

José Alfonso Aguilar Calderón

A goal-oriented approach for managing  
requirements  
in the development of Web applications

– PhD. Thesis –

Advisor: Irene Garrigos, Jose-Norberto Mazón López

Depto. Lenguajes y Sistemas Informáticos  
Universidad de Alicante



*Cada uno da lo que recibe,  
y luego recibe lo que da,  
nada es más simple,  
no hay otra norma:  
nada se pierde,  
todo se transforma.*

*Jorge Drexler, de su canción “Todo se transforma”.*



---

## Agradecimientos

A mis compañeros del grupo de investigación Lucentia del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante, Cristina Cachero, Jose Jacobo Zubcoff, Sergio Luján, Rafael Romero y Lily Muñoz por las discusiones de trabajo y toda la colaboración que me brindan.

A los miembros del grupo de investigación IWAD del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante, en especial a Jaime Gómez, Irene Garrigós y Santiago Meliá por el apoyo recibido durante este período de investigación.

A todos los miembros del grupo de investigación GPLSI del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante por todos los momentos compartidos.



---

## Acknowledgments

Thanks to....uff sooooo many people haha





---

## Preface

In recent years, there have been new proposals to address the development of such applications, some of them are mainly focused in the representation of the Web application at some level of abstraction (conceptual model), others meanwhile, are focused on specific tasks of the development process leaving aside the requirements phase. Moreover, because of the increasing complexity of the Web applications (i.e., changes in the platform implementation technology) and the multiple audiences involved in their use (i.e., heterogeneous audience), the requirements phase is more difficult to perform and to maintain. As a result, a problem emerges in these proposals: the absence of a design guide that facilitates the development of the Web applications based on the user's needs and expectations.

To overcome the lack of such a process, this PhD Thesis proposes a contribution to a existing model-driven approach for the development of Web 1.0 applications. Specifically, we propose the requirements specification in a conceptual model based in the  $i^*$  goal-oriented modeling framework, with which, the automatic derivation of the Web application conceptual models are possible. A requirements managing support is also proposed to avoid problems at the conceptual level with regard to (i) requirements traceability, (ii) change impact analysis and (iii) the design choices based on non-functional requirements maximization. Finally, as a proof of concept a set of *Eclipse* plugin's have been implemented. These are applied in a case study.

This PhD Thesis is composed of a set of published and submitted papers. In order to write this PhD Thesis as a collection of papers, several requirements must be taken into account as stated by the University of Alicante. With regard to the content of the PhD Thesis, it must specifically include a summary with regard to the description of the initial hypotheses, research objectives, and the collection of publications itself. This summary of the PhD Thesis includes the research results and the final conclusions. Finally, this summary corresponds to the part I of this PhD Thesis (chapter 1 has been written in Spanish while chapter ?? is in English).

It is important to highlight that this PhD Thesis has been developed under the PhD program “*Aplicaciones de la Informática*” of the Department of Software and Computing Systems (*Departamento de Lenguajes y Sistemas Informáticos*, DLSI) of the University of Alicante. This PhD work was funded by the CONACYT (*Consejo Nacional de Ciencia y Tecnología*, Mexico) and University of Sinaloa, Mexico.

Finally, this research was developed under the following projects: MANTRA (GRE09-17) from the University of Alicante, SERENIDAD (PEII-11-0327-7035) from Junta de Comunidades de Castilla La Mancha (Spain) and by the MESOLAP (TIN2010-14860) project from the Spanish Ministry of Education and Science and QUASIMODO (PAC08-0157-0668) projects from the Castilla-La Mancha Ministry of Education and Science (Spain).



---

## Contents

---

### Part I Summary

---

<b>1</b>	<b>Síntesis en Castellano</b> .....	<b>3</b>
1.1	Tesis Doctoral como Compendio de Artículos .....	3
1.1.1	Publicaciones Pertenecientes a la Tesis Doctoral .....	4
1.1.2	Artículos en Proceso de Revisión Pertenecientes a la Tesis Doctoral ....	5
1.1.3	Otras Publicaciones .....	6
1.2	Objetivos de Investigación e Hipótesis Inicial .....	6
1.3	Resumen del Contenido de la Tesis Doctoral .....	8
1.3.1	Una Aproximación Orientada a Objetivos para el Análisis de Requisitos en Ingeniería Web .....	8
1.3.2	Gestión de Requisitos en A-OOH .....	15
1.3.3	Implementación .....	18
1.4	Conclusiones .....	22
	References .....	22



## Part I

---

### Summary



## Síntesis en Castellano

A razón de que la tesis doctoral se ha realizado mediante la modalidad de compendio de artículos, este capítulo está dedicado a describir los objetivos, hipótesis y el conjunto de trabajos que la conforman. Además, es resumido el contenido científico de la tesis por medio de un resumen global de los resultados obtenidos así como de las conclusiones finales. Por último, cabe destacar que el contenido de este capítulo ha sido escrito en castellano, una de las lenguas oficiales de la Comunidad Autónoma, mientras que el capítulo siguiente corresponde a su traducción en inglés.

### 1.1 Tesis Doctoral como Compendio de Artículos

Los requisitos que debe cumplir una tesis doctoral para ser realizada en la Universidad de Alicante mediante un compendio de publicaciones fueron definidos por el Pleno de la Comisión de Doctorado de fecha 2 de marzo de 2005. A continuación, se exponen aquellos directamente relacionados con el contenido de la tesis:

1. *“La tesis debe incluir una síntesis, en una de las dos lenguas oficiales de esta Comunidad Autónoma, en la que se presenten los objetivos, hipótesis, los trabajos presentados y se justifique la unidad temática.”*
2. *“Esta síntesis debe incorporar un resumen global de los resultados obtenidos, de la discusión de estos resultados y de las conclusiones finales. Esta síntesis deberá dar una idea precisa del contenido de la tesis.”*
3. *“Los trabajos deben ser publicados, o aceptados para la publicación, con posterioridad al inicio de los estudios de doctorado. Los artículos en periodo de revisión pueden formar parte de la tesis como apéndices del documento, que debe presentarse adjunta a los artículos publicados.”*

Con el propósito de satisfacer los requisitos, la estructura de la tesis queda constituida en tres partes. La primera parte (Parte I) consiste en una síntesis de la tesis y se encuentra dividida en dos capítulos. El primer capítulo corresponde a la síntesis en castellano (Capítulo 1) y el segundo capítulo a su versión en inglés (Capítulo ??). La Parte ?? presenta el conjunto de artículos publicados que forman el núcleo de la tesis. Finalmente, la Parte ?? consiste en un apéndice donde se presentan dos trabajos que se encuentran actualmente en proceso de revisión.

Asimismo, es muy importante subrayar que la tesis doctoral ha sido materializada gracias al apoyo económico otorgado por el Consejo Nacional de Ciencia y Tecnología (CONACYT) México, por medio del Programa de Becas de Estudios de Posgrado en el Extranjero. Finalmente, es necesario destacar el interés y apoyo otorgado por parte de la Universidad Autónoma de Sinaloa, a través del Programa de Formación de Recursos Humanos en Áreas Estratégicas.

### 1.1.1 Publicaciones Pertenecientes a la Tesis Doctoral

A continuación se describen brevemente las cuatro publicaciones seleccionadas para que formen parte de la tesis doctoral. El criterio utilizado para la selección consistió en la relevancia y contribución científica de cada una de las publicaciones. Es decir, fueron seleccionadas los artículos publicados en revistas indexadas en JCR *Journal Citation Report* y en congresos ubicados en la clasificación CORE *Computer Research and Education*.

