

Proceso y herramienta para la rastreabilidad de requisitos

Process and tool for requirements traceability

Marco Toranzo¹ Gilberto Cysneiros² Felipe Tirado¹

Recibido 29 de noviembre de 2011, aceptado 3 de abril de 2013

Received: November 29, 2011 Accepted: April 3, 2013

RESUMEN

El artículo presenta un proceso que identifica todos los *stakeholders* interesados en el proceso de rastreabilidad y los elementos del proyecto que ellos desean rastrear. Para cada *stakeholder*, los elementos de rastreabilidad son identificados, analizados, modelados y validados. Como resultado, un diagrama de clases es producido que muestra los elementos de rastreabilidad y sus relaciones, y un diagrama de caso de uso que muestra los servicios de rastreabilidad (relaciones) para cada *stakeholder* interesado en la rastreabilidad. El proceso es apoyado por un conjunto de directrices que usa un conjunto de tipos de relaciones predefinidas para desarrollar el modelo de rastreabilidad. Para apoyar el enfoque una herramienta fue creada y validada por un caso de estudio del sistema de biblioteca universitario.

Palabras clave: Proceso, directrices, herramienta, rastreabilidad, requisitos.

ABSTRACT

The paper presents a process that identifies all stakeholders of a project who are interested in the traceability process and the elements of the project that they want to track. For each stakeholder, traceability elements are identified, analyzed, modelled and validated. As result, a class diagram is produced that shows the traceability elements and their relationships, and a use case diagram that shows the traceability services (relationships) for each stakeholder interested in the traceability. The process is supported by a set of guidelines that uses a set of predefined relationship types that help to develop the traceability model. To support the approach a tool was created and validated through a university library system case study.

Keywords: Process, guidelines, tool, traceability, requirements.

INTRODUCCIÓN

La rastreabilidad de requisitos es un componente crítico de cualquier proceso riguroso de desarrollo de *software* [1], más aun cuando el costo de mantenimiento del *software* organizacional es del 70% [3-4] o puede variar entre 85% y 90% [2].

Dos importantes eventos relacionados directamente con la rastreabilidad, como *First Workshop on Grand Challenges for Traceability* (GCW'06) y el

International Symposium on Grand Challenges in Traceability (GCT'07) permitieron a un grupo de investigadores identificar y consensuar los desafíos de rastreabilidad que deben ser resueltos [2], como:

1. Definición de proceso, procedimiento, *benchmarks* para la rastreabilidad de requisitos.
2. Recuperación de las relaciones de rastreabilidad, por ejemplo, desde el documento de requisitos.
3. La evolución y mantenimiento de las relaciones de rastreabilidad.

¹ Departamento de Computación e Informática. Universidad Católica del Maule. Avenida San Miguel 3605. Talca, Chile. E-mail: mtoranzo@ucm.cl; ftirado@ucm.cl

² Departamento de Estatística e Informática. Universidade Federal Rural de Pernambuco. Rua Dom Manoel de Medeiros s/n, Dois Irmãos - CEP: 52171-900. Recife, Brasil. E-mail: g.cysneiros@soi.city.ac.uk

4. Definir la semántica de las relaciones de rastreabilidad.
5. La creación de métodos y herramientas para la elaboración, visualización, mantenimiento y evolución de relaciones de rastreabilidad.

Recientemente se publicó el informe titulado “Código Crítico: productividad del *software* para la Defensa” [5], que considera las necesidades del Departamento de Defensa (DoD de Estados Unidos) y las prioridades de investigación en el área *software*, así como también sugiere una agenda de investigación y acciones relacionadas. En el informe la rastreabilidad de requisitos juega un papel importante en términos de:

1. Mejorar el soporte para la rastreabilidad (página 10).
2. Aseguramiento de *software* (página 15).
3. Generación de código (página 79).
4. Localizar la correcta documentación y la consistencia de la misma con otros artefactos.

Gotel [6-7] plantea otros problemas y desafíos para la rastreabilidad, como:

1. Cambiar la percepción que la rastreabilidad es una tarea tediosa y repetitiva. La rastreabilidad de un proyecto necesita de constantes revisiones para mantener actualizados y consistentes los enlaces entre los diferentes los requisitos y otros artefactos del proceso de *software*.
2. Distribuir la responsabilidad de la rastreabilidad entre varias personas.
3. Planear una estrategia para asegurar la continuidad y mantenimiento de la rastreabilidad.
4. Determinar niveles de confiabilidad de la rastreabilidad.
5. Rastrear por producto.
6. Identificar los *stakeholders* interesados en la rastreabilidad y sus respectivas necesidades.

De acuerdo con Cleland-Huang[1], la rastreabilidad tiene varios desafíos, por ejemplo, entrenamiento y certificación; soporte a la evolución de las relaciones entre artefactos; semántica de las relaciones de rastreabilidad; y rastreabilidad por intermedio de la organización. Informaciones detalladas de estos y otros trabajos de rastreabilidad pueden ser encontradas en Cysneiro [8].

Las contribuciones del presente trabajo son presentadas desde la perspectiva de los problemas citados anteriormente, a saber:

- Problema 1: Definición de proceso, procedimiento, *benchmarks* para la rastreabilidad de requisitos. La contribución del artículo es la propuesta de un proceso para definir un modelo de rastreabilidad y un conjunto de directrices para elaborar un modelo de rastreabilidad de requisitos.
- Problema 2: Definir la semántica de las relaciones de rastreabilidad. Este artículo actualizó y refinó la semántica de un conjunto de tipos de relaciones para apoyar la elaboración de un modelo de rastreabilidad, propuesto en Toranzo [9-17], y mejorado posteriormente en este artículo. Esas relaciones fueron implementadas en el editor de MyMT (My Management Tool).
- Problema 3: identificar los *stakeholders* interesados en la rastreabilidad y sus respectivas necesidades. Respecto de este problema, el artículo presenta un proceso para elaborar un modelo de rastreabilidad. El proceso agrupa un conjunto de directrices para elaborar un diagrama de caso uso que ilustra todos los *stakeholders* con sus respectivas necesidades. Las directrices fueron refinadas a partir de los trabajos de Toranzo [9-17] y de su adaptación y aplicación por Castro [18], Pinto [19, 22-24] y Castor [20-21] en las arenas del paradigma de orientación a agentes.
- Problema 4: La creación de métodos y herramientas para la elaboración, visualización, mantención y evolución de relaciones de rastreabilidad. Respecto de este problema, la contribución es el desarrollo de la herramienta MyMT para apoyar la elaboración de un modelo de rastreabilidad y la administración de requisitos. La herramienta MyMT es la sucesora de la herramienta Labrador [9] e incorpora nuevas funcionalidades, por ejemplo, un editor gráfico, administración de proyecto y manejo de plantilla de documentos.

