

# Traceability en la Etapa de Elicitación de Requerimientos

Leandro Antonelli, Alejandro Oliveros

LIFIA, Facultad de Informática, UNLP, Dep. de Computación, Facultad de Ingeniería, UBA  
[lanto@sol.info.unlp.edu.ar](mailto:lanto@sol.info.unlp.edu.ar), [oliveros@fibertel.com.ar](mailto:oliveros@fibertel.com.ar)

**Resumen.** Este artículo presenta un conjunto de reglas que permiten asegurar que se lleve a cabo traceability en el modelo de elicitación de requerimientos “Client-oriented baseline approach” [1]. Estas reglas indican la forma de mantener la consistencia de la información a lo largo de los cambios. A partir de la modificación de un elemento del “client-oriented baseline” determinan los cambios que deben sufrir el resto de los elementos, para que la información sea consistente. Estas heurísticas están implementadas en una aplicación llamada Baseline Mentor Workbench [2], la cual implementa las herramientas del “client-oriented baseline”.

**Palabras Clave:** requerimientos, LEL, escenarios, tarjetas CRC, traceability,

## 1. Introducción

La necesidad de traceability<sup>1</sup> (*traceabilidad, trazabilidad, rastreabilidad*) de los distintos productos que se generan en el proceso de desarrollo de software es ampliamente reconocida en el ámbito de la Ingeniería Software en general y de la Ingeniería de Requerimientos en particular. En este caso nos concentramos sobre un aspecto de los requerimientos: esto es el que abarca desde el proceso de elicitación hasta la generación de las CRC (classes-responsibilities-collaborations) siguiendo el particular enfoque del “client oriented requirements baseline”. En particular se establece la posibilidad de realizar traceability con dicho enfoque.

En la sección 2 se analiza el concepto de traceabilidad dentro de la literatura actual de Ingeniería de Requerimientos. En la sección 3 se presenta la importancia de traceability. En la sección 4 se muestran las herramientas del “client oriented requirements baseline”: LEL, escenarios y tarjetas CRC. En la sección 5 se muestra una alternativa para atacar el problema de traceability en “client oriented requirements baseline”, a través de reglas cuyo objetivo es mantener la consistencia de la información. En la

---

<sup>1</sup> La traducción de la palabra “traceability” es una fuente de problemas. Habitualmente se la traduce como “trazabilidad”, esta opción, muy común en el campo de la Ingeniería de Calidad, también genera ciertos problemas. “Trazabilidad” es la habilidad de “trazar”, y “trazar” es la acción de dejar trazas, no de encontrarlas. Siguiendo estrictamente los criterios de la Academia de la Lengua, debería utilizarse “rastreabilidad”. En efecto: “Rastreabilidad” es la habilidad de “rastrear”, y “rastrear” es seguir el rastro. Sin embargo “rastreabilidad” no es de uso habitual. Por eso los autores han optado por utilizar la palabra original inglesa y hacer esta aclaración.

sección 6 se compara nuestra propuesta con otras alternativas. Y por último en la sección 7 se muestra BMW, una herramienta que implementa los elementos tratados.

## 2. El concepto de traceability

Para [3] traceability es un atributo de la *especificación de requerimientos de software* (SRS):

*"A software requirements specification is traceable if (i) the origin of each of its requirements is clear and if (ii) it facilitates the referencing of each requirement in future development or enhancement documentation" [3]*

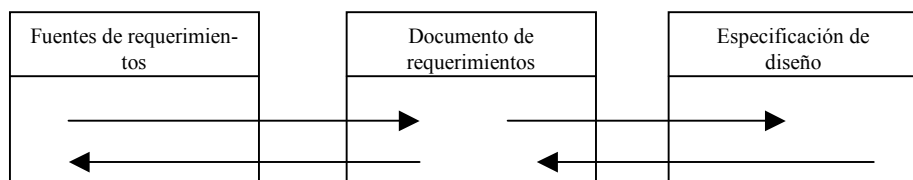
La definición de traceability de este estándar, que se reconoce como la más difundida [4], claramente se concentra en la SRS y requiere propiedades de ella. En primer lugar, que permita remontarse hacia atrás en el origen de los requerimientos y que también posibiliten referenciar a cada requerimiento en la documentación de desarrollo.

Por su parte Wieringa [5] propone una generalización del concepto extendiéndolo más allá de la especificación de requerimientos de software hasta abarcar todos los documentos que se generan a lo largo del ciclo de vida:

*"The documents created and maintained during the initial development and throughout the life cycle of the product are called traceable if each part of each document can be traced to any other part of any relevant document in such a way that parts that must change together, are linked to each other. The linked documents may describe user needs, system requirements, system design, system implementation, system context, etc."*

La generalización de Wieringa es doble: Por un lado traceability es un atributo que, además de la SRS, poseen otros documentos. Y esos documentos abarcan todo el *ciclo de vida de desarrollo de software* (SDLC).

Para Davis [6] traceability son links de cuatro tipos posibles. Con centro en los requerimientos se dirigen desde y hacia las fuentes de los requerimientos y especificaciones de diseño. Gráficamente el enfoque de Davis se puede representar [7]:



Se ha señalado que el aspecto más importante de traceability de la información [7] es el que registra las relaciones entre los mismos requerimientos. En la misma línea se encuentra [8]:

*"Traceability information is information which allows you to find dependencies between requirements, and between the requirements and the system design components and documentation."*

Indudablemente resulta de importancia la relación entre los mismos requerimientos, pero cuando se avanza en el desarrollo, resulta de mayor relevancia para el ingeniero de software la posibilidad de establecer los vínculos de los distintos elementos con los requerimientos del sistema.

La conceptualización más rica, y que en cierta medida es la que inspira los conceptos de Wieringa [5] es la de Gotel [4]. Traceability es un atributo del proceso que, amplía la definición de IEEE para considerar los vínculos de los requerimientos con sus antecedentes y su evolución como tal:

*"Requirements traceability refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases)." [4]*

Gotel [9] introduce una mayor precisión en las ideas de traceability hacia adelante o hacia atrás (*forward and backward traceability*) con los conceptos de traceability de requerimientos (RT) horizontal y vertical. *RT horizontal* es traceability a través de versiones de los requerimientos, en tanto que *RT vertical* es traceability entre las fases previas y posteriores del SDLC.

Traceability hacia adelante o hacia atrás se refiere a la dirección en la que se realiza la RT. En el caso de la RT vertical, traceability hacia atrás es desde los requerimientos hasta sus orígenes y hacia adelante es desde los requerimientos hasta su realización. En la RT horizontal consiste en, a partir de una versión, avanzar hacia versiones posteriores o retroceder hacia versiones anteriores.