#### Capítulo ??

*J.A. Aguilar, I. Garrigós, J.-N. Mazón, J. Trujillo. Web Engineering Approaches for Requirements Analysis - A Systematic Literature Review. 6th Web Information Systems and Technologies (WEBIST 2010), Vol. 2, pp. 187-190, 2010.*

Este capítulo presenta los temas de estudio que dieron origen a la investigación asociada a la tesis doctoral, así como el material básico de referencia para comprender los detalles de la especificación, análisis y modelado de requisitos en ingeniería Web. Principalmente, el capítulo se enfoca en la revisión bibliográfica y el estado de la cuestión (ingeniería de requisitos en el dominio Web). Por último, presenta un análisis de las aproximaciones metodológicas más importantes en el ámbito de la ingeniería Web, enfocado únicamente en aspectos como el análisis y especificación de requisitos, trazabilidad y las herramientas de soporte ofrecidas por cada una de ellas.

#### Capítulo ??

*J.A. Aguilar, I. Garrigós, J.-N. Mazón, J. Trujillo. An MDA Approach for Goal-oriented Requirement Analysis in Web Engineering. Journal of Universal Computer Science (J.UCS), 16(17): 2475-2494 (2010).*

Este capítulo describe la propuesta base de la tesis. En el capítulo anterior, se realizó una revisión sistemática de la literatura para estudiar las técnicas ingenieriles en el desarrollo de aplicaciones Web. Los resultados demuestran que la mayoría de las aproximaciones se enfocan en las etapas de análisis y diseño, por tanto, no ofrecen un soporte integral a la fase de requisitos. La aproximación descrita en este capítulo, ha tomando como sustento las carencias detectadas en el capítulo anterior para desarrollar una aproximación basada en el marco de modelado orientado a objetivos  $i^*$  y en MDA (*Model-Driven Architecture*). Con la propuesta, es posible derivar los modelos conceptuales que conforman una aplicación Web 1.0 (*Platform Independent Models*) a partir de la especificación de requisitos (*Computational Independent Model*). Lo anterior, por medio de un conjunto de transformaciones descritas de manera formal utilizando el lenguaje QVT (*Query/View/Transformation*).

#### Capítulo ??

*J.A. Aguilar, I. Garrigós, J.-N. Mazón. Impact Analysis of Goal-Oriented Requirements in Web Engineering. The 11th International Conference on Computational Science and Its Applications (ICCSA 2011), June 20-23, 2011, Santander, Spain. Part V, Lecture Notes in Computer Science, Vol. 6786, pp. 421-436, 2011.*

En capítulos anteriores se ha resaltado la importancia de la etapa de análisis y especificación de requisitos en la ingeniería Web, obligada, principalmente, por la audiencia heterogénea y por la evolución constante en las tecnologías de implementación. Éstas características particulares de las aplicaciones Web ocasionan, en la mayoría de los casos, inconsistencias entre los requisitos. Por consiguiente, es importante conocer las dependencias entre los requisitos para garantizar, en lo posible, que la aplicación Web satisfaga las necesidades y expectativas de los usuarios. Comprender y analizar las dependencias entre los requisitos le permite al diseñador brindar una mejor gestión y mantenimiento de la aplicación Web. En este capítulo se presenta un algoritmo para manejar las dependencias, entre los requisitos funcionales y los requisitos



no-funcionales de la aplicación Web en un contexto orientado a objetivos (*goal-oriented*). Con el algoritmo, es posible comprender cuál es el impacto en los requisitos procedente de un cambio en los modelos conceptuales que conforman la aplicación Web, así como saber qué requisitos necesitan ser implementados para cumplir, en medida de lo posible, los proósitos establecidos en el análisis orientado a objetivos.

## Capítulo ??

*J.A. Aguilar, I. Garrigós, J.-N. Mazón. A Goal-Oriented Approach for Optimizing Non-Functional Requirements in Web Applications. The 8th th International Workshop on Web Information Systems Modeling (WISM 2011), held in conjunction with the International Conference on Conceptual Modeling (ER 2011), 31 October - 03 November 2011, Brussels, Belgium. Part X, Lecture Notes in Computer Science, Vol. X, pp. XXX-XXX, 2011.*

La idea de considerar los requisitos no-funcionales desde las etapas iniciales (requisitos) del proceso de desarrollo con el fin de mejorar la aplicación a desarrollar, ha sido objeto de investigación en el contexto del desarrollo dirigido por modelos (*Model-Driven Development*, por sus siglas en inglés). La idea se fundamenta en la implementación de los requisitos funcionales a partir de los requisitos no-funcionales. En este sentido, los requisitos no-funcionales deben de ser priorizados de acorde al contexto de los usuarios de la aplicación Web. En este capítulo se presenta una adaptación del algoritmo de Pareto (Frontera de Pareto) para evaluar y seleccionar la configuración de requisitos óptima que maximice los requisitos no-funcionales de la aplicación Web. La solución del algoritmo proporciona al diseñador de la aplicación Web un conjunto de configuraciones de entre las cuales podrá elegir qué requisitos funcionales implementar (configuración óptima) considerando la prioridad de los requisitos no-funcionales.

### 1.1.2 Artículos en Proceso de Revisión Pertenecientes a la Tesis Doctoral

En este apartado, se presentan dos trabajos que forman parte de la tesis doctoral. Los trabajos están actualmente bajo proceso de revisión.

## Apéndice ??

*Requirements in Web engineering: a systematic literature review. Este artículo se ha enviado a la revista XXXXXXXX.*

En este trabajo se realiza una profunda revisión del estado de la cuestión, en lo referente al análisis, especificación y trazabilidad de requisitos en ingeniería Web. Concretamente, se analizan: (i) las técnicas utilizadas por las metodologías ingenieríles en la etapa de análisis y especificación de requisitos, (ii) el tipo de requisitos y la terminología utilizada por cada metodología Web, (iii) el soporte para trazabilidad y (iv) las herramientas de soporte que ofrecen. Finalmente, el artículo es una extensión del Capítulo ??.

## Apéndice ??

*A Goal-Oriented Requirements Engineering Approach to Distribute Functionality in RIAs. Este artículo se ha enviado a 12th International Conference on Web Information System Engineering (WISE 2011).*

Como es sabido, la Web evoluciona constantemente y las metodologías Web deben ser adaptadas para lidiar, por ejemplo, con las tecnologías de implementación. Parte de la evolución son las aplicaciones RIAs (*Rich Internet Applications*), las cuales ofrecen, entre otras cosas, una mejor interactividad con el usuario, similar a la ofrecida por las aplicaciones *software* de escritorio. En este trabajo, se presenta la adaptación de la propuesta descrita en el capítulo ?? para auxiliar al diseñador Web en la distribución entre el cliente y el servidor de la funcionalidad de la aplicación RIA. Para lograrlo, se adaptó el algoritmo de Pareto para obtener, en lo posible, un equilibrio entre los requisitos funcionales y los no-funcionales.

### 1.1.3 Otras Publicaciones

En el transcurso de la investigación asociada a la tesis doctoral se han publicado (o enviado) cinco artículos a distintos eventos nacionales y/o internacionales. Los trabajos no han sido incluídos en el núcleo de la tesis, sin embargo, complementan el progreso de la investigación.