La Figura 1 presenta la herramienta Labrador, la primera herramienta desarrollada dentro de nuestra investigación realizada en el área de rastreabilidad.

En la misma figura se muestra el establecimiento de las asociaciones entre clases, usando los tipos de relaciones (problema 2, citado anteriormente).

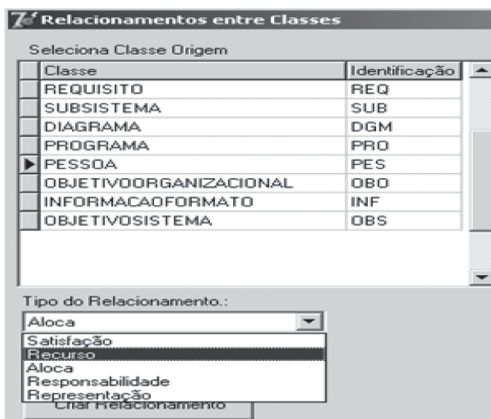


Figura 1. Herramienta Labrador.

MyMT todavía está en fase de desarrollo (80% del total), a la fecha se tiene implementado: un editor para modelar un modelo de rastreabilidad; permite administrar y relacionar requisitos; manipular documentos predefinidos (requisitos, propuesta de cambio, por ejemplo) e implementa una propuesta de tipos de relaciones para elaborar un modelo de rastreabilidad.

La parte restante del artículo está estructurada como sigue. La próxima sección presenta una clasificación de las informaciones rastreables de un proyecto. La siguiente sección identifica y explica las actividades del proceso para elaborar un modelo de rastreabilidad de requisitos. Posteriormente se presentan y ejemplifican las directrices para elaborar un modelo de rastreabilidad sobre un sistema de biblioteca y, en forma paralela, se presenta la herramienta MyMT. La última sección presenta las conclusiones del artículo y futuros trabajos.

INFORMACIONES RASTREABLES

Esta sección presenta una revisión de las contribuciones importantes en el área de rastreabilidad de requisitos y finaliza presentando una clasificación de las informaciones a rastrear dentro de un proyecto de desarrollo de *software*.

Los trabajos de investigación sobre rastreabilidad generalmente no formulan preguntas sistemáticas ni tampoco procesos sistemáticos para identificar los diferentes *stakeholders* con sus respectivas necesidades de rastreabilidad. Muchas investigaciones están preocupadas de responder las preguntas

relacionadas con recuperación, evolución y mantenimiento de las relaciones de rastreabilidad, como Corley [10], Klock [11] y Czauderna [12], cuyos trabajos tratan con la recuperación de relaciones (o *link*) de rastreabilidad.

En Corley [10] se emplea la herramienta *BugTrace* que toma como entrada un repositorio Bugzilla, el correspondiente CVS y (opcionalmente) un ID del error de interés. La salida es un conjunto de link de rastreabilidad recuperados y cada uno de ellos está asociado con un método modificado en un código fuente.

En Klock [11] se propone un *plugin* para Eclipse que permite a los desarrolladores de *software* especificar, ver y manipular relaciones de rastreabilidad dentro de Eclipse, además proporciona un API mediante el cual las técnicas de recuperación se pueden añadir, se especifican, y se ejecutan dentro de un ambiente de desarrollo integrado.

El trabajo de Czauderna [12] está relacionado con la aplicación de métodos de recuperación de información que permitan la generación de *link* de rastreabilidad, por ejemplo, mejorar la calidad de los *links* de rastreabilidad generados con la computación de los valores *idf* (*inverse document frequency*). En su trabajo se usa VSM (*vector space model*) donde cada consulta y cada documento es representado por un vector, en el caso del documento el vector contiene los pesos de cada uno de los términos del mismo documento.

Parte de la contribución de este artículo consiste en proponer una clasificación de las informaciones rastreables que ayuden a responder preguntas, como ¿Quiénes son los *stakeholders* de la rastreabilidad de un proyecto?, ¿cuáles artefactos podrían ser rastreados?, ¿cuáles son las relaciones de rastreabilidad (matrices) de interés de cada *stakeholders*? El hecho de que una organización tenga una herramienta para administrar requisitos no significa que la misma disponga de un proceso para identificar y administrar aspectos relacionados con rastreabilidad.

La Figura 2 presenta los cuatro niveles de la clasificación propuesta: Externo, Estratégico, Administración de proyecto y Operativo. Los niveles de información no son necesariamente disjuntos.

La clasificación está fundamentada en que las organizaciones están insertas en un entorno legal, político y económico que está en constante evolución y cuyos cambios pueden afectar los sistemas de información organizacionales.



Figura 2. Informaciones rastreables.

El nivel externo engloba todas las informaciones relacionadas con el contexto político, legal y económico, externos a la organización, que pueden afectar sus sistemas de información, por ejemplo, la modificación de las leyes de asignación familiar y maternal reguladas por el Ministerio del Trabajo y Previsión Social obligan a las organizaciones a realizar mantenimientos adaptativos a su sistema de remuneraciones. Otro ejemplo es la portabilidad numérica realizada por la Subsecretaría de Telecomunicaciones del Gobierno de Chile que llevó a las empresas telefónicas a ajustar sus sistemas para otorgar a los usuarios el derecho a trasladarse libremente de compañía manteniendo su número telefónico local.

El nivel estratégico representa los conceptos como objetivos, estrategias y metas organizacionales y procesos organizacionales. La combinación de algunos/todos los conceptos del nivel estratégico gatillan las adquisiciones y/o desarrollo de sistemas, entonces los sistemas de información deben estar alineados con la organización. Luego, es importante que los analistas de sistemas conozcan dónde se inicia y termina un proceso organizacional, dónde se inician y terminan las necesidades organizacionales para que se desarrolle un *software* alineado con los procesos de negocios de la organización y no lo contrario.