Gotel [9] también hace la siguiente distinción en RT teniendo en cuenta la especificación de requerimientos (Requirement Specification, RS):

- Pre-RS, concierne a aquellos aspectos de la vida de los requerimientos anteriores a la inclusión en la RS.
- Post-RS, concierne a aquellos aspectos de la evolución de los requerimientos que resultan de su inclusión en la RS.

Pinheiro por su parte [10] considera traceability a partir de los "rastros" (traces) que son generadas en el proceso de desarrollo de software y que se manifiestan por "diversas" marcas en los productos de software. En ese contexto traceability es:

“...ability to define, capture and follow the traces left by requirements on other elements of software development environment and the traces left by those elements on requirements”

### 3. Importancia y necesidad de traceability

Katonya [7] pone los requerimientos en el contexto mayor de la gestión de cambios. Para Katonya un aspecto crítico del proceso de change management es la evaluación del impacto del cambio en el resto del sistema. El cambio puede producirse en diferentes momentos:

- Cuando se están desarrollando los requerimientos, por lo cual se puede evaluar el efecto del cambio sobre los otros requerimientos
- Si se está en la implementación, es necesario evaluar el impacto en los requerimientos, diseño e implementación.
- Si el sistema está en operación, se requiere una evaluación adicional a la anterior: cómo son afectados los stakeholders.

La evaluación de este impacto requiere información sobre:

- Dependencias de requerimientos
- Fundamentos de los requerimientos
- Implementación de los requerimientos

Esto se denomina "traceability information": "Change impact assessment depend on this traceability information to find out which requirements are affected by a proposed change".

Esto es, la importancia reside en disponer de los elementos para ejecutar adecuadamente el change management y en especial, evaluar el impacto de los cambios en las diferentes etapas del desarrollo de software.

Según Davis [6], hay dos fuentes de razones para traceability:

- para diseñar o testear cualquier componente del sistema es necesario saber qué requerimientos satisface (aunque sea parcialmente)
- para testear el sistema software es necesario saber qué requerimientos están siendo validados en cada test

Por su parte, Sommerville [8] en "Define Traceability Policies" y "Maintain a Traceability Manual" que son dos guías básicas de la ingeniería de requerimientos, plantea los beneficios de contar con políticas de traceability:

- Se dispone de información en la evaluación de los cambios de requerimientos
- Son la base para el control de costos y calidad

Wiegiers [11], con un enfoque muy práctico de los requerimientos, encuentra los siguientes beneficios en traceability de los requerimientos:

- Puede ayudar a demostrar que se implementaron todos los requerimientos (*certificación*)
- Analizar el impacto de los *cambios*
- Ayudar al *mantenimiento*
- Seguimiento de la funcionalidad que se implementa
- *Reingeniería* de los legacy systems
- Facilita el reuso de componentes identificando paquetes de requerimientos, diseños, código, test, y otros artefactos
- Reducción del *riesgo*. Conocer los links entre las componentes ayuda a superar el problema de gente clave que abandona un proyecto
- *Testeo*

El Software Engineering Body of Knowledge, entre las actividades de la Ingeniería de Producto Software menciona al mantenimiento de traceability y consistencia entre los documentos producidos en el desarrollo [12]

De lo anterior se concluye que traceability es un elemento clave en la RE y que crecientemente se la reconoce como un punto importante por estándares y guías para prácticas de ingeniería de requerimientos. A pesar de muchos avances, RT sigue siendo un área de problema ampliamente reportada [4]. Muchos de los problemas atribuidos a RT se deben a una falta de traceability pre-RS [13] [14]. Por lo tanto, se requieren técnicas para registrar y rastrear esta información [4].

#### 4. LEL y escenarios

El desarrollo de un sistema de software requiere el entendimiento del macrosistema en el cual éste será utilizado. En el trabajo de Leite “client-oriented baseline” [1], se presenta una metodología para adquirir información y modelar el macrosistema usando el lenguaje del cliente [15].

El “client-oriented baseline” está compuesto por el Language Extended Lexicon (LEL) y por escenarios. Estos dos elementos se relacionan entre sí formando un hipertexto (hypertext view). Además, el LEL y los escenarios, están organizados en versiones, cuya organización se puede apreciar a través de la llamada vista de configuración.

Para utilizar junto al “client-oriented baseline” se han desarrollado heurísticas [15] para derivar tarjetas CRC [16] a partir de las entradas del LEL y los escenarios. Estas tarjetas identifican entidades relevantes en el dominio de la aplicación. Estas entidades son candidatas a convertirse en clases en un modelo preliminar de objetos de una aplicación que satisfaga los requerimientos del dominio elicitado.

El client-oriented baseline, junto a las tarjetas CRC y a las heurísticas de derivación, conforman las herramientas necesarias para comenzar la construcción de software en la etapa de requerimientos.

## 5. Traceability pre-RS y Client-oriented baseline

Este artículo presenta un conjunto de reglas para mantener la consistencia de la información que conforma el “client-oriented baseline”, posibilitando así hacer traceability entre estos elementos.

“Client-oriented baseline” presenta herramientas para utilizar en el marco de pre-RS, ya que la información obtenida en el LEL y los escenarios, son la materia prima, para la elaboración de los requerimientos, y por ende del documento de especificación de requerimientos.

Con respecto al sentido (hacia el origen o hacia la implementación) planteado en la sección 2, las reglas presentadas, se corresponden con el segundo tipo, hacia la implementación. Por otra parte el LEL provee el vocabulario para la definición del mismo LEL y los escenarios. Luego, a partir del LEL y el conjunto de escenarios, se obtienen tarjetas CRC. Por este motivo primero se construye el LEL, luego los escenarios y por último las tarjetas CRC. Las reglas indican los pasos que se deben cumplir para garantizar traceability en este sentido: LEL, escenarios, tarjetas CRC.

Si bien el orden de construcción es: LEL, escenarios y tarjetas CRC, es impracticable la tarea de terminar completamente el LEL, antes de comenzar con los escenarios. Y de igual manera para las tarjetas CRC. Por lo tanto cuando se consigue un nivel aceptable del LEL, se comienza con los escenarios. Y una vez realizados y corregidos los escenarios, se aplica el algoritmo de derivación [15] para obtener las tarjetas CRC. El problema que se presenta es que ante alguna modificación de un elemento del LEL o de un escenario, se debe revisar el resto de los elementos del modelo para determinar que otros cambios se deben realizar para que la información mantenga la consistencia. Las tareas a realizar son variadas. Por ejemplo, dado el caso en donde se modifica un símbolo del LEL, no las nociones ni los impactos sino la palabra en sí, es muy probable que esta palabra esté utilizada para definir otros símbolos (principio de circularidad), entonces es deseable que las otras palabras también se modifiquen. Otro caso hipotético es que se elimine un símbolo del LEL, es decir que se descarte la entrada, esto podría ocasionar que se elimine alguna tarjeta CRC derivada a partir de la entrada de LEL eliminada. Esto se debe a que las reglas de derivación de tarjetas CRC, determinan que se busquen actores de escenarios que estén definidos en el LEL, para obtener tarjetas CRC. Si la entrada del LEL desaparece, la tarjeta CRC no debiera existir tampoco, por lo cual se debe eliminar [15].