- J.A. Aguilar**, I. Garrigós, J.-N. Mazón. Aproximaciones en Ingeniería Web para el Análisis de Requisitos: una Revisión Sistemática de la Literatura. *Actas del IV Congreso Nacional de Informática y Ciencias de la Computación (CNICC 2009)*, Mazatlán, Sinaloa, México, 2009. ISSN: XXXXXXXX.
- J.A. Aguilar**, I. Garrigós, J.-N. Mazón. Modelos de *weaving* para Trazabilidad de Requisitos Web en A-OOH. *Actas del VII Taller de Desarrollo de Software Dirigido por Modelos (DSDM 2010) en XV Jornadas de Ingeniería de Software y Bases de Datos (JISBD 2010)*, en conjunto con el Congreso Español de Informática (CEDI), pp. 146-155. SISTEDES, Valencia, España, 2010.
- J.A. Aguilar**, I. Garrigós, J.-N. Mazón. Modelo Requisitos y Modelo de Dominio, trazabilidad mediante modelos de *Weaving*. *Actas de VIII Jornadas para el Desarrollo de Grandes Aplicaciones de Red (JDARE 2010)*. GrupoM, Alicante, España, 2010. ISBN: XXXXXXXX.
- J.A. Aguilar**, I. Garrigós, J.-N. Mazón. Automatic Generation of Conceptual Models from Requirements Specification in Web Engineering using ATL. *Actas de IX Jornadas para el Desarrollo de Grandes Aplicaciones de Red (JDARE 2011)*. GrupoM, Alicante, España, 2011. Enviado.
- J.A. Aguilar**, I. Garrigós, J.-N. Mazón. Una Propuesta Orientada a Objetivos para el Análisis de Requisitos en RIAs. *Actas de XVI Jornadas de Ingeniería de Software y Bases de Datos (JISBD 2011)*. La Coruña, España, 2011. Enviado.

## 1.2 Objetivos de Investigación e Hipótesis Inicial

De forma similar a los sistemas software desarrollados exclusivamente para un entorno de escritorio, los sistemas Web necesitan la aplicación de conceptos de ingeniería para obtener éxito en la aplicación Web final. Para lograrlo, es necesario definir técnicas y enfoques que consideren la gran variedad de usuarios, plataformas y entornos para su implementación. En este sentido, uno de los factores de éxito más importantes en el desarrollo de *software* es la elicitación, gestión y análisis de requisitos. Sin embargo, en el desarrollo de *software* en ingeniería Web llevar a cabo una correcta gestión de los requisitos es una tarea complicada. Principalmente, esto se debe a que la ingeniería Web enfrenta continuos cambios que dificultan la etapa de requisitos a razón de las características particulares de las aplicaciones Web, como el caso de: (i) la gran cantidad de información que ofrecen (contenido), (ii) el acceso a los diferentes escenarios donde ofrecen esa información (navegación), (iii) como proveer dicha información al usuario o grupos de usuarios (funcionalidad) del sitio Web y (iv), la audiencia heterogénea que tiene acceso a la Web. Como consecuencia de estos factores, los analistas, desarrolladores y diseñadores se enfrentan a retos cada vez más complejos para gestionar el diseño y mantenimiento de las aplicaciones Web. Por lo tanto, definir los requisitos (funcionales y no-funcionales) que el sistema debe cumplir para satisfacer las necesidades de los usuarios es una tarea que necesita una atención especial.

Actualmente, existen una notable cantidad de aproximaciones metodológicas para el desarrollo de aplicaciones Web (A-OOH [5], UWE [7], NTD [3], OOWS [14], etc.) que toman en cuenta la aplicación de distintas técnicas para llevar a cabo el desarrollo de aplicaciones Web. Sin embargo, la mayoría de las aproximaciones utilizan reconocidas técnicas de ingeniería de *software* para gestionar correctamente los requisitos de los usuarios, como el caso de UWE (casos de uso) [8]. Lamentablemente, la mayoría de las técnicas utilizadas por las metodologías resultan insuficientes para representar características muy particulares de las aplicaciones Web, tales como: la navegación y la audiencia heterogénea. Por lo tanto, es necesaria la inclusión de nuevas técnicas que permitan lidiar con las características particulares de las aplicaciones

Web y que posibiliten la correcta especificación de las necesidades de los diferentes actores implicados.

Por otro lado, el desarrollo dirigido por modelos (*Model Driven Development*, MDD) se ha convertido en una alternativa valiosa para resolver los problemas asociados con el desarrollo de *software* de manera sistemática, estructurada, integrada y completa mediante la utilización de modelos como artefactos principales en el proceso de desarrollo de las aplicaciones Web. El desarrollo dirigido por modelos es una aproximación al desarrollo de software basado en el modelado del sistema software y su generación a partir de los modelos. En este sentido, MDD sólo proporciona una estrategia general a seguir en el desarrollo de *software*, pero no define las técnicas a utilizar. El impacto de MDD en la ingeniería Web ha permitido la llegada de la ingeniería Web dirigida por modelos (*Model Driven Web Engineering*, MDWE) como una nueva aproximación para el desarrollo de aplicaciones Web. Su supuesto básico es la consideración de los modelos como entidades de primera clase que impulsan el proceso de desarrollo desde el análisis de requisitos hasta la implementación final. Básicamente, cada paso del proceso consiste en la generación de uno o más modelos de salida a partir de uno o más modelos de entrada. Por lo tanto, las transformaciones entre modelos son la clave para completar cada paso del proceso del proceso de desarrollo dirigido por modelos.

En este contexto, la arquitectura dirigida por modelos (*Model Driven Architecture*, MDA) es un estándar del OMG (*Object Management Group*) [13] que promueve el MDD y que se ha aplicado con resultados favorables al MDWE. MDA está formada por un conjunto de capas y transformaciones que proporcionan un marco conceptual en donde encontramos tres tipos de modelos, el primero de ellos es modelo independiente de la computación (*Computational Independent Model*, CIM), utilizado para la especificación de los requisitos de la aplicación a desarrollar, el segundo es el modelo independiente de la plataforma (*Platform Independent Model*, PIM), como su nombre lo indica, se caracteriza por ser independiente de la plataforma de implementación de la aplicación, por ejemplo, un diagrama de clases, y el modelo específico de la plataforma (*Platform Specific Model*, PSM), el cual es obtenido del PIM y contiene la información sobre una plataforma de desarrollo o alguna tecnología en específico donde será implementada la aplicación final, esto es, el código fuente de la aplicación [12].

Actualmente, MDA ha sido aplicado para el desarrollo de aplicaciones Web (OOWS [15], NDT y UWE [9]). Sin embargo, el trabajo presentado en la tesis doctoral es el primero que aborda el modelado conceptual de aplicaciones Web a partir del nivel CIM de MDA utilizando técnicas orientadas a objetivos para la obtención automática de modelos conceptuales a nivel PIM. De esta forma se asegura que los modelos conceptuales obtenidos a partir del nivel CIM de MDA, sean correctos semánticamente.

El **objetivo de investigación** de esta tesis doctoral es la propuesta de una metodología para el análisis de requisitos para aplicaciones Web 1.0, que considere:

- Una etapa de análisis de requisitos orientada a objetivos para representar las expectativas reales del usuario de la aplicación Web.
- Mecanismos para la comprensión de los objetivos de negocio que debe lograr la aplicación Web gracias al uso del análisis de requisitos orientada a objetivos.
- Soporte para la gestión de los requisitos en aspectos como la trazabilidad y el análisis de impacto.
- Considerar los requisitos no-funcionales desde la etapa de análisis y especificación de requisitos.
- Asistir al diseñador al momento de la selección de los requisitos funcionales a implementar a través de alternativas de diseño que consideren el balance y maximización de los requisitos no-funcionales.
- Semi-automatizar el desarrollo de aplicaciones Web por medio de un conjunto de transformaciones formales para obtener los modelos conceptuales a partir de la especificación de los requisitos.

Cabe destacar que la hipótesis de partida de la investigación asociada a la tesis doctoral consiste en que sí es factible el desarrollo de una metodología MDD-MDA que contemple una

etapa de análisis de requisitos que permita comprender los objetivos y expectativas reales de la audiencia heterogénea de una aplicación Web.

### 1.3 Resumen del Contenido de la Tesis Doctoral

El objetivo de investigación de esta tesis doctoral se aborda en dos etapas, la primera es la definición de una propuesta orientada a objetivos para el análisis y especificación de requisitos en ingeniería Web (Web 1.0). La finalidad de la primera etapa es la obtención de los modelos conceptuales de la aplicación Web (dominio y navegación). La segunda consiste en la especificación y aplicación de técnicas para la gestión de requisitos, concretamente, aquellas relacionadas a la trazabilidad de requisitos, análisis de impacto y maximización de requisitos no-funcionales.