En el nivel de administración de proyecto se puede afirmar que, con excepción de Ramesh [28], la mayoría de las investigaciones entregan poca atención a cómo la rastreabilidad puede integrarse con la administración

de proyectos de *software*. De acuerdo con Humphrey [13], uno de los grandes problemas que afectan el desarrollo de *software* es la mala administración de los proyectos. Luego, los gerentes de proyectos de *software* deberían estar preocupados de que su plan de proyecto de *software* está basado en la satisfacción y cambios de requisitos [14], y en prestar atención a cómo la actividad de rastreabilidad de requisitos es realizada porque ella forma parte de un proceso de desarrollo y mantenimiento de *software* que debe liberar un producto de calidad. En particular, nos interesa establecer una relación entre las tareas (de la administración de proyecto) y los requisitos del proyecto para ayudar a los gerentes de proyectos a mejorar el control y seguimiento de los requisitos en las diferentes etapas de un proceso de desarrollo de *software*.

La herramienta MyMT permite definir las tareas de un proyecto y relacionarlas con los requisitos. Actualmente se trabaja en MyMT para permitir al analista y programador definir y adjuntar su propia carta *gant* a la tarea que le fue asignada por el jefe de proyecto.

El nivel operativo representa los artefactos consensuados entre los *stakeholders* interesados en la rastreabilidad. En este nivel se ejecuta la estrategia para realizar la actividad de rastreabilidad manual, automática o semiautomática, es decir, ¿cómo se identifican los requisitos dentro de un texto?, ¿cómo se identifican las relaciones entre los requisitos?, ¿cómo se relacionan los requisitos con los artefactos? y ¿qué tipo de relación usar para relacionar dos ítems?

Algunos beneficios de la propuesta de los cuatro niveles de información son:

1. Presentar forma alternativa para identificar, estructurar y analizar las informaciones que pueden formar parte de un modelo de rastreabilidad.
2. Cada nivel tiene *stakeholders* interesados en la rastreabilidad para verificar que sus necesidades y restricciones son satisfechas por el *software*.
3. Puede contribuir a mejorar la práctica y entendimiento de la rastreabilidad en el desarrollo y mantenimiento de *software* porque la rastreabilidad es un problema de equipo y no el problema y solución de una persona.

ACTIVIDADES DEL PROCESO

La Figura 3 presenta un proceso iterativo e incremental para producir un modelo de rastreabilidad. El proceso se inicia con la actividad “requisitos y restricciones organizacionales” que engloba, por ejemplo, las informaciones, requisitos y restricciones (tiempo, recurso y normas a satisfacer) externas que pueden afectar el sistema a desarrollar y el trabajo de rastreabilidad a realizar dentro de una organización.

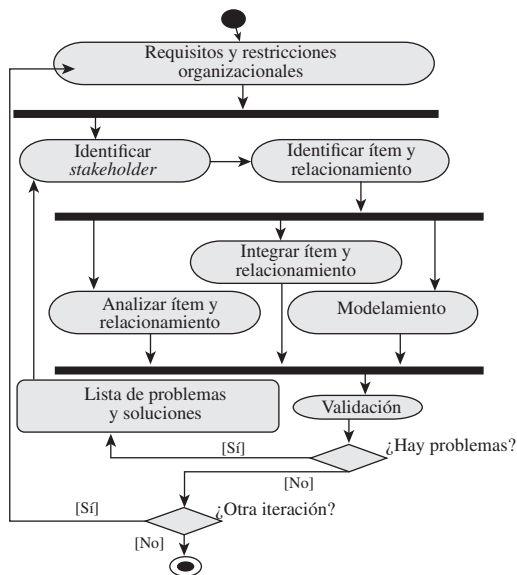


Figura 3. Actividades del proceso.

Después se identifican los *stakeholders* para proceder a obtener los ítems y relacionamientos. La actividad de obtención va acompañada de tareas concurrentes enlazadas con el análisis, integración y modelado de los ítems y relacionamientos. El producto resultante (diagrama) debe validarse. Si la validación identifica problemas se genera un listado de problemas y acciones para consensuarlos con los *stakeholders* afectados y encontrar las soluciones. Si la validación no encuentra problema, se evalúa la realización de una siguiente iteración. A continuación se explican algunas de las actividades concurrentes.

La actividad “identificar *stakeholder*” identifica e integra a las personas correctas e interesadas (entendidas para establecer acuerdos de rastreabilidad) para trabajar en asuntos de rastreabilidad, ya sea en forma individual y/o colectiva en actividades, como identificar, analizar, modelar y validar ítems y relacionamientos de rastreabilidad.

La actividad de “identificar ítem y relacionamiento” se basa en las reuniones con los *stakeholders* y en las fuentes de información disponibles (objetivos, requisitos, programas y las relaciones) para descubrir los ítems y relacionamientos pertinentes que deben formar parte del modelo de rastreabilidad del proyecto.

La actividad “analizar ítem y relacionamiento” evalúa si los ítems y relaciones identificados son pertinentes con las necesidades organizacionales/proyecto y restricciones existentes para rastrear los elementos identificados.

La actividad “integrar ítem y relacionamiento” incluye y asocia los ítems y relacionamientos de un *stakeholder* con los ya existentes en el modelo de rastreabilidad que fueron creados con las informaciones de otros *stakeholders* (internos/externos a la organización).

La actividad de “modelamiento” produce un modelo de rastreabilidad con las informaciones de los *stakeholders*. El modelo de rastreabilidad es un diagrama de clase que ayuda a comprender las preguntas ¿qué rastrear?, ¿qué tipo de relación usar para relacionar los ítems?, ¿cuál es el modelo de rastreabilidad del proyecto? El modelo de rastreabilidad es complementado con un diagrama de caso de uso que ayuda a comprender las preguntas ¿quiénes son los interesados en la rastreabilidad dentro del proyecto?, ¿cuáles son los intereses de rastreabilidad de cada *stakeholder*? Con posterioridad, ambos diagramas deben ser validados con los *stakeholders*. A la fecha, MyMT permite elaborar solamente el diagrama de clase. En la actualidad se trabaja en la implementación de un editor de diagrama de caso de uso que deberá estar integrado con el diagrama de clase, específicamente, con las relaciones.

Si dentro del proceso de elaboración de un modelo de rastreabilidad se desea comenzar a modelar, entonces lo primero a realizar en MyMT es definir un proyecto.

La Figura 4 presenta esa operación para crear un proyecto para el sistema de biblioteca.

Por último, la actividad de “validación” realiza una validación con los *stakeholders* a los que se

Figura 4. Creación de un proyecto.

les presenta los diagramas de clase y de caso de uso para facilitar la comunicación y comprensión. Si el resultado de la validación genera una lista de problemas, entonces para cada uno de ellos se debe analizar la no conformidad y realizar las actividades pertinentes para solucionar el problema.