Las reglas propuestas en este artículo determinan una forma sistemática de propagar los cambios por todos los elementos (LEL, escenarios y tarjetas CRC) que forman el “client-oriented baseline”, para que a pesar de los cambios que sufra éste, se pueda practicar traceability de sus elementos.

La confección de las reglas involucra el problema de asegurar que las mismas cubran todos los posibles cambios que se pueden realizar sobre todos los atributos de todos los elementos que conforman el client-oriented baseline. Para este fin, se condujo un análisis exhaustivo y sistemático.

La *tabla 1*, resume los tres elementos claves: cambios, elementos y atributos. Los elementos que pueden sufrir cambios, son los elementos del client-oriented baseline: entradas del LEL, escenarios y tarjetas CRC. Los atributos de cada uno se describen

en forma textual y conforman un grafo. Los cambios que pueden sufrir cada unidad de información tienen que ver con la modificación del texto y el armado de links.

Sobre la matriz de *tabla 1*, se analizó cada una de las intersecciones, para determinar si la acción sobre el elemento, debiera desencadenar o no algún tipo de corrección sobre el resto de los elementos. Las intersecciones con números son aquellas acciones sobre elementos que necesitan de acciones correctivas para asegurar la consistencia de la información. Distintas acciones sobre el mismo elemento, pueden desencadenar la misma acción correctiva. Los mismos números ponen de manifiesto este hecho, ya que cada uno identifica la necesidad de una acción correctiva distinta.

Elementos	Atributos	Cambios		Agregar		Modificar				Eliminar	
				Agregar		Modificar		Eliminar		Eliminar	
		Texto	Link	Texto	Link	Texto	Link	Texto	Link	Texto	Link
Entrada del LEL	Sinónimos	1		1		2		3		3	
	Noción										
	Impactos	4	4	4	4	4	4	4	4	4	4
Escenario	Título			5	5	5	5	5	5		
	Objetivo										
	Contexto										
	Actores	6	6	6	6			7	7	7	7
	Recursos										
	Episodios		8		8				9		9
Tarjeta CRC	Nombre										
	Responsabilidades		10		10				10		10
	Colaboraciones										

Tabla 1. Acciones, las cuales disparan funciones para mantener traceability

Siguiendo los conceptos de situación, decisión y acción tratados en [17], se presentan las reglas.

#### Regla 1.

**Situación:** Un sinónimo de una entrada del LEL se escribió en forma incorrecta.

**Decisión:** Rescritura de un sinónimo de una entrada del LEL.

**Acción:** Se deben cambiar todas las ocurrencias del sinónimo incorrecto escrito en otras entradas del LEL, escenarios o tarjetas CRC.

#### Regla 2.1

**Situación:** Se han definido en una entrada del LEL, dos sinónimos, cuando en realidad son dos entradas independientes.

**Decisión:** La eliminación de un sinónimo de una entrada del LEL.

**Acción:** Se deben eliminar los links de otras entradas del LEL, escenarios o tarjetas CRC donde se utiliza el sinónimo borrado.

#### Regla 2.2

Situación: Se han definido en una entrada del LEL dos sinónimos, cuando en realidad se debe usar uno solo.

Decisión: La eliminación de un sinónimo de una entrada del LEL

Acción: Se debe reemplazar el texto del sinónimo eliminado por otro sinónimo de la entrada del LEL modificada, en todas las entradas del LEL, escenarios y tarjetas CRC con links a la entrada de LEL modificada.

Regla 3.

Situación: Se ha encontrado que una nueva palabra es sinónimo de una entrada de LEL existente.

Decisión: El agregado del sinónimo a la entrada del LEL.

Acción: Se deben crear nuevos links entre las entradas del LEL, escenarios y tarjetas CRC que utilicen la nueva palabra y la entrada de LEL modificada. Si se crea un link para un escenario, con un actor que concuerda con el nuevo sinónimo, se debe crear la tarjeta CRC correspondiente.

Regla 4.

Situación: Los impactos de una entrada del LEL no son los adecuados.

Decisión: La modificación de los impactos de la entrada del LEL (los cuales están directamente relacionados con las responsabilidades de las tarjetas CRC)

Acción: Las responsabilidades de la tarjeta CRC asociada (aquella obtenida a partir de la entrada del LEL) deben sufrir las mismas modificaciones.

Regla 5.

Situación: El título de un escenario no es el correcto.

Decisión: La modificación del título del escenario.

Acción: Se debe modificar el texto en todos los escenarios y tarjetas CRC que poseen links a este escenario.

Regla 6.1

Situación: En un escenario se ha omitido un actor. Este actor tiene su correspondiente definición en el LEL. Pero esta entrada del LEL no tiene la tarjeta CRC asociada.

Decisión: El agregado del actor en el escenario.

Acción: Se debe obtener la tarjeta CRC asociada a la entrada del LEL correspondiente al actor.

Regla 6.2

Situación: En un escenario se ha omitido un actor. Este actor tiene su correspondiente definición en el LEL. Y esta entrada del LEL tiene la tarjeta CRC asociada.

Decisión: El agregado de un actor a un escenario.

Acción: Se deben actualizar las colaboraciones de la tarjeta CRC con relación al escenario.



#### Regla 7.1

Situación: Se ha incorporado como actor de un escenario, uno que no debe serlo. Este actor incorrecto, no lo es de ningún otro escenario. Ni tampoco es referenciado en las responsabilidades de una tarjeta CRC primaria.

Decisión: La eliminación de un actor de un escenario.

Acción: Se debe eliminar la tarjeta CRC asociada al actor.

#### Regla 7.2

Situación: Se ha incorporado como actor de un escenario, uno que no debe serlo. Este actor incorrecto, no lo es de ningún otro escenario. Pero es referenciado en las responsabilidades de una tarjeta CRC primaria.

Decisión: La eliminación de un actor de un escenario.

Acción: Se debe transformar la tarjeta CRC de primaria a secundaria.

#### Regla 8.

Situación: En los episodios de un escenario se omitió hacer mención a un término que está definido en el LEL. Este término tiene una tarjeta de CRC asociada.