#### 1.3.1 Una Aproximación Orientada a Objetivos para el Análisis de Requisitos en Ingeniería Web

En este apartado se resume la propuesta para el análisis y especificación de requisitos en ingeniería Web aplicada por medio del marco de modelado orientado a objetivos  $i^*$  y el método de ingeniería Web A-OOH (*Adaptive Object-Oriented Hypermedia*) [5].

A-OOH es la extensión del método OOH (*Object-Oriented Hypermedia*) [6] con estrategias de personalización. El proceso de desarrollo de A-OOH esta basado en MDA, es decir, los requisitos de la aplicación Web son definidos en un nivel CIM y a partir de son derivados los modelos conceptuales de la aplicación Web. Los modelos conceptuales corresponden al nivel PIM de la arquitectura MDA como puede verse en Fig. 1.1, estos son:

- **Modelo de dominio.** El modelo de dominio de A-OOH se expresa como un diagrama de clases UML (*Unified Modeling Language*) [16]. Este modelo refleja la parte estática de la aplicación Web encapsulando su estructura y funcionalidad. Los elementos principales para el modelado de un diagrama de clases son las clases (con sus atributos y operaciones) y sus relaciones.
- **Modelo de navegación.** El modelo de navegación de A-OOH se compone de nodos de navegación y las relaciones entre ellos. Este modelo indica los caminos de navegación que el usuario puede seguir en la Web (enlaces de navegación). Hay tres tipos de nodos: (i) clases navegacionales (que son vistas parciales de las clases de dominio), (ii) destinos navegacionales (que agrupan elementos del modelo que colaboran en el cumplimiento de uno o más requisitos de navegación del usuario) y (iii) colecciones (que son estructuras, posiblemente jerárquicas, que se definen entre clases de navegación o destinos navegacionales). La colección más común es la colección de clasificación (*C-collection*), que actúa como un mecanismo de abstracción para el concepto de menú agrupando enlaces de navegación. Con respecto a los enlaces de navegación, A-OOH define dos tipos principales: enlaces de travesía (*Transversal-links*) (definidos entre dos nodos de navegación) y enlaces de servicio (*Service-links*), en donde la navegación sucede al activar una operación que modifica la lógica de negocio y además implica la navegación a un nodo mostrando información cuando la ejecución del servicio ha finalizado.
- **Modelo de presentación.** Este modelo permite definir la interfaz gráfica de la aplicación Web, por ejemplo el tipo de fuente utilizada en el texto, el color, etc.
- **Modelo de personalización.** Este modelo es utilizado para la especificación de estrategias de personalización.
- **Modelo de usuario.** Este modelo permite la descripción de los usuarios en términos de información personal, sus relaciones con un dominio en particular y las acciones de navegación realizadas en tiempo de ejecución. La estructura de la información necesaria para la personalización también se describe en este modelo.
- **Modelo de *weaving*.** Es un tipo especial de modelo utilizado por A-OOH para almacenar referencias entre los elementos del modelo de requisitos y su correspondencia con los elementos de cada uno de los modelos conceptuales de A-OOH.

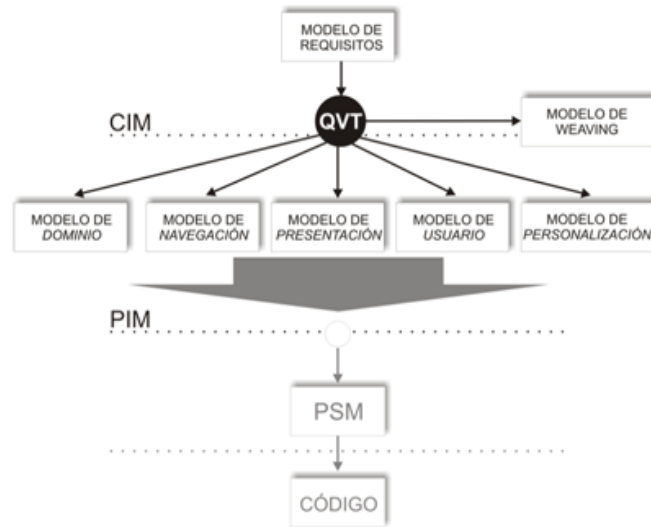


Fig. 1.1. Método Web A-OOH.

### Especificación del Modelo de Requisitos a Nivel CIM

El primer paso de la propuesta presentada en esta tesis doctoral es la especificación y el modelado de los requisitos de las aplicaciones Web. Para una explicación más amplia el lector puede referirse al Capítulo ?? de la tesis doctoral.

Los requisitos Web se definen en un modelo independiente de computación (CIM) utilizando el marco de modelado de requisitos  $i^*$ . El marco de modelado  $i^*$  [17] es uno de los más utilizados para analizar los objetivos de los *stakeholder's*<sup>1</sup> (actores en  $i^*$ ) y cómo el sistema a diseñar debería satisfacerlos. Además,  $i^*$  permite razonar acerca de cómo estos objetivos pueden contribuir a la selección de diferentes alternativas de diseño según su viabilidad. Con el fin de motivar esta parte de la investigación, se ha realizado una revisión del estado de la cuestión (ver apéndice ??)

El marco de modelado  $i^*$  consiste básicamente en dos modelos: el modelo SD (*Strategic Dependency*) para especificar las relaciones de dependencia entre varios actores en un contexto organizacional y el modelo SR (*Strategic Rationale*), utilizado para describir los intereses y preocupaciones del actor y como es que podrían abordarse. El modelo SR representa una manera detallada para modelar los elementos intencionales y las relaciones de cada actor. Los elementos intencionales son objetivos (*Goals*), tareas (*Tasks*), recursos (*Resources*) y *Softgoals*. Las relaciones intencionales son links del tipo *Means-end* los cuales representan formas alternativas para satisfacer objetivos; los *Decomposition-links* representan los elementos necesarios para que una tarea sea realizada; o los *Contribution-links* que sirven para modelar como es que un elemento intencional contribuye a la satisfacción de un requisito no funcional (*Softgoal*). En la Tabla 1.1 se describen los elementos más importantes del marco de modelado  $i^*$ .

Por otra parte, el marco de modelado  $i^*$  tiene que adaptarse al dominio de la ingeniería Web con el fin de poder modelar actores, sus objetivos y las relaciones entre ellos. De esta manera se podrían reflejar los requisitos específicos de Web que no son tenidos en cuenta en las aproximaciones de análisis de requisitos tradicionales como el caso de la navegación. Para realizar la adaptación de  $i^*$  al dominio Web nuestra propuesta utiliza la taxonomía de requisitos Web presentada en [4], la cual clasifica a los requisitos Web en seis tipos, estos son descritos a continuación:

<sup>1</sup> .....

**Table 1.1.** Principales elementos para el modelado en  $i^*$ 

Elemento	Descripción
ACTOR ( $\bigcirc$ )	Es una entidad que lleva a cabo acciones para cumplir con sus objetivos. Se relaciona con varios elementos intencionales (objetivo, tarea o recurso).
GOAL ( $\bigcirc$ )	Representa una condición o estado que el actor le gustaría alcanzar.
TASK ( $\nabla$ )	Representa una manera particular de hacer algo.
RESOURCE ( $\square$ )	Es una entidad que debe estar disponible para su uso.
MEANS-ENDS ( $\nabla \triangleright$ )	Son asociaciones que describen cómo se alcanzan los objetivos, es decir, los posibles caminos para satisfacer un objetivo.
DECOMPOSITION ( $\dashv$ )	Son asociaciones que definen elementos adicionales necesarios para llevar a cabo una tarea.