DIRECTRICES PROPUESTAS

Esta sección presenta y ejemplifica un conjunto de directrices para desarrollar un modelo de rastreabilidad de requisitos para un sistema de biblioteca, empleando la herramienta MyMT. Los objetivos del sistema de biblioteca son integrar y gestionar los datos bibliográficos de todas las bibliotecas de una universidad que tiene diferentes sedes localizadas en distintas ciudades. El sistema administra la información de todos los usuarios, operadores, material bibliográfico y la circulación de las obras, entre otras cosas.

Las directrices fueron refinadas a partir de los trabajos de Toranzo [9-17], Castro [18], Pinto [19-22-23-24] y Castor [20-21]. La actual propuesta mejoró los metamodelos de las relaciones; eliminó relaciones; eliminó y agregó directrices; organizó las directrices dentro de un proceso y las necesidades de rastreabilidad de los usuarios mediante un diagrama de caso de uso.

La Figura 5 presenta la implementación del requisito “préstamos de obra” en el sistema de biblioteca.

La Figura 6 presenta la implementación del requisito “ingreso de obras”. La visión general

del sistema bibliotecario será complementada con otras ilustraciones y explicaciones presentadas en el artículo.

Figura 5. Préstamos de obras.

Figura 6. Ingreso de obras.

El conjunto de directrices para elaborar un modelo de rastreabilidad puede ser clasificado como sigue:

- a) Recolección de información
 - Identificar las personas del grupo de desarrollo de *software* y de la alta dirección interesadas en la rastreabilidad
 - Identificar las informaciones externas a la organización
 - Identificar los objetivos, estrategias o reglas de negocio
 - Incluir informaciones de la administración de proyectos
 - Identificar subsistemas
 - Incluir la clase requisitos

- Identificar los diagramas
 - Identificar programas
 - Identificar documentos
- b) Estructuración y construcción del modelo de rastreabilidad
- Excluir las clases irrelevantes
 - Unificar las clases y relaciones con el mismo significado
 - Establecer relaciones entre las clases
 - Organizar las clases
 - Definir los atributos de las clases
 - Actualizar clases y relaciones
- c) Validación
- Validar las clases y relaciones identificadas.
- d) Matrices de rastreabilidad
- Definir una matriz para cada relación del modelo

Directriz 1: identificar las personas del grupo de desarrollo de *software* y de la alta dirección interesadas en la rastreabilidad. Todas esas personas pueden estar interesadas en saber como el sistema satisface las actividades del proceso organizacional que están bajo su responsabilidad. Los roles profesionales identificados fueron: analista, programador y jefe de proyecto (para mejorar la supervisión de los trabajos).

Directriz 2: identificar las informaciones externas a la organización que pueden afectar el sistema que será rastreado. La palabra “informaciones” se refiere a las leyes, estatutos, normas, entre otros, cuyos cambios pueden requerir un análisis de impacto sobre los sistemas de información organizacional para determinar los recursos y tiempo para satisfacerlos, por ejemplo, en Chile las compañías telefónicas demoraron aproximadamente un año para realizar la implantación de la portabilidad numérica de los teléfonos móvil y fijo.

Dentro del contexto de identificar las informaciones externas a la organización que pueden afectar sus sistemas, se recomienda formularse algunas preguntas: ¿cuáles requisitos dependen de organizaciones externas?, ¿cuáles requisitos dependen de la alta dirección que pueden afectar el sistema? El resultado de la aplicación de la directriz fue la creación de la clase *InformFormato* para referenciar las informaciones de los formatos de visualización y de catalogación de obras, llamados *Machine-Readable Cataloging* y *Anglo-American Cataloging Rules*.

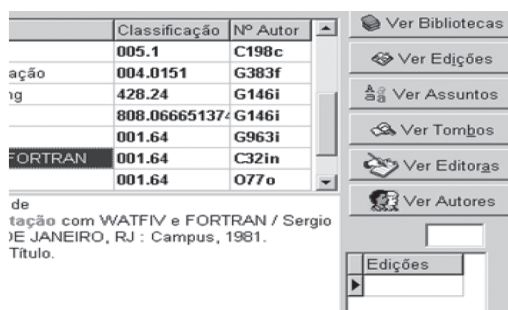


Figura 7. Clasificación de obra.

La Figura 7 presenta la implementación de la clasificación de obra en el sistema de biblioteca. La fuente de información de los datos obtenidos fue el jefe de biblioteca.

Directriz 3: identificar los objetivos, estrategias y reglas de negocio que serán rastreables. La identificación de los objetivos debe permitir la obtención de los requisitos de un sistema que pueden ser complementados con las reglas de negocio de la organización, por ejemplo, las bibliotecas de universidades otorgan diferentes períodos de tiempos a los préstamos de libros para profesores y estudiantes.

En el contexto de la aplicación de la directriz 3 se recomiendan algunas preguntas: ¿Cuáles son los objetivos organizacionales que debe satisfacer el sistema?, ¿cuáles son las reglas de negocio implementadas en el sistema?, ¿cuáles requisitos están relacionados con las reglas de negocio?

La aplicación de la directriz permitió identificar los siguientes objetivos organizacionales:

1. Administrar y centralizar las informaciones de todos los usuarios del sistema de biblioteca.
2. Incorporar sistemas de clasificación de obras nacionales e internacionales.
3. Incluir normas internacionales para administrar y representar las informaciones bibliográficas.
4. Automatizar y controlar los préstamos y reservas de las obras.

La aplicación de la directriz 3 permitió crear la clase *ObjetivoOrganizacional* en el modelo de rastreabilidad.

Directriz 4: incluir informaciones de la administración de proyectos en el modelo de rastreabilidad. El objetivo de la directriz es recomendar la inclusión de la clase *Tarea* en el modelo de rastreabilidad. Después de una retrospectiva de todas las actividades realizadas durante el proyecto se decidió crear la clase *Tarea* dentro del proyecto, a pesar de no contar con una carta Gantt detallada, pero cuyo estado ayudó a los objetivos del presente artículo porque en ella estaban las tareas (reuniones) relacionadas con la identificación y validación de requisitos, entre otras cosas. La Figura 4 presentó la creación de un proyecto, mientras la Figura 8 presenta la implementación de la definición de una “Tarea” en MyMT.

Figura 8. Definiendo una tarea en MyMT.