Decisión: El agregado de la referencia a la entrada del LEL en los episodios del escenario.

Acción: Se deben actualizar las colaboraciones para todas las tarjetas CRC cuyas entradas del LEL asociadas se referencian en el escenario.

#### Regla 9.

Situación: En los episodios de un escenario se hace mención por única vez a un término que está definido en el LEL. Esta mención no debe existir. Este término tiene una tarjeta de CRC asociada.

Decisión: La eliminación de la referencia a la entrada del LEL de los episodios del escenario.

Acción: Se deben actualizar las colaboraciones para todas las tarjetas CRC cuyas entradas del LEL asociadas se referencian en el escenario.

#### Regla 10.

Situación: Se debe incorporar por primera vez, una referencia a una entrada del LEL (que no tienen tarjetas CRC asociada) en las responsabilidades de una tarjeta CRC.

Decisión: La incorporación de la referencia a la entrada del LEL.

Acción: Se debe crear la tarjeta CRC secundaria para la entrada del LEL.

#### Regla 11.

Situación: Se debe eliminar de las responsabilidades de una tarjeta CRC, la última referencia a una entrada del LEL. Esta última tiene una tarjeta CRC secundaria asociada.

Decisión: La eliminación de la referencia a la entrada del LEL.

Acción: Se deben eliminar la tarjeta CRC secundaria.

Regla 12.

Situación: Una tarjeta CRC definida, no es lo suficientemente importante para que esté definida.

Decisión: La eliminación de una tarjeta CRC

Acción: Se deben eliminar todas las colaboraciones en donde la tarjeta CRC eliminada sea referenciada.

A partir de las reglas se define la figura 1. En esta figura se muestran los cambios sobre los elementos que desencadenan acciones correctivas a través de un diagrama de actividades [18]. El significado de los símbolos se muestra en figura 2. En el diagrama de figura 1, se muestran 4 columnas. Las tres primeras corresponden a LEL, escenarios y tarjetas CRC. En cada una, se agrupan los cambios que desencadenan acciones correctivas. Cada cambio, a su vez puede ocasionar otras modificaciones, las cuales a su vez necesitan de operaciones correctivas. Esto se refleja a través de las flechas. Por ejemplo eliminar un escenario, trae aparejado el hecho de eliminar un actor, el cual puede originar un problema de consistencia. Ya que al eliminar un actor puede originar la eliminación de una tarjeta CRC, que a su vez puede llevar a modificar las colaboraciones de otras tarjetas CRC. Por lo tanto las flechas, indican las acciones que se disparan, las cuales terminan en alguna actividad de la columna funciones de figura 1.

El diagrama de la figura 1 no se describirá en forma completa debido a su extensión, sin embargo en este artículo se comentan a modo de ejemplo las acciones a realizar disparadas por las actividades que llevan el número 4 y 7.

Para entender estos ejemplos, es necesario recordar las heurísticas de derivación de tarjetas CRC [15]. De acuerdo a estas heurísticas si una entrada de LEL aparece en la lista de actores de un escenario, este hecho justifica la aparición de una tarjeta CRC, con el mismo nombre que la entrada del LEL. Tarjetas de CRC hay de dos tipos: primarias y secundarias. Lo descrito anteriormente es la creación de una tarjeta primaria, ya que juega un rol activo dentro del dominio. Por el contrario las secundarias, contribuyen para complementar la funcionalidad de las primarias, y se las identifica por ser referencias a entradas del LEL presentes en las responsabilidades de la tarjeta CRC. Regresando a la CRC creada, sus responsabilidades se copian de los impactos del LEL, y los impactos del LEL (de acuerdo al principio de circularidad) se deben describir maximizando el uso de las entradas del LEL. Por lo tanto, estas últimas entradas del LEL contenidas en las responsabilidades de la tarjeta CRC se convierten en tarjetas CRC secundarias.

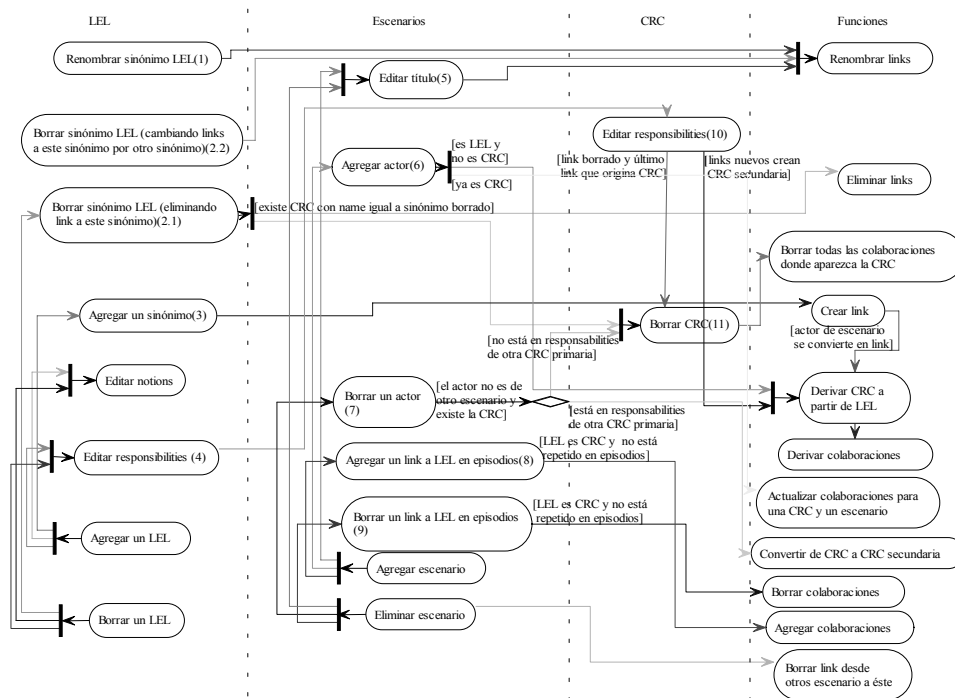


Figura 1. Descripción de traceability

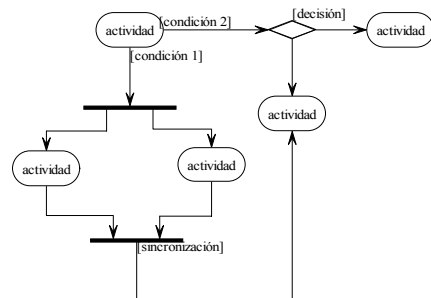


Figura 2. Leyenda del diagrama

Con esta breve explicación, es posible seguir la cadena de actividades desencadenadas por la modificación identificada en el diagrama de la figura 1 como *número 4*. Esta modificación es sobre los impactos de una entrada del LEL. Esta entrada del LEL puede haber originado una tarjeta CRC. Entonces las responsabilidades de la tarjeta CRC tienen que ser modificadas también. Como consecuencia nuevas referencias a entradas del LEL se pueden haber agregado a las responsabilidades. Por lo tanto si la tarjeta CRC era del tipo primaria, las nuevas referencias, deben dar origen a tarjetas CRC secundarias.