- **Requisitos de contenido (*Content*)**. Con este tipo de requisitos se define el contenido que el sitio Web presenta a sus usuarios. Algunos ejemplos pueden ser: “información del libro” o “categorías del producto”.
- **Requisitos de servicio (*Service*)**. Este tipo de requisito hace referencia a la funcionalidad interna que el sistema debe proveer a los usuarios. Por ejemplo: “registrar un nuevo cliente”, “agregar un producto”, etc.
- **Requisitos de navegación (*Navigational*)**. Un sistema Web debe también definir caminos de navegación disponibles para los usuarios. Algunos ejemplos son: “consultar productos por categoría”, “consultar el carrito de compras”, etc.
- **Requisitos de interfaz (*Layout*)**. Los requisitos también pueden definir la interfaz visual para los usuarios. Por ejemplo: “presentar un estilo diferente para los adolescentes”.
- **Requisitos de personalización (*Personalization*)**. También consideramos requisitos de personalización en nuestra propuesta. El diseñador puede especificar las acciones de personalización a ser ejecutadas en el sitio Web. Por ejemplo: “adaptar el estilo de la fuente para las personas con deficiencia visual”.
- **Requisitos no funcionales (*Softgoals*)**. Estos requisitos representan criterios de calidad que el sistema debe conseguir. Algunos ejemplos de estos requisitos pueden ser: “eficiencia”, “atraer mas usuarios” y “buena experiencia del usuario”.

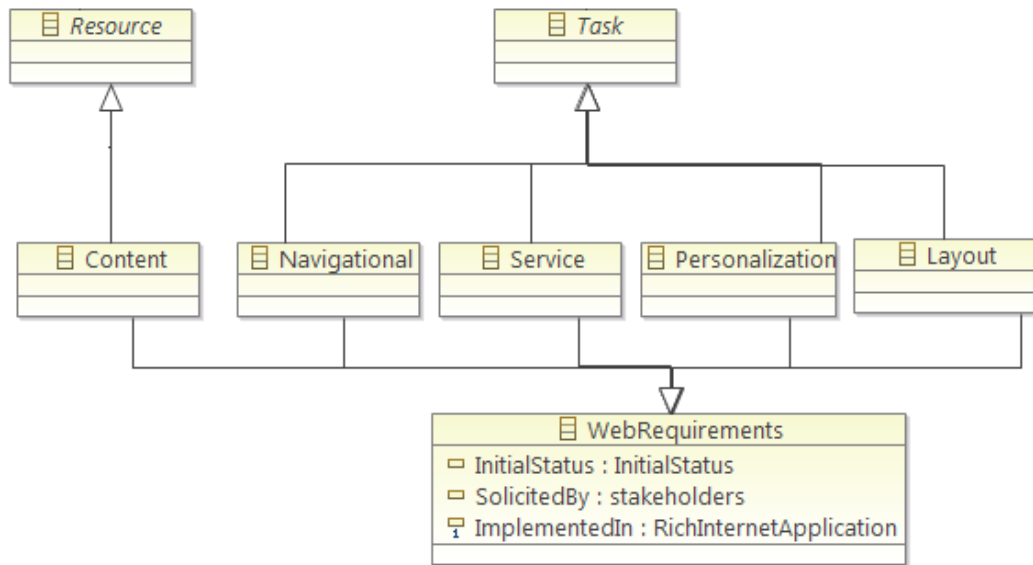
Para poder utilizar el marco de modelado  $i^*$  dentro de MDA se ha implementado un metamodelo. El metamodelo se implementó utilizando la tecnología EMF (*Eclipse Modeling Framework*) de Eclipse [2] (Fig. 7). Los elementos intencionales de  $i^*$  fueron extendidos con nuevas clases para representar cada uno de los tipos de requisitos Web descritos anteriormente (*Navigational*, *Service*, *Personalization*, *Layout* y *Content*). Los requisitos no-funcionales se modelan directamente utilizando elementos de  $i^*$  (*Softgoals*).

A continuación, se resume la generación de los modelos conceptuales de Dominio y Navegación. Se remite al lector a los capítulos específicos de esta tesis doctoral para una explicación más detallada.

## Generación de los Modelos Conceptuales a nivel PIM

Definidos los requisitos en un CIM, el siguiente paso consiste en utilizarlos para derivar los modelos conceptuales de dominio y navegación de la aplicación Web. Para lograrlo, es necesario que los modelos cumplan con la sintaxis abstracta de un dominio específico, es decir, que sean conformes a un metamodelo. A continuación se describen los metamodelos utilizados para la derivación de los modelos conceptuales a nivel PIM de MDA.

UML (*Unified Modeling Language*) es el lenguaje específico de dominio (*Domain Specific Language*, DSL) utilizado para la derivación del modelo de dominio de A-OOH. El metamodelo es una descripción de UML realizada en UML. Describe los objetos, atributos y relaciones



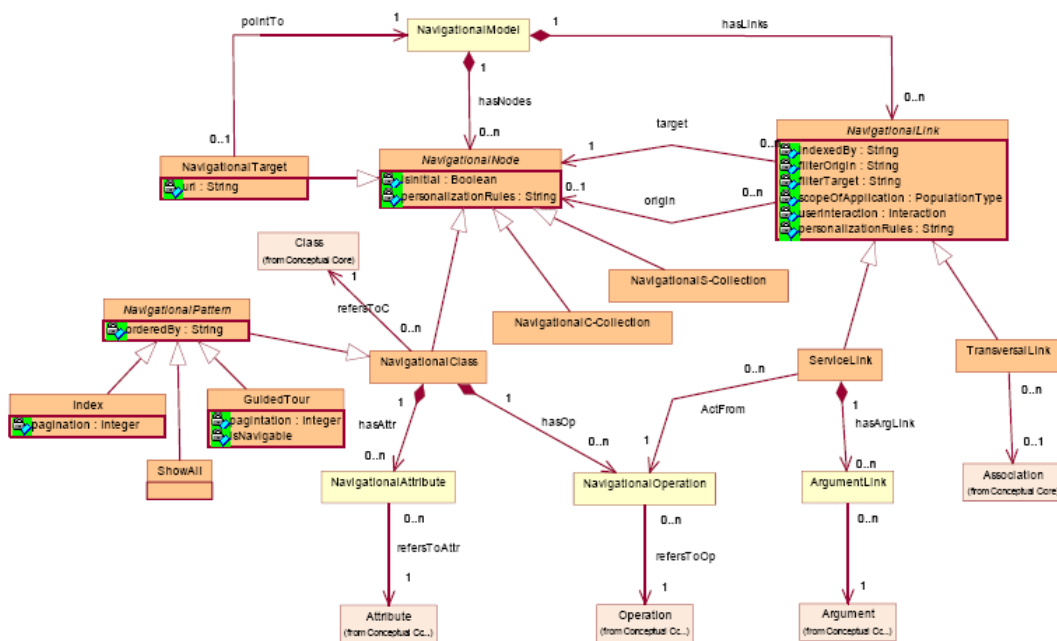
**Fig. 1.2.** Extracto de metamodelo de requisitos Web del método A-OOH.

necesarias para representar los conceptos de UML dentro de una aplicación de software. La notación UML la componen dos subdivisiones, la primera es utilizada para modelar los elementos estáticos del diseño como las clases, atributos y relaciones. La segunda subdivisión es aplicada para modelar los elementos dinámicos de diseño como lo son los objetos, mensajes y máquinas de estado finito. Los modelos estáticos son presentados en diagramas llamados Diagramas de Clases. El propósito de un diagrama de clase es representar a las clases dentro de un modelo. En una aplicación orientada a objetos, las clases tienen atributos (variables miembros), operaciones (funciones miembro) y relaciones con otras clases. Estas características aplican perfectamente para la representación del modelo de dominio de A-OOH.