Directriz 5: identificar subsistemas. Un importante desafío para los ingenieros de requisitos y arquitectos ha sido mantener los requisitos y arquitectura consistentes durante los cambios [15]. El sistema de biblioteca no es la excepción, ya que fueron identificados e implementados los subsistemas con sus respectivos requisitos. La arquitectura del sistema de biblioteca está basada en un modelo de repositorio central. Durante la puesta en producción del sistema de biblioteca se realizaron cambios en los requisitos y se reconoce que los cambios erosionan la estructura interna del sistema. Respecto de esto Ling [16] presenta el método CoRE (*Change-Oriented Requirements Engineering*) que permite anticiparse a las necesidades de cambio, pero cuyo objetivo básico es separar en capas los requisitos (de los *stakeholders* y la organización). CoRE separa los requisitos en cuatro capas de diferente “volatilidad y causa de cambio”. Desde lo más estable a los más volátiles, las capas son: patrones; restricciones funcionales; restricciones no funcionales; y las políticas y reglas

de negocio. Ciertamente el conocimiento sobre los patrones y las restricciones funcionales ayudan en el diseño e implementación del sistema para que las restricciones no funcionales, las políticas y reglas de negocio puedan ser cambiadas sin afectar al resto del sistema.

Respecto de la aplicación de la directriz “Identificar subsistemas”, el resultado fue la creación de la clase *Subsistema* para registrar y representar, por ejemplo, los subsistemas: administración de obra; administración de usuarios, administración de préstamos y administración de configuración del sistema.

Directriz 6: incluir la clase requisitos. Es obligatorio y obvio crear la clase requisitos. Una lección aprendida de la elaboración y aplicación de esta directriz es que las personas que trabajan en rastreabilidad siempre están pensando en los requisitos y en su satisfacción y localización.

Directriz 7: identificar los diagramas en los cuales fueron modelados los requisitos. Los objetivos de la directriz son: identificar los diagramas en los cuales fueron modelados los requisitos y definir el camino lógico. El camino lógico es la secuencia de nombres de carpetas (finalizando con el nombre del archivo) que identifica el lugar donde está almacenado el diagrama. El resultado de la aplicación de la directriz 7 fue la creación de la clase Diagrama. Por el momento la rastreabilidad disponible en MyMT permite saber el nombre del diagrama donde fue modelado el requisito, es decir, la granularidad es alta.

Directriz 8: identificar programas. El objetivo de la directriz es identificar los programas en los cuales fueron implementados los requisitos del sistema y definir un camino lógico para recuperarlos. El resultado de la aplicación de la directriz 8 fue la creación de la clase Programa.

Por el momento la rastreabilidad disponible en MyMT permite saber el nombre del programa donde fue implementado el requisito, es decir, la granularidad es alta. Una vez que se guarde y genere el modelo de rastreabilidad (Figura 12), MyMT permite detallar cada una de esas relaciones, por ejemplo, la Figura 9 ilustra el detalle entre Requisito y Programa. En el caso de un Programa se muestra

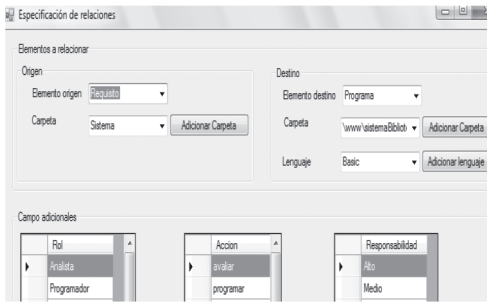


Figura 9. Relación requisito con programa.

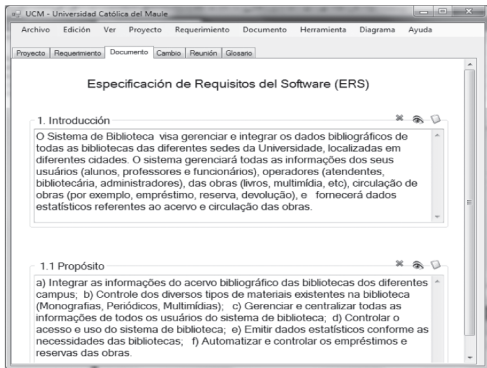


Figura 10. Documento de requisitos.

que se selecciona la carpeta de los archivos fuentes de los programas y el lenguaje de programación para facilitar la recuperación de los archivos.

Directriz 9: identificar documentos. Aquí son identificados los documentos del proyecto. La Figura 10 presenta la plantilla de especificación de requisitos disponible en MyMT, pero también se tienen otras plantillas, como documento de visión; casos de uso y propuesta de cambio.

Directriz 10: excluir las clases irrelevantes. Los *stakeholders* pueden establecer necesidades de rastreabilidad que no son soportadas por los servicios de la herramienta disponible en la organización y/o están fuera del alcance de los objetivos del proyecto y organización. Por ejemplo, la clase *Log* (registro de las operaciones realizadas por los bibliotecarios dentro del sistema) fue propuesta por el jefe de biblioteca, pero fue irrelevante para la actividad de rastreabilidad, ya que esa necesidad fue implementada en el sistema de biblioteca.

Directriz 11: unificar las clases y relaciones con el mismo significado. Considerando que los

ítems y relaciones de rastreabilidad son obtenidos de los diferentes *stakeholders* (interesados en la rastreabilidad) mediante diferentes reuniones individuales y/o grupales, es posible que puedan existir nombres de ítems y relaciones cuya funcionalidad es la misma. En caso del sistema de biblioteca no fue necesario unificar ítems.

Directriz 12: organizar las clases. Se deben organizar y estructurar las clases que formarán parte del modelo de rastreabilidad del proyecto. Dada la simplicidad del modelo de rastreabilidad del sistema de biblioteca no fue necesario organizar las clases usando el relacionamiento de generalización. La experiencia nos indica que los modelos de rastreabilidad deben ser simples y fáciles de implementar en una herramienta que soporte la rastreabilidad.

Directriz 13: establecer relaciones entre las clases. Esta directriz se aplica en forma paralela a la identificación de ítems y/o integración de clases candidatas de los diferentes *stakeholders*. Una forma alternativa y complementaria para comunicar las necesidades de rastreabilidad de los *stakeholders* fue con la presentación de un diagrama de caso de uso (ver Figura 11) que fue empleado también en la validación del modelo de rastreabilidad.

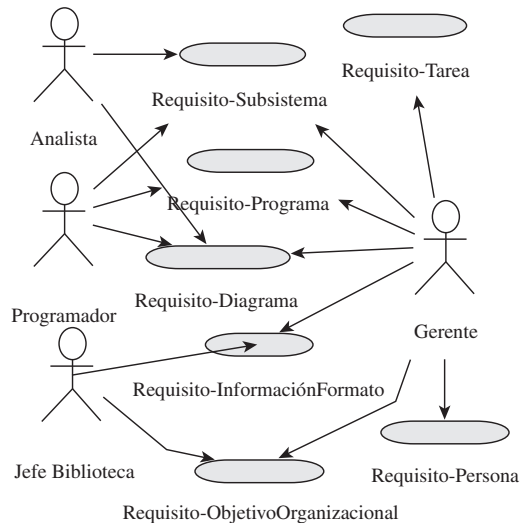


Figura 11. Diagrama de caso de uso.

