevo Desde Ticket Desde los 3 roles		
Ver	Desde ticket Desde los 3 roles Desde asignar		Desde calcular estadísticas
Comenzar	Desde ticket Desde especialista		
Terminar	Desde ticket Desde especialista		
Loguear	Desde asignar		Desde redistribuir
Terminado		Desde asignado	
Calcular estadísticas	Desde ticket Desde 3 roles		
Redistribuir	Desde jefe de área		

Tabla 8-27. Detalle de símbolos y sus referencias del sujeto #7

	Nuevo	Asignado	Terminado
Nuevo			
Ticket		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas
Jefe de área		Desde ver	Desde calcular estadísticas Desde redistribuir
Jefe de Sección		Desde ver	Desde calcular estadísticas
Especialista		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas Desde redistribuir

Crear			
Asignar			
Asignado	Desde Nuevo Desde Ticket Desde los 3 roles		
Ver	Desde ticket Desde los 3 roles Desde asignar		Desde calcular estadísticas
Comenzar	Desde ticket Desde especialista		
Terminar	Desde ticket Desde especialista		
Autorizar	Desde asignar		Desde redistribuir
Loguear	Desde asignar		Desde redistribuir
Notificar	Desde asignar		Desde redistribuir
Terminado		Desde asignado	
Calcular estadísticas	Desde ticket Desde 3 roles		
Redistribuir	Desde jefe de área		

Tabla 8-28. Detalle de símbolos y sus referencias del sujeto #8

	Nuevo	Asignado	Terminado
Nuevo			
Ticket		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas
Jefe de área		Desde ver	Desde calcular estadísticas Desde redistribuir
Jefe de Sección		Desde ver	Desde calcular estadísticas
Especialista		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas Desde redistribuir
Crear			
Asignar			
Autorizar			Desde redistribuir
Notificar			Desde redistribuir
Asignado	Desde Nuevo Desde Ticket Desde los 3 roles		
Ver	Desde ticket Desde los 3 roles Desde asignar		Desde calcular estadísticas
Comenzar	Desde ticket Desde especialista		
Terminar	Desde ticket Desde especialista		
Loguear	Desde asignar		Desde redistribuir
Terminado		Desde asignado	
Calcular estadísticas	Desde ticket		

	Desde 3 roles		
Redistribuir	Desde jefe de área		

Tabla 8-29. Detalle de símbolos y sus referencias del sujeto #9

	Nuevo	Asignado	Terminado
Nuevo			
Ticket		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas
Jefe de área		Desde ver	Desde calcular estadísticas Desde redistribuir
Jefe de Sección		Desde ver	Desde calcular estadísticas
Especialista		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas Desde redistribuir
Crear			
Asignar			
Notificar			Desde redistribuir
Asignado	Desde Nuevo Desde Ticket Desde los 3 roles		
Ver	Desde ticket Desde los 3 roles Desde asignar		Desde calcular estadísticas
Comenzar	Desde ticket Desde especialista		
Terminar	Desde ticket Desde especialista		
Autorizar	Desde asignar		Desde redistribuir
Loguear	Desde asignar		Desde redistribuir
Terminado		Desde asignado	
Calcular estadísticas	Desde ticket Desde 3 roles		
Redistribuir	Desde jefe de área		

Tabla 8-30. Detalle de símbolos y sus referencias del sujeto #10

	Nuevo	Asignado	Terminado
Nuevo			
Ticket		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas
Jefe de área		Desde ver	Desde calcular estadísticas Desde redistribuir

Jefe de Sección		Desde ver	Desde calcular estadísticas
Especialista		Desde ver Desde comenzar Desde terminar	Desde calcular estadísticas Desde redistribuir
Crear			
Asignar			
Asignado	Desde Nuevo Desde Ticket Desde los 3 roles		
Ver	Desde ticket Desde los 3 roles Desde asignar		Desde calcular estadísticas
Comenzar	Desde ticket Desde especialista		
Terminar	Desde ticket Desde especialista		
Autorizar	Desde asignar		Desde redistribuir
Loguear	Desde asignar		Desde redistribuir
Notificar	Desde asignar		Desde redistribuir
Terminado		Desde asignado	
Calcular estadísticas	Desde ticket Desde 3 roles		
Redistribuir	Desde jefe de área		