Por otra parte, A-OOH dispone de un DSL para representar las rutas de navegación que el usuario puede seguir durante su interacción con la aplicación Web. En la Figura 1.3 se muestra el metamodelo de navegación utilizado en A-OOH. Los elementos principales del metamodelo son: (i) Nodo Navegacional (*Navigational Node*) y (ii) Enlace Navegacional (*Navigational Link*).

El Nodo Navegacional representa vistas restringidas de los conceptos del dominio y sus relaciones indican las rutas de navegación que el usuario de la aplicación Web puede seguir. Existen tres tipos diferentes de Nodos Navegacionales, los cuales se describen a continuación:

- **Clases de Navegación** (*Navigational Classes*), son clases del dominio enriquecidas con atributos y métodos cuya visibilidad ha sido restringida, dependiendo claro de los permisos de acceso del usuario y de los requisitos de navegación. Es representada por una clase UML estereotipada *NavigationalClass*.
- **Objetivos de Navegación** (*Navigational Targets*), agrupan los elementos del modelo que colaboran en el cumplimiento de cada requisito de navegación del usuario. Son representados utilizando la notación UML de paquetes con el estereotipo *NavigationalTarget*.
- **Colecciones** (*Collections*), son estructuras jerárquicas definidas en Clases Navegacionales u Objetivos Navegacionales. Proveen al usuario de la aplicación Web nuevas formas de acceder a la información. La colección más común es C-Collection (Clasifier Collection), la cual actúa como un mecanismo de abstracción para el concepto de menú, agrupa de esta forma Ligas Navegacionales (Navigational Links). Otra colección importante es la llamada S-Collection (Selector Collection) mediante la cual podemos representar un mecanismo de selección. Las colecciones son representadas por medio de una clase UML estereotipada como *NavigationalC-Collection* o *NavigationalS-Collection*.



**Fig. 1.3.** Definición del metamodel de navegación en A-OOH.

El Enlace Navegacional define las rutas de navegación que el usuario puede seguir a través de la aplicación Web. A-OOH en su metamodelo de navegación define dos tipos principales de enlaces (links):

- **T-Links** (*Transversal Links*), son enlaces definidos entre dos nodos navegacionales (clases navegacionales, colecciones u objetivos navegacionales). La navegación es realizada para mostrar información a través de la interfaz del usuario sin modificar en absoluto la lógica de negocio. Estos tipos de ligas son representadas por el estereotipo TransversalLink.
- **S-Link** (*Service Links*), con estos tipos de enlaces la navegación es realizada para activar una operación, la cual, en forma opuesta a los T-Links modifica la lógica del negocio y además implica que la navegación a un nodo muestre información cuando termine la ejecución del servicio. Se establece cuando un servicio de la clase navegacional es activado. Estos tipos de ligas son representados por el estereotipo ServiceLink y tiene asociado el nombre del servicio que lo invoco.

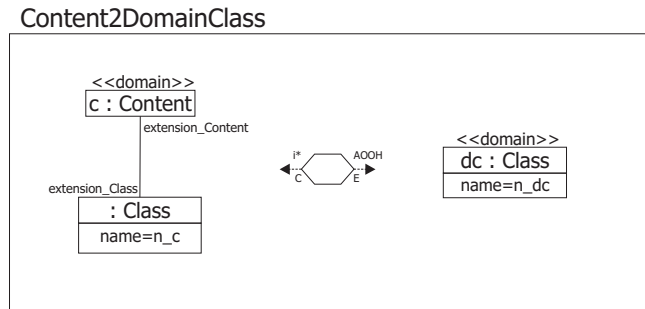
Un modelo representa algún aspecto de un sistema *software* y conforma con el metamodelo del DSL con el que está expresado. Esto significa que los elementos del modelo son instancias de elementos del metamodelo. Una vez se han presentado y descrito los metamodelos utilizados por A-OOH se procederá a explicar la derivación de los modelos conceptuales de dominio y navegación.

El modelo de dominio en A-OOH encapsula la estructura y funcionalidad de los conceptos relevantes del dominio de la aplicación Web y también refleja la parte estática de la misma, este se representa como un diagrama de clases en notación UML. El objetivo consiste en obtener el modelo de dominio a partir del modelo de requisitos en  $i^*$  por medio de un conjunto de reglas de transformación descritas formalmente en QVT. Con las reglas, se obtendrá un modelo de *weaving* para mantener enlaces bidireccionales entre los elementos de ambos modelos (a nivel de sus respectivos elementos). A continuación se describen las principales reglas QVT para generar el modelo de dominio y el modelo de *weaving*:

- **Content2DomainClass.** El dominio origen de esta relación está compuesto por un conjunto de elementos que representan una clase estereotipada como Content. Cuando se detecta este patrón en el modelo de entrada se fuerza la creación de una clase tipo UML-Class

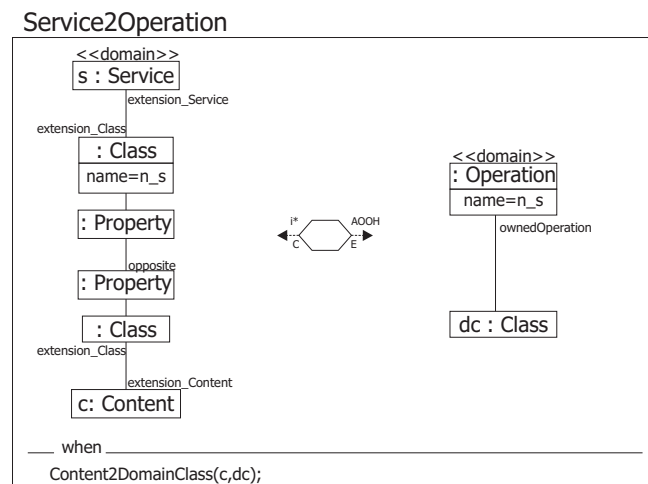


en el modelo destino (modelo de dominio). Por tanto, por cada requisito de contenido se obtiene una clase en el modelo de dominio (Figura 1.4).



**Fig. 1.4.** Regla QVT para obtener las clases del modelo de dominio.

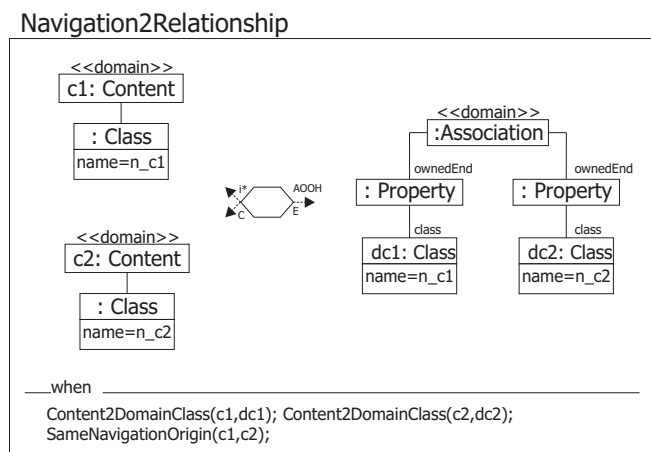
- **Service2Operation.** La regla detecta un conjunto de elementos en el modelo de entrada (modelo de requisitos) que corresponden con una clase estereotipada como Service asociada a una clase estereotipada como Content. Una vez detectado este patrón de elementos, se crea en el modelo de salida (modelo de dominio) una clase Operation en la clase del modelo de dominio correspondiente (Figura 1.5).



**Fig. 1.5.** Regla QVT para obtener las clases del modelo de dominio.

- **Navigation2Relationship.** Esta relación permite crear asociaciones entre clases en el modelo de dominio. Existen dos clases como origen para detectar clases estereotipadas como Content. Si estas dos clases se usan para cumplir el mismo requisito de navegación, entonces se crea una clase Association entre las clases del modelo de dominio que las representan (Figura 1.6).
- **CreateModelTrace.** Al ejecutarse por primera vez, esta regla crea un modelo de *weaving*. Cada vez que una regla QVT transforma un elemento del modelo de entrada (modelo de requisitos) crea un nuevo enlace en el modelo de *weaving*.