La Figura 11 muestra que las personas interesadas en la rastreabilidad son: gerente, analista, programador y jefe de biblioteca; por ejemplo, el gerente está interesado en todas las relaciones; el programador

demonstró interés en rastrear requisitos con subsistema, programa y diagrama; ya el analista tenía sus preferencias de relacionar requisitos con subsistema y diagrama. Basados en el listado de ítems y relaciones de rastreabilidad y en el diagrama de caso de uso, la aplicación de la directriz 13 permitió la identificación de varias relaciones, por ejemplo, de la clase *Requisito* para las clases: *Programa*, *Persona* y *Subsistema*.

La Figura 12 presenta el editor gráfico de MyMT con todas las notaciones para los diferentes tipos de relaciones y las clases del modelo de rastreabilidad para el sistema de biblioteca.

MyMT implementa siete tipos de relaciones de rastreabilidad: agregación, generalización, recurso, localización, satisfacción, representación y responsabilidad.

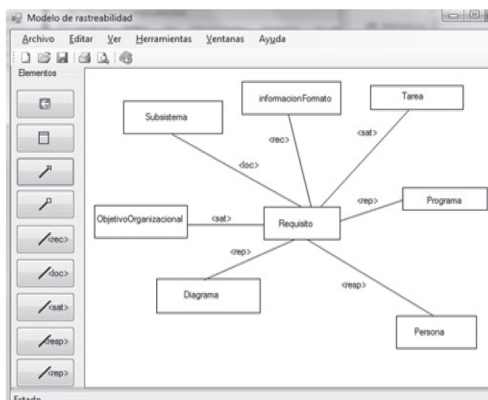


Figura 12. Modelo de rastreabilidad en MyMT.

Todas las relaciones son bidireccionales, excepto la agregación y generalización. En MyMT falta implementar los símbolos de cardinalidad. Las definiciones de los conceptos de agregación y generalización son los mismos definidos en la literatura de la orientación objetos. A continuación se explican los otros tipos de relaciones.

Todas las relaciones fueron organizadas en un metamodelo desarrollado por Toranzo [17] y que este trabajo refinó a partir de los resultados obtenidos de los trabajos de Castro [18], Pinto [19, 22-24], Castor [20-21] y Cysneiro [8, 24-26]. El trabajo de Cysneiro [8] propone nueve (9) tipos de relaciones organizadas en un modelo de referencia que forma parte de un *framework* de rastreabilidad basado en

reglas para la generación automática de relaciones de rastreabilidad.

La relación recurso (<Rec>) expresa que las instancias de la clase origen tienen una dependencia física o de información con las instancias de la clase destino, por ejemplo, en la Figura 12 se ilustró la dependencia de las instancias de *Requisito* con las instancias de *InformFormato* (formato de clasificación de obra) para expresar que algunas descripciones de los requisitos depende de las informaciones de clasificación de obras. La Figura 7 ilustró la implementación de las clasificaciones de obra dentro del sistema de biblioteca.

La relación satisfacción (<Sat>) especifica que las instancias de la clase origen tienen una dependencia de satisfacción alguna cosa (o conformidad) respecto de las instancias de la clase destino. Se usa el término satisfacción para expresar que algo deberá ser realizado para satisfacer alguna otra cosa, por ejemplo, la satisfacción de los objetivos organizacionales depende de los requisitos, es decir, el objetivo de la universidad de incentivar el préstamo de obras está relacionado con requisitos asociados con el catastro de usuarios y el control de préstamos.

La relación responsabilidad (<Resp>) especifica que las instancias de la clase origen tienen una dependencia de responsabilidad respecto de las instancias de la clase destino. Se usa el término responsabilidad para expresar que una persona es responsable de un artefacto/elemento dentro de un proyecto. La Figura 12 ilustra que las instancias de la clase *Persona* son responsables por instancias de la clase *Requisito*, es decir, existen *stakeholders* y desarrolladores asociados con requisitos.

La relación de representación (<Rep>) es para reflejar que un requisito está expresado (transformado) en otra notación sobre otro artefacto, por ejemplo, un requisito (un texto) es expresado en notación gráfica en un diagrama. La Figura 12 ilustra qué instancias de la clase *Requisito* están relacionadas con instancias de la clase *Diagrama*, es decir, existen requisitos que fueron modelados en diagramas.

La última relación, llamada de localización (<Loc>), ayuda a expresar que un requisito está asignado a un subsistema. Entender la relación entre los requisitos

y la arquitectura es importante, ya que los requisitos representan las funciones y la arquitectura determina la forma [27]. Bass [27] identifica varios problemas de relacionar requisitos con la arquitectura, como:

1. Existen varios *stakeholders* con sus propias agendas e intereses que es difícil de consensuar.
2. La determinación de todos los requisitos del proyecto es difícil.
3. No todos los requisitos pueden ser implementados con el tiempo establecido y el presupuesto asignado.
4. El ambiente en el cual está siendo desarrollado el sistema cambiará, por ejemplo, cambiará el ambiente tecnológico, el ambiente legal y ambiente competitivo.

A modo de ilustrar otras relaciones que pueden ser establecidas en MyMT, la Figura 13 ilustra la asociación entre Requisitos y Tarea.

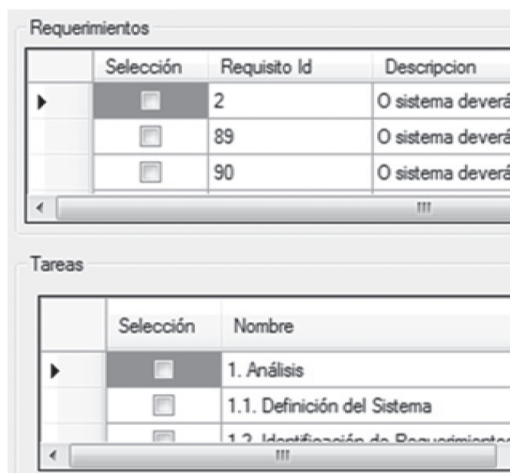


Figura 13. Asociando requisitos y tareas.