Por el contrario, la modificación indicada como *número 7*, se refiere a la eliminación de un actor de un escenario. Dado un actor en un escenario, es posible que exista una tarjeta CRC para este actor, esto es posible si el actor está definido en el LEL. Ahora, si la entrada del LEL era solamente actor en el escenario que se borró (y origina la modificación indicada como *número 7*) y de ningún otro, no hay ninguna justificación para que la tarjeta siga existiendo como primaria. Por lo tanto habría que eliminar la tarjeta CRC como así toda referencia externa desde otras tarjetas CRC.

Sin embargo, si la tarjeta se la referencia desde otra tarjeta primaria en el atributo de responsabilidades, la tarjeta CRC primaria, no se debería eliminar, en cambio se debería transformar a tarjeta secundaria.

## 6. Comparación con otros modelos

El esquema de traceability propuesto resulta de una aplicación de la propuesta de Rolland [17]. Las entradas del LEL, los escenarios y las tarjetas CRC cuadran dentro de la categoría de *evolutionary objects*, propuesta por Rolland. Sobre estos objects propone tres categorías de evoluciones: transformación, expansión y mutación, las cuales están presentes en nuestro modelo. La modificación de una entrada del LEL o un escenario, puede verse como una evolución de transformación. La evolución del tipo de expansión se presenta con la creación y eliminación de las tarjetas CRC, las cuales forman el *spatial environment* del LEL y los escenarios. Con respecto a la evolución de mutación, las tarjetas de CRC pueden cambiar entre ser del tipo primario al tipo secundario y por su parte, las entradas del LEL se pueden mapear en tarjetas de CRC.

Al igual que en [19] nuestro objetivo es hacer trace desde el punto de vista del desarrollo y no del mantenimiento. No se pretende capturar las distintas configuraciones del LEL, escenarios y tarjetas CRC dejando en forma explícita los cambios producidos, con su justificación, el responsable, fecha y hora. Por el contrario, se hace trace entre las distintas etapas del ciclo de vida (requerimientos LEL y Escenarios, diseño preliminar tarjetas CRC), y para que esto sea viable, en este artículo se presentan reglas para mantener consistente la información de estos tres conjuntos LEL, escenarios y CRC. Reglas que son necesarias ya que estos elementos evolucionan. La evolución que se aprecia, es la del tipo intra escenario según la clasificación de [20], sin embargo de las operaciones que cuadran bajo esta categoría, solo observamos las de *input, modification y exclusion*.

En [19] se muestra un modelo de escenarios que evoluciona sobre dos ejes. El primero es el eje de desarrollo, por el cual van sucediendo diferentes configuraciones de escenarios: de requerimientos, de diseño, de codificación. Y el segundo eje, el de mantenimiento, refleja el versionado, con relaciones ortogonales a distintas versiones, para determinar distintas configuraciones. A diferencia, nuestra propuesta sólo se ocupa de la etapa de requerimientos, se preocupa de mantener la homogeneidad entre entradas del LEL, escenarios y tarjetas CRC. Y si bien prevé el manejo de versiones, las relaciones existentes entre los elementos, quedan contenidos dentro de cada versión, sin traspasar sus límites.

En [21] se presentan dos alternativas, modelar traceability de acuerdo al producto y de acuerdo al proceso. En este trabajo, están presentes estas dos visiones. Con respecto a traceability de acuerdo al producto, está presente a través de los links que

pecto a traceability de acuerdo al producto, está presente a través de los links que existen en la definición de los atributos que definen las entradas del LEL, escenarios y tarjetas CRC. Cada atributo se define en forma textual, y en la definición aparecen referencias a otros elementos, esto es lo que en [21] llaman traceability basado en producto. Con respecto a procesos, el único que aparece en esta propuesta es el de derivación de tarjetas CRC a partir del LEL y escenarios, por lo tanto los links de traceability basados en el proceso, están dados por el LEL y escenarios y las tarjetas que generan.

Si bien el modelo presentado no es un método de diseño, de todas formas se puede explicar en función del trabajo de [22] de la siguiente forma. Los *artifacts* de este modelo son las entradas del LEL, los escenarios y las tarjetas CRC. Los *steps*, son las modificaciones que los usuarios realizan sobre las entradas del LEL y los escenarios. El *issue*, es el carácter de la edición realizada, modificación de textos o links de los distintos atributos que conforman los artifacts. *Position* es la regla que se debe aplicar dada la situación del issue. Y por último, los *arguments*, están dados por las reglas de derivación [15].

En [23] se define un modelo a partir de la herramienta *TOOR*. Se definen artefactos y los links entre ellos. Para los artefactos se definen clases (*template*) para luego instanciar casos concretos. Cada clase está definida por atributos. En este caso se tendrían 3 clases LEL, escenarios y CRC. Con respecto a las relaciones, está la de derivación que es el que permite obtener CRC a partir de LEL y escenarios. Luego también está la relación *parte de*, que es la que está presente, cuando se utiliza un símbolo del LEL para definir otro elemento.

## 7. Baseline Mentor Workbench: generalidades de la aplicación

Baseline Mentor Workbench (BMW) [2] [25], es una aplicación que fue construida para asistir al experto del dominio durante la fase de ingeniería de requerimientos utilizando el esquema del “client-oriented baseline”. La importancia de herramientas sobre metodologías específicas se trata en [14] y en [26].

BMW implementa las herramientas del “client-oriented baseline”, las tarjetas CRC con las heurísticas para derivar CRC y los procedimientos sobre traceability. Toda esta información se maneja a través de las características de edición y navegación [27] que provee la aplicación.

La arquitectura de la aplicación es una típica *Arquitectura de Repositorio* [28]. En figura 3 se muestra la arquitectura de la aplicación.

El mantenedor de traceability implementa completamente el diagrama de la figura 1. A través de una clase llamada *Trace* se provee toda la funcionalidad necesaria. Esta clase posee el comportamiento necesario para determinar cuales son los cambios que se producen en los elementos del baseline y realizar las acciones correctivas correspondientes.