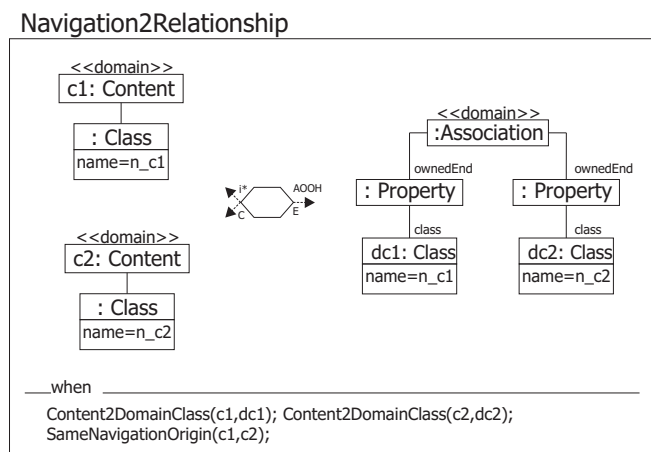
Como se menciono en anteriormente, el modelo de navegación esta compuesto de nodos navegacionales y sus respectivas relaciones para indicar las rutas de navegación de la aplicación Web. Para derivar el modelo de navegación los requisitos tomados en cuenta son los



**Fig. 1.6.** Regla QVT para obtener las relaciones entre las clases del modelo de dominio A-OOH.

requisitos de contenido (*Content*), servicio (*Service*), navegación (*Navigational*) y personalización (*Personalization*) para generar un conjunto de reglas de transformación QVT. En este caso, las reglas QVT para la obtención del modelo de navegación son las siguientes:

- **Navigation2NavClass.** Esta regla de transformación detecta cada requisito navegacional, derivando su correspondiente clase. En concreto, cuando se detecta en el modelo origen una clase estereotipada como Navigational unida a una clase Content, entonces se crea una clase estereotipada como NavigationalClass en el modelo destino. Además cada una de estas nuevas clases es el destino de una nueva asociación TransversalLink desde una C-Collection previamente creada por la función createMenu que se encuentra en la cláusula when (Figura 1.7).



**Fig. 1.7.** Regla QVT para obtener las clases navegacionales.

- **Personalization2NavClass.** La regla es similar a la anterior pero detectando los requisitos de personalización (estereotipo Personalization) que tienen un requisito de contenido asociado. Se derivan en el modelo de navegación los mismos elementos que en la regla anterior.
- **Navigation2TransversalLink.** Permite crear asociaciones entre clases navegacionales en el modelo de navegación. Existen dos modelos origen con el fin de detectar clases estereotipadas como Content. Si estas dos clases se usan para cumplir con el mismo requisito de

navegación (comprobado en la cláusula `when` con la operación `SameNavigationOrigin`), entonces se crea una asociación estereotipada como `TransversalLink` entre las clases del modelo de navegación que las representan.

- ***Service2ServiceAndSLink***. Esta relación detecta un conjunto de elementos en el modelo origen que corresponden con una clase estereotipada como `Service` asociada a una clase estereotipada como `Content`. Una vez detectado este patrón de elementos, se crea en el modelo destino una clase `Operation` en la clase del modelo de navegación correspondiente. Además, se crea una asociación estereotipada como `ServiceLink` para cada operación añadida. Este nuevo enlace de servicio se asocia a una nueva clase navegacional (`NavigationalClass`) destino. Antes de ejecutarse esta relación, se debe verificar que se cumple con las sentencias dispuestas en la cláusula `when`, en este caso `Navigation2NavClass` y `Personalization2NavClass`, con el fin de crear en el modelo destino todas las posibles clases de navegación a partir de cada una de las clases estereotipadas como `Content` en el modelo origen.

### 1.3.2 Gestión de Requisitos en A-OOH

El uso de técnicas orientadas a objetivos para la especificación de requisitos en ingeniería Web permitieron reflejar desde las etapas iniciales del proceso de desarrollo los objetivos, tareas y relaciones de los stakeholders. De esta forma se consideran las necesidades y objetivos del usuario de la aplicación Web. Sin embargo, a pesar de que los requisitos son especificados en un modelo conceptual, los stakeholders necesitan observar que han sido reflejados correctamente en la aplicación Web final. Una forma de brindar soporte a esta necesidad es proveer al diseñador Web con una etapa de requisitos que considere la gestión de los mismos.

Por tanto, la propuesta presentada en la tesis se ha extendido para proveer al diseñador con soporte para: (i) trazabilidad de requisitos (CIM-PIM), (ii) evaluar el impacto derivado de un cambio en los modelos conceptuales (análisis de impacto) y (iii) alternativas de diseño considerando la maximización y/o balance de los requisitos no-funcionales.

### Trazabilidad de Requisitos en A-OOH

En el campo de la ingeniería Web dirigida por modelos, realizar el seguimiento de los requisitos durante la etapa de desarrollo de la aplicación Web hasta su implementación final es una tarea compleja. Esto se debe a que en ingeniería Web se deben generar varios modelos conceptuales a partir de los requisitos como son el modelo de dominio o de navegación. Además, debido al desarrollo gradual de las necesidades de los usuarios de una aplicación Web, estos modelos cambian constantemente por lo que la trazabilidad de los requisitos se hace indispensable.

La trazabilidad de requisitos se define como la capacidad de describir y seguir la vida de un requisito, en ambas direcciones [27]: (i) determinar qué partes del modelo están relacionadas con cada uno de los requisitos, y (ii) determinar qué requisitos dieron origen a qué partes del modelo. Actualmente existen dos estrategias para gestionar y almacenar la información para la trazabilidad entre modelos: (i) la información se puede integrar en los modelos a los que se refiere y (ii) la información de trazabilidad se puede almacenar por separado en otro modelo [16]. La primera de estas dos opciones tiene como desventaja que si la información es almacenada en el mismo modelo, el modelo será contaminado con información poco relevante para el contexto del modelo y por lo tanto, será difícil de mantener y utilizar. Por otro lado, la segunda estrategia consiste en almacenar la información en un modelo aparte. De esta forma se pueden corregir las desventajas mencionadas.

Con el fin de brindar soporte para la trazabilidad de requisitos en A-OOH se ha utilizado el concepto de modelo de *weaving*. Un modelo de *weaving* es un tipo de modelo formado por enlaces y referencias a elementos, los enlaces están dirigidos a las referencias de los elementos de un modelo origen y de un modelo destino. A continuación, se presenta el metamodelo base para *weaving* [15] y una extensión para proveer a dicho metamodelo con elementos útiles para representar la trazabilidad entre modelos [6]. El metamodelo se muestra en la Figura 1.8:

- **WElement**. Es el elemento base del cual los seis elementos restantes heredan, esta formado por los atributos nombre y descripción.
- **WModel**. Representa el elemento raíz que contiene a todos los elementos del modelo. Está compuesto por las referencias y relaciones a los modelos de entrada y salida.
- **WLink**. Sirve para representar un enlace entre los elementos del modelo.
- **WLinkEnd**. Este elemento representa el extremo final de un enlace.
- **WElementRef**. Este elemento se asocia a una función de identificación, creando un identificador único para cada elemento del modelo ligado, por tanto WElementRef permite referenciar el mismo elemento del modelo enlazados por diversos elementos WLinkEnd.
- **WModelRef**. Representa un identificador único de un modelo.