Directriz 14: definir los atributos de las clases. Se deben definir los atributos de las clases del modelo de rastreabilidad mientras se identifican y analizan las clases candidatas. En la Figura 14 se definen los atributos de la clase requisito.

En la definición se deben informar el nombre del campo; tipo de datos; longitud del campo; si es obligatorio u opcional; y el controlador visual asociado (*textbox, comboBox*, por ejemplo).

Las directrices del 1 al 14 fueron aplicadas para desarrollar e identificar las clases y relaciones

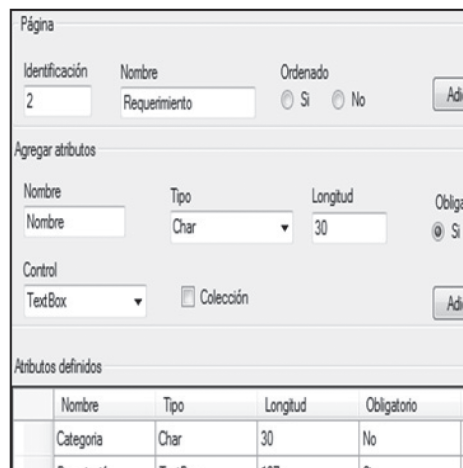


Figura 14. Definiendo los atributos.

candidatas del modelo de rastreabilidad del sistema de biblioteca. Sin embargo, eso no es suficiente porque la necesidad de cambio está presente en cualquier proyecto, independiente de su tamaño y complejidad. Este proyecto no fue la excepción, y por eso fue incorporada una directriz para tratar con las actualizaciones (cambios) realizadas sobre el modelo de rastreabilidad.

Directriz 15: actualizar clases y relaciones. Durante la ejecución del proyecto se consensuaron las necesidades de los programadores, analistas y jefe de proyecto en un modelo preliminar de rastreabilidad, pero fue posteriormente modificado con la incorporación de las necesidades del jefe de biblioteca que no había participado hasta ese instante de ninguna de las reuniones para elaborar el modelo de rastreabilidad.

Hasta el momento la aplicación de las directrices (1 al 15) está focalizada en la generación y actualización del modelo de rastreabilidad que está siendo elaborado para un proyecto. Es necesario validar el modelo con los *stakeholders* para consensuar el modelo de rastreabilidad del proyecto. Para esto se creó una directriz de validación.

Directriz 16: validación de las clases y los relacionamientos. La validación consiste en reunir y presentar a todos los *stakeholders* el diagrama de clase (parte estructural del trabajo) y el diagrama de caso de uso (usuario con sus respectivas necesidades de rastreabilidad). Si la validación identifica problemas, un listado de errores se

generará y posteriormente una o varias directrices presentadas deberán ser realizadas, caso contrario se procede a definir las matrices de rastreabilidad.

Directriz 17: definir una matriz para cada relación del modelo. El objetivo es desarrollar varias matrices para representar las relaciones del modelo de rastreabilidad. La Figura 15 es un ejemplo de la representación matricial de la relación responsabilidad entre *Persona* y *Requisito* cuya representación matricial tiene tres (3) argumentos: el papel de la persona; actividad a ser realizada; y grado de responsabilidad sobre la actividad.

	PES	REQ-101	REQ-102
PES-1		<ana, avaliar, A>	<prg, programar, ...>
PES-2		<ana, avaliar, A>	<test, depurar, A>
PES-3		<prg, programar, ...>	<prg, programar, ...>

Figura 15. Representación matricial.

La intersección de una fila con una columna representa la relación entre las *Personas* y *Requisitos*, por ejemplo, la representación matricial de la Figura 15 indica que la intersección entre el requisito “REQ-102” y la persona “PES-2”, es representado por “<test, depurar, A>”, esto significa que la persona PES-2 realizó pruebas (primera componente “test”) para depurar (segunda componente “depurar”) la implementación del requisito REQ-102, y fue altamente responsable (tercera componente “A”) sobre esa tarea.

CONCLUSIONES

Este artículo presentó un proceso; una clasificación de informaciones que pueden ser rastreadas; un conjunto de directrices para elaborar un modelo de rastreabilidad sobre un sistema de biblioteca; y la herramienta MyMT.

La relevancia del trabajo radica en proponer un proceso que apoye la identificación, análisis y

validación de los ítems que deben formar parte de un modelo de rastreabilidad y que, a partir del mismo, ayude a obtener las matrices de rastreabilidad. Dentro de un proyecto no es recomendable improvisar la rastreabilidad porque se puede perder tiempo, recurso y se corre el riesgo de desechar y hacer retrabajo.

Las ventajas del trabajo pueden ser analizadas desde diferentes perspectivas:

1. Proceso. Se define un conjunto de actividades orientadas a elaborar un modelo de rastreabilidad. Los datos de entradas pueden ser de uno de los siguientes segmentos de información: externo; estratégico; administración de proyecto; y operativo. Las salidas del proceso son los diagramas de clase y de caso de uso.
2. Conjunto de directrices. Las directrices guían al usuario mediante el proceso para identificar los diferentes ítems y relaciones que deben formar parte de un modelo de rastreabilidad. Las directrices usan los segmentos de información (externo; estratégico; administración de proyecto; y operativo) para extraer datos.
3. La herramienta MyMT. La propuesta de trabajo incluye la construcción de la herramienta MyMT, la segunda herramienta de administración de requisitos desarrollada por integrantes del grupo de investigación. En forma paralela Cysneiro [8] ha desarrollado un *framework* basado en reglas para la generación automática de relaciones de rastreabilidad.
4. Caso de estudio. La validación del proceso y de las directrices fueron realizadas sobre un sistema de biblioteca universitario. A pesar de tener otros casos de estudio, faltan nuevos proyectos para validar y refinar la propuesta de trabajo presentada en este artículo. Actualmente se están desarrollando siete (7) nuevos proyectos para validar el enfoque de trabajo propuesto en este artículo.

Futuros trabajos incluyen implementar mecanismos de identificación de requisitos en texto y establecer relaciones entre los requisitos escritos en lenguaje natural. También se trabaja para tener una granularidad final en las relaciones entre requisitos y otros artefactos. MyMT continúa en desarrollo y está siendo implementado en Visual Basic .NET y Mysql 5.051b.

En forma paralela se está implementando un sistema *Web* para la administración de requisitos, llamado *SimplyRequirement*, implementado con Mysql 5.051b y el *framework Cakephp 2.0.3*.