Durante la edición en línea se van chequeando algunos elementos, como ser la generación de links, sin embargo, la parte más importante se realiza en el momento de aceptar la edición de una entrada del LEL o escenario. En ese momento se revisa el elemento, para determinar frente a que situación se está. A veces se necesita confirma-

ción por parte del usuario para determinar específicamente la situación. Luego se realizan las modificaciones necesarias ilustradas en la figura 1.

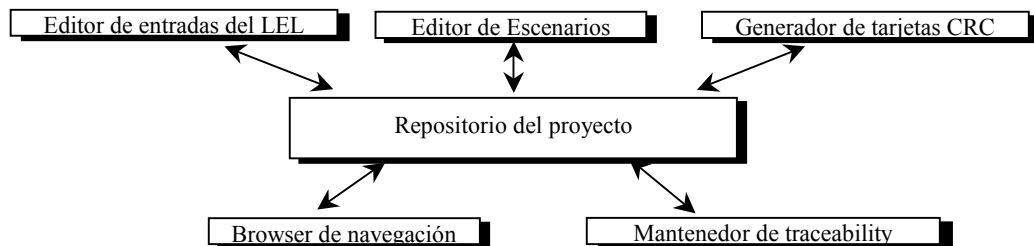


Figura 3. Arquitectura del Baseline Mentor Workbench

A continuación se muestran algunos ejemplos para ilustrar el funcionamiento.

Para el caso de la regla 1, en ella se indica que si se altera un sinónimo de una entrada del LEL, se deben cambiar todas las ocurrencias del sinónimo incorrecto en las otras entradas del LEL, escenarios o tarjetas CRC. Sea como ejemplo la entrada *intimación negativa*, se determina que el nombre no es el adecuado, y que en realidad se lo denomina *desconocido*. La definición de la entrada *intimación negativa* es la de la figura 4.

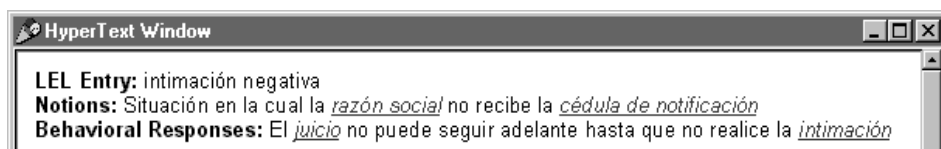


Figura 4. Entrada de LEL intimidación negativa

Esta entrada del LEL, es referenciada desde otros elementos, como por ejemplo desde el escenario *intimación de una razón social* (figura 5). Allí puede verse el link hacia *intimación negativa*, en el último renglón de los episodios.

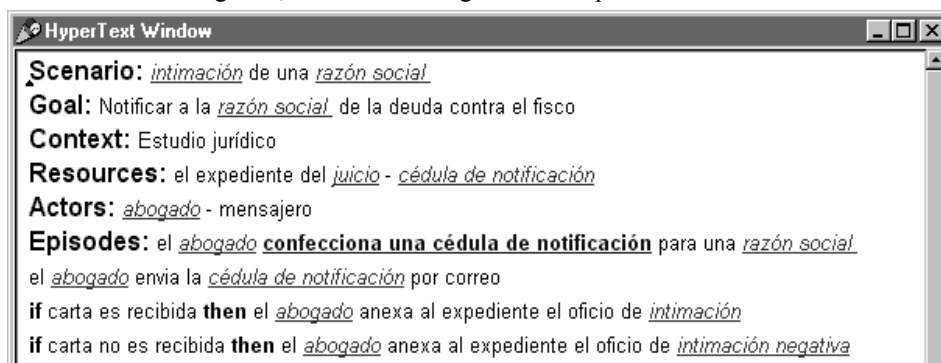


Figura 5. Escenario intimidación de una razón social

Luego de editar el sinónimo intimación negativa y cambiarlo por desconocido, al confirmar la operación, a través de la presión del botón ok, BMW pregunta si se desea actualizar los links (figura 6). El motivo de esta pregunta es que tanto las reglas 1, como 2.1 y 2.2 tienen el mismo desencadenante, la desaparición de algún sinónimo del LEL, entonces BMW no tiene forma de determinar a cual de las reglas pertenece la situación, por lo tanto se le hace la pregunta al usuario.

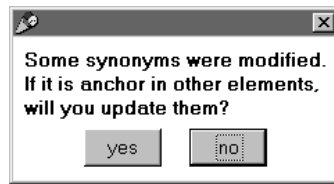


Figura 6. Pedido de confirmación para actualizar los links

Al dar la respuesta yes, se realizan las modificaciones. El escenario intimación de una razón social, queda como el de la figura 7, en donde se reemplazó el texto intimación negativa, por desconocido, pero el link a la entrada del LEL, se mantiene.

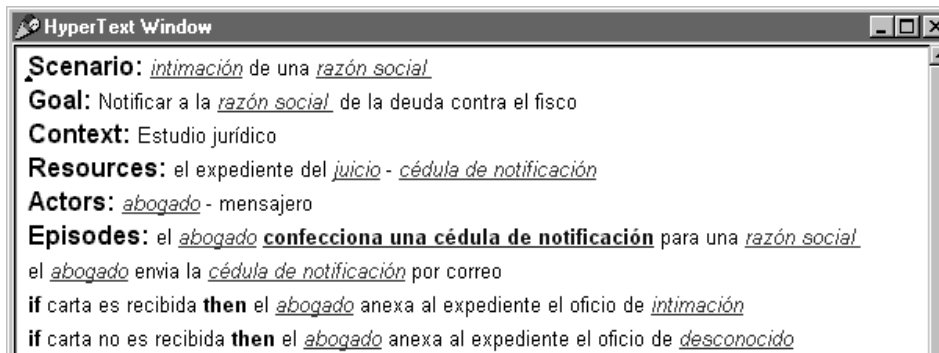


Figura 7. Escenario intimación de una razón social luego de actualizados los links

Sea el caso de la situación de la regla 2.1, en donde hay dos entradas definidas como sinónimos, cuando en realidad son dos entradas distintas, que se ejemplifica con la figura 9.