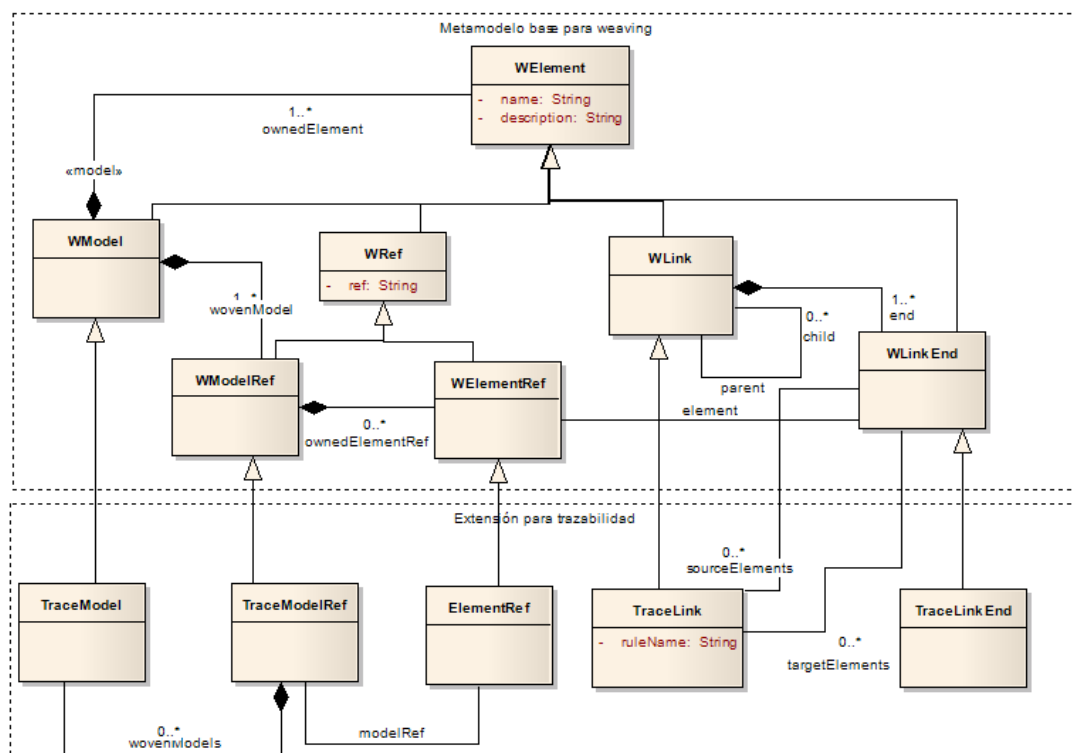


Fig. 1.8. Metamodelo de *weaving* y extensión para trazabilidad.

En la parte inferior de la Figura 1.8, se ilustra la extensión del metamodelo para *weaving* que permite la representación de la trazabilidad. Esta extensión la forman los siguientes elementos:

- **TraceModel**. Es el elemento que representa al modelo de trazabilidad, esta integrado por referencias a otros modelos.
- **TraceModelRef**. Representa la referencia a otros modelos, es decir, es un único identificador para los modelos que conforman el modelo de trazabilidad.
- **ElementRef**. Es un identificador para señalar cada elemento que integran los modelos ligados.
- **TraceLink**. Un enlace de rastreo, utilizado para representar las correspondencias entre las referencias de los elementos de los modelos enlazados. Como información de trazabilidad, almacena el nombre de la regla de transformación que ha sido ejecutada.
- **TraceLinkEnd**. Su función es similar al elemento WLinkEnd del cual hereda, pues permite crear una relación uno a muchos (1-N) entre las referencias de los elementos del modelo de entrada (sourceElements) y los del modelo de salida (targetElement).

De esta forma se ha definido formalmente la regla de transformación *CreateModelTrace* para derivar un modelo de *weaving* que almacene un conjunto de enlaces dirigidos a los elementos de los modelos conceptuales de A-OOH.

- ***CreateModelTrace***. Al ejecutarse por primera vez, la regla crea un modelo de *weaving* con referencias a los modelos origen y destino. De esta forma, cada vez que una regla QVT transforma un elemento del modelo de entrada (modelo de requisitos) en uno o varios elementos de los modelos conceptuales A-OOH (dominio y navegación) crea un nuevo enlace en el modelo de *weaving*.

## Content2DomainClass

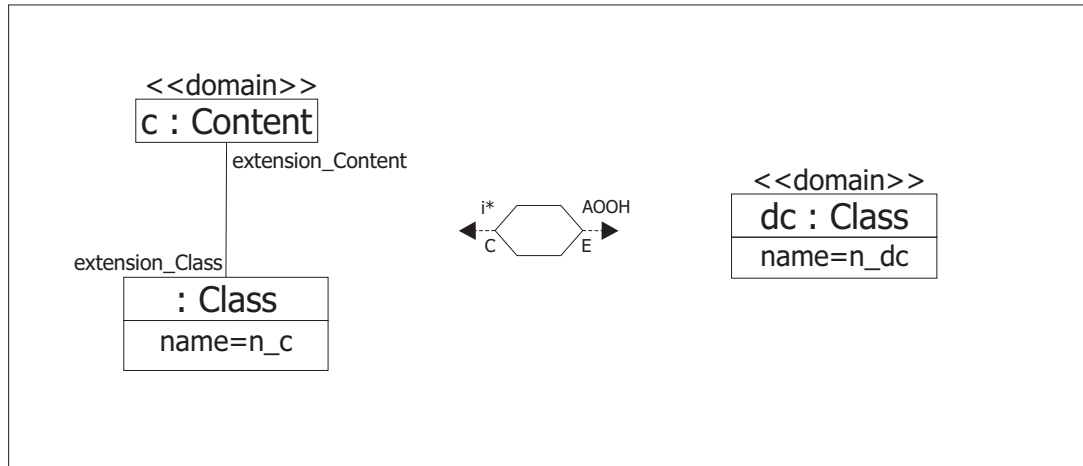


Fig. 1.9. Regla QVT para obtener el modelo de *weaving*.

## Análisis de Impacto en A-OOH

Los requisitos evolucionan constantemente a razón de la naturaleza dinámica de la Web. Debido a esto, es común encontrar inconsistencias entre los modelos conceptuales de la aplicación Web y los requisitos. Una de las ventajas ofrecidas por el soporte para trazabilidad en A-OOH es el de conocer las dependencias entre los elementos de los modelos conceptuales y los requisitos. Por tanto, es posible conocer los requisitos afectados debido a un cambio en alguno de los modelos conceptuales. Análisis de impacto (*Change Impact Analysis*) es la tarea de identificar las consecuencias potenciales de un cambio o estimar qué es necesario modificar para llevar a cabo el cambio [1]. Comúnmente, el análisis de impacto se ha realizado de forma intuitiva por los diseñadores de la aplicación Web por medio de un análisis superficial del código y documentación de la aplicación. Esto quizá sea suficiente para aplicaciones Web no muy grandes, pero no es suficiente para aplicaciones Web sofisticadas. Asimismo, investigación empírica demuestra que, incluso los desarrolladores más experimentados deducen un análisis de impacto incompleto [11].

Para esto, se ha definido un algoritmo para conocer analizar las dependencias entre los requisitos funcionales. El algoritmo analiza el modelo de requisitos A-OOH para conocer las dependencias entre los requisitos funcionales además de saber qué requisitos no funcionales se ven afectados. De esta forma, a través del soporte para trazabilidad es posible conocer que otros elementos de los modelos conceptuales son afectados. Finalmente, debido a que el modelo de requisitos A-OOH es orientado a objetivos, el algoritmo puede mostrar al diseñador un camino alternativo en el cual se indique qué requisitos tienen que ser implementados para seguir cumpliendo con el objetivo (*Goal*).

## Alternativas de Diseño en A-OOH considerando a los Requisitos No-Funcionales

### 1.3.3 Implementación

La propuesta definida en esta tesis doctoral se ha implementado en la plataforma de desarrollo *Eclipse* [2] utilizando tecnología EMF (*Eclipse Modeling Framework*) (CITA) y GMF (*Graphical Modeling