REFERENCIAS

- [1] A. Cleland-Huang, A. Czauderna, A. Dekhtyar, O. Gotel, J. Huffman, E. Keenan, G. Leach, J. Maletic, D. Poshyvanyk, Y. Shin, A. Zisman, G. Antoniol, B. Berenbach, A. Egyed and P. Maeder. "Grand Challenges, Benchmarks, and Tracelab: Developing Infrastructure for the Software Traceability Community". Proceeding of the 6th international workshop on Traceability in emerging forms of *software* engineering. ACM., pp. 17-23. New York, NY, USA. 2011.
- [2] I. Sommerville. "Software Engineering". Pearson Education. 9th edition. NJ, USA. 2011.
- [3] P. Grubb and A. Takang. "Software Maintenance: Concept and practice". World Scientific Publishing. 2nd edition. NJ, USA. 2003.
- [4] G. Alkhatib. "The maintenance problem of application *software*: An empirical analysis". Journal of Software Maintenance: Research and Practice. Vol. 4, Issue 2, pp. 83-104. June, 1992. DOI: 10.1002/smr.4360040203.
- [5] National Research Council of the National Academies. "Critical Code: Software Producibility for Defense". National Academies Press. Washington D.C., USA. 2010.
- [6] O. Gotel. "Traceability-Problems in a Word". The Newsletter of the Requirements Engineering Specialist Group of the British Computer Society. RQ49. 2008.
- [7] O. Gotel. "Traceability-Putting the 'y' First". Requirements Quarterly: The Newsletter of the Requirements Engineering Specialist Group of the British Computer Society. RQ50. 2009.
- [8] G. Cysneiros. "Software Traceability for Multi-Agent System Implemented using BDI Architecture". Doctoral Thesis, City University London. London. 2011.
- [9] M. Toranzo e E. Mellon. "Uma proposta para melhorar o rastreamento de requisitos". Workshop de Ingeniería de Requisitos. pp. 194-209. Valencia, España. 2002.
- [10] C. Corley, L. Etzkorn, N. Kraft and S. Lukins. "Recovering Traceability Links between Source Code and Fixed Bugs via Patch Analysis". Proceeding of the 6th international workshop on Traceability in emerging forms of *software* engineering. ACM., pp. 31-37. New York, NY. 2011.
- [11] S. Klock, M. Gethers, B. Dit and D. Poshyvanyk. "Traceclipse: An Eclipse Plug-in for Traceability Link Recovery and Management". Proceeding of the 6th international workshop on Traceability in emerging forms of *software* engineering. ACM., pp. 24-30. New York, NY. 2011.
- [12] A. Czauderna, M. Gibiec, G. Leach, Y. Li, Y. Shin, E. Keenan and J. Cleland-Huang. "Traceability Challenge 2011: Using TraceLab to Evaluate the Impact of Local versus Global IDF on Trace Retrieval". Proceeding of the 6th international workshop on Traceability in emerging forms of *software* engineering. ACM, pp. 75-78. New York, NY, USA. 2011.
- [13] W. Humphrey and W. Thomas. "Reflections on Management: How to Manage Your Software Projects, Your Teams, Your Boss, and Yourself". Addison-Wesley. Boston, MA, Estados Unidos. 2010.
- [14] K. Wiegers. "Software Requirements". Microsoft Press. 2nd edition. Redmond, WA, Estados Unidos. 2003.
- [15] J. Grundy, P. Lago, P. Avgeriou, J. Hall, and I. Mistrík. "Theoretical Underpinnings and Reviews". Relating Software Requirements and Architectures. Springer-Verlag, pp. 13-15. Berlin Heidelberg. 2011.
- [16] S. Ling and A. Finkelstein. "Anticipating Change in Requirements Engineering". Relating Software Requirements and Architectures. Springer-Verlag, pp. 17-34. Berlin Heidelberg. 2011.
- [17] M. Toranzo. "A Framework to Improve Requirements Traceability". Tesis de doctorado. Centro de Informática. Universidad Federal de Pernambuco. Brasil. 2002.
- [18] J. Castro, R. Pinto, A. Castor and J Mylopoulos. "Requirements Traceability in Agent Oriented Software Engineering". Lecture Notes in

- Computer Science. LNCS 2603. V. 2603, Springer-Verlag, pp. 57-72. 2003.
- [19] R. Pinto, C. Silva, T. Lima and J. Castro. "Support for Requirement Traceability: The Tropos Case". XIX Simpósio Brasileiro de Engenharia de Software-SBES'05, 2005, Uberlândia. Anais do XIX Simpósio Brasileiro de Engenharia de Software-SBES'05, pp. 40-55. 2005.
- [20] A. Castor. R. Pinto, C. Silva and J. Castro. "Towards Requirement Traceability in TROPOS". VII Workshop on Requirements Engineering. Proceedings of the VII Workshop on Requirements Engineering-WER04, Tandil. Argentina, pp. 189-200. 2004.
- [21] A. Castor. "Rastreamento de requisitos no processo de desenvolvimento de *software* orientado a agentes". Tesis de Magíster. Centro de Informática. Universidade Federal de Pernambuco. Brasil. 2004.
- [22] R. Pinto, J. Castro, P. Tedesco, M. Silva and F. Alencar. "A Traceability Reference Model for Agent Oriented Development". Proceedings of the Third Workshop on Software Engineering for Agent-oriented Systems. Editora Universitária da Paraíba/UFPB, pp. 27-38. João Pessoa-Brasil. 2007.
- [23] R. Pinto, J. Castro and M. Toranzo. "Requirements Traceability in Agent Oriented Development". SELMAS'2002-International Workshop on Software Engineering for Large-Scale Multi-Agent Systems. 2002. Orlando, Estados Unidos. 2010.
- [24] R. Pinto. "Improving Traceability in Agent Oriented Development". Tesis de Doctorado. Centro de Informática. Universidade Federal de Pernambuco. Brasil. 2004.
- [25] G. Cysneiros and A. Zisman. "Traceability for Agent-Oriented Design Models and Code". 19th International Conference on Software Engineering and Knowledge Engineering (SEKE 2007). USA. July, 2007.
- [26] G. Cysneiros and A. Zisman. "Traceability and Completeness Checking for Agent-Oriented Systems". 23rd Annual ACM Symposium on Applied Computing-Technical Track on Agent-Oriented Programming, Systems, Languages, and Applications. 2008.
- [27] L. Bass. "Preface". Relating Software Requirements and Architectures. Springer-Verlag, pp. V-VII. Berlin Heidelberg. 2011.
- [28] B. Ramesh and M. Jarke. "Towards Reference Models for Requirements Traceability". IEEE Transactions on Software Eng. Vol. 27, pp. 58-93. 2001.