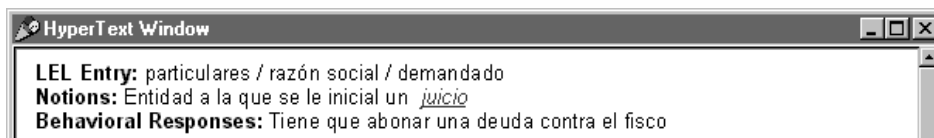
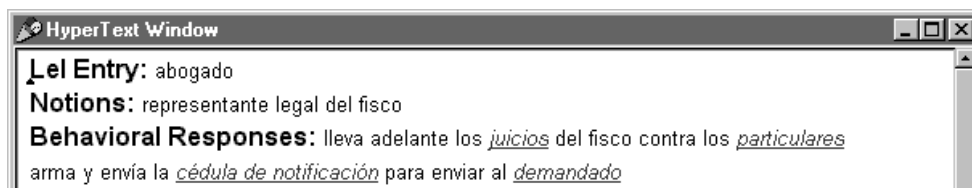


Figura 8. Entrada de LEL razón social

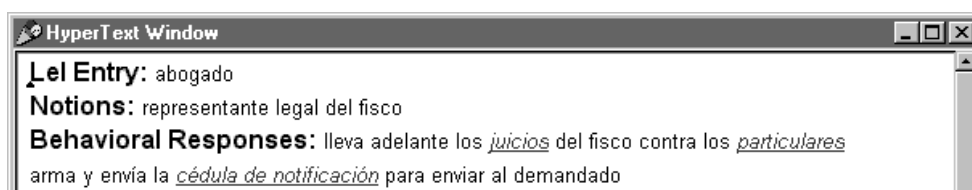
En este ejemplo demandado no es sinónimo de los otros dos, si bien el símbolo existe, no tiene la misma noción que particulares y razón social, por lo tanto se debe eliminar, pero a diferencia del caso como ejemplo de la regla 1 donde se renombra el

ancla y el link permanece, en este caso, el link debe desaparecer, pero el texto del ancla debe permanecer invariable. Como se muestra en la figura 9, la entrada de LEL abogado utiliza en su definición el símbolo demandado aunque la definición del mismo no sea la correcta.



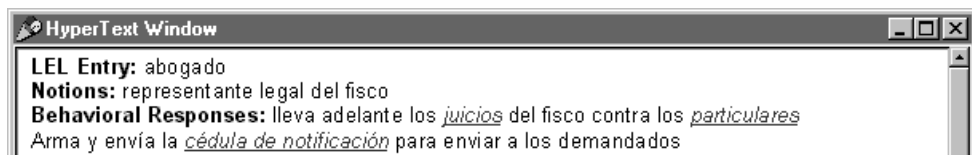
**Figura 9.** Referencia a una entrada de LEL con múltiples sinónimos

Luego de la eliminación del sinónimo demandado, de la entrada de LEL particulares / razón social / demandados, BMW necesita determinar frente a que situación (si la de perteneciente a la regla 1, regla 2.1 o regla 2.2) está presente, por lo que vuelve a mostrar la confirmación de la ventana de la figura 6. Ante la respuesta negativa, la entrada abogado se transforma en la de la figura 10, en donde el texto demandado deja de ser un link.



**Figura 10.** Referencia a una entrada de LEL con múltiples sinónimos

En los 2 ejemplos presentados, se trata la corrección de links y anclas, sin embargo, como resultado de un cambio en los elementos del baseline, se puede llegar a crear una tarjeta CRC nueva. Tómese como ejemplo la entrada de LEL abogado, que se muestra en la figura 11.



**Figura 11.** Entrada de LEL abogado

Esta entrada de LEL tiene una tarjeta CRC asociada con la cual comparte los impactos (figura 12):



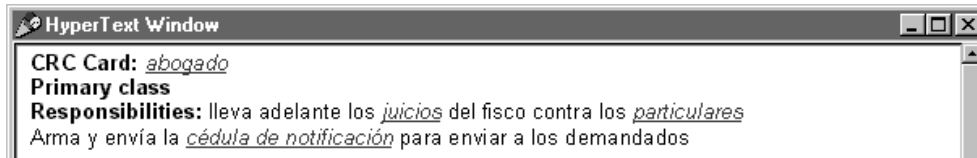


Figura 12. Tarjeta CRC abogado

Sin embargo, los impactos de la entrada de LEL abogado, no son los correctos, ya que están incompletos. Hace falta indicar que el abogado libra los oficios de intimación e intimación negativa. La regla 4, determina que al alterar los impactos de una entrada de LEL, se deben modificar las responsabilidades de la tarjeta de CRC asociada. Luego de realizada la modificación, la entrada del LEL y la tarjeta CRC, quedan como se muestra en figura 13 y 14 respectivamente.

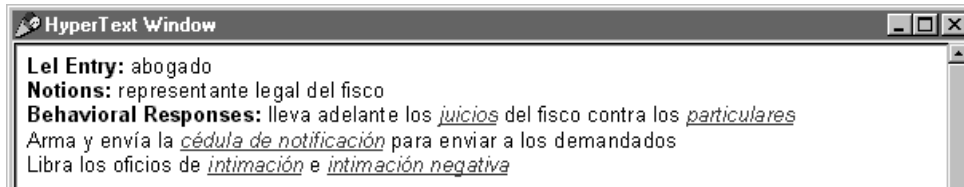


Figura 13. Entrada de Lel abogado

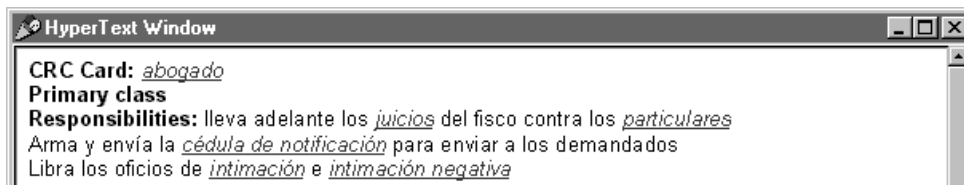


Figura 14. Tarjeta CRC abogado

La modificación de las responsabilidades de una CRC primaria, se concuerda con la situación de la regla 10. Específicamente se agregan referencias a dos entradas del LEL (intimación e intimación negativa) que no tienen tarjeta CRC asociadas. Como resultado se crean dos tarjetas CRC secundarias nuevas, las mismas se presentan en figura 15 y 16.

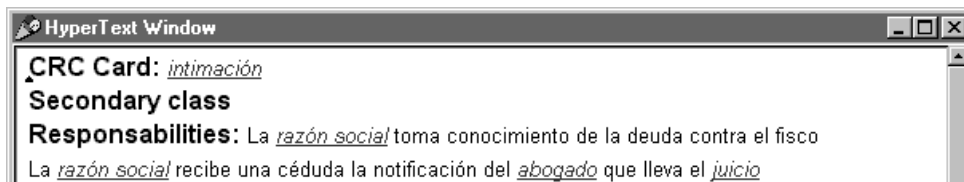


Figura 15. Tarjeta CRC asociada a la entrada de LEL intimación

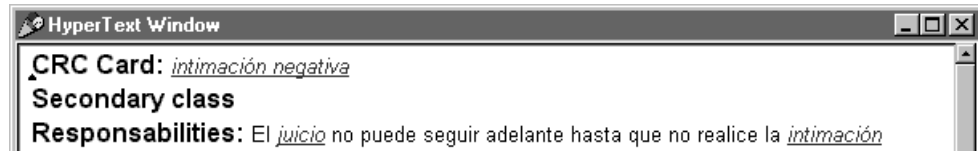


Figura 16. Tarjeta asociada a la entrada de LEL intimación negativa

## 8. Trabajos futuros

Se centrarán en extender el modelo de traceability. Hasta ahora se parte del LEL y escenarios, para llegar a las tarjetas CRC. Se pretende modelizar los elementos del mundo real que son las fuentes para la construcción del LEL y los escenarios. Y poder así hacer traceability desde las fuentes.

Al mismo tiempo se pretende mejorar el BMW. Se está llevando a cabo un proceso de evaluación por parte de estudiantes de posgrado, los cuales deben utilizar la herramienta y completar una encuesta sobre la usabilidad de la herramienta. Con el análisis de los formularios se pretende optimizar a BMW, como herramienta de soporte para la ingeniería de requerimientos.

### Nota

Los autores agradecen a un revisor anónimo de una versión anterior de este artículo, cuyas indicaciones resultaron valiosas para elaborar esta versión.

## Referencias

- [1] Leite, J.C., Oliveira, A., "A Client Oriented Requirements Baseline", *Second IEEE International Symposium on Requirements Engineer*, IEEE Computer Society Press, York, England, Marzo 27-29, 1995, pp 108-115.
- [2] Antonelli, L., Oliveros, A., Rossi, G., "Baseline Mentor, An Application that Derives CRC Cards from Lexicon and Scenario" *XXVIII JAIO, II Workshop Iberoamericano en Ingeniería de Requerimientos, WER '99*, Buenos Aires, Argentina, Septiembre 9 y 10, 1999.
- [3] IEEE, "IEEE Guide to Software Requirements Specifications", *ANSI/IEEE Standard 830-1984*, 1984.
- [4] Gotel, O.C.Z., Finkelstein, A.C.W., "An Analysis of the Requirements Traceability Problem", *International Conference on Requirements Engineering, ICRE'94*, Los Alamitos, California, Abril, 1994, pp 94-101.
- [5] Wieringa, R., *An Introduction to Requirements Traceability*, Reporte Técnico, Faculty of Mathematics and Computer Science, Vrije Universiteit, Esprit Project 2RARE, Noviembre 1, 1995.
- [6] Davis, A., *Software Requirements. Objects, Functions and States*, Englewood Cliffs, Prentice Hall, New Jersey, 1993.
- [7] Katonya, G., Sommerville, I., *Requirements Engineering, Processes and Techniques*, Chichester, John Wiley & Sons, 1998.
- [8] Sommerville, I., Sawyer, P., *Requirements Engineering*, Chichester, John Wiley & Sons, 1997.

- [9] Gotel, O.C.Z., "Contributions Structures for Requirements Traceability", University of London, Department of Computing, Imperial College of Science and Medicine, PhD Thesis, Agosto, 1995.
- [10] Pinheiro, F.A.C., "Formal and Informal Aspects of Requirements Tracing", *III Workshop de Engenharia de Requisitos, WER'2000*, Rio de Janeiro, Brasil, Julio, 2000.
- [11] Wiegers, K.E., *Software Requirements*, Microsoft Press, Redmond, 1999.
- [12] Hilburn, T.B., Hirmanpour, I., Khajenoori, S., Turner, R., Qasem, A., "A Software Engineering Body of Knowledge", Version 1.0, SEI, *Technical Report, CMU/SEI-99-TR-004, ESC-TR-99-004*, 1999.
- [13] Dömges, R., Pohl, K., "Adapting Traceability Environments to Project-specific Needs", *Communications of the ACM Vol 41 Nro 12*, Diciembre, 1998, pp 54-62.
- [14] Ramesh, B., "Factors Influencing Requirements Traceability Practice", *Communications of the ACM Vol 41 Nro 12*, Diciembre, 1998, pp 37-44.
- [15] Leite, J.C., Leonardi, C., Rossi, G., "Deriving Object Oriented Specification from External Scenarios", Provisto por el autor, 1997.
- [16] Wilkinson, N.M., *Using CRC Cards: An Informal Approach to Object-Oriented*, AT&T Bell Laboratories, 1995.
- [17] Rolland, C., "Modelling the Evolution Artifacts", *First International Conference on Requirements Engineering, ICRE'94*, IEEE Computer Society Press, Colorado Springs, Colorado, USA, Abril 18-22, 1994, pp 216-219.
- [16] Fowler, M., Scott, K., *UML Distilled: Applying the Standard Object Modelling Language*, Addison-Wesley, 1997.
- [19] Breitman, K.K., Leite, J.C., "Scenario Evolution: a Closer View on Relationship", *International Conference on Requirements Engineering, ICRE'2000*, Schaumburg, Illinois, USA, Junio 19-23, 2000.
- [20] Breitman, K.K., Leite, J.C., "A Framework for Scenario Evolution", *III IEEE International Conference on Requirements Engineering, ICRE'98*, Colorado Springs, Colorado, USA, Abril 6-10, 1998, pp 214-221.
- [21] Lavazza, L., Valetto, G., "Enhancing Requirements and Change Management through Process Modelling and Measurement", *Fourth International Conference on Requirements Engineering, ICRE'2000*, Schaumburg, Illinois, USA, Junio 19-23, 2000, pp 106-115.
- [22] Potts, C., "A Generic Model for Representing Design Methods", *11<sup>th</sup> International Conference on Software Engineering, ICSE'89*, Pittsburg, USA, 1989, pp 217-226.
- [23] Pinheiro, F.A.C., Goguen, J.A., "An Object Oriented Tool for Tracing Requirements", *IEEE Software Vol 13 Nro 2*, Marzo, 1996, pp 52-64.
- [25] Antonelli, L., Oliveros, A., "Traceability en la Etapa de Elicitación de Requerimientos", *Workshop de Investigadores en Ciencias de la Computación, WICC'2000*, La Plata, Argentina, Mayo 22-23, 2000.
- [26] Alspaugh, T.A., Antón, A.I., Barnes, T., Mott, B., "An Integrated Scenario Management Strategy", *Fourth International Symposium on Requirements Engineering, RE'99*, Limerick, Ireland, Junio, 1999, pp 142-149.
- [27] Schwabe, D., Rossi, G., "An Object Oriented Hypermedia Design Model", *Communication of ACM Vol 38 Nro 8*, 1995, pp 45-46.
- [28] Sommerville, I., *Software Engineering*, Addison-Wesley, 1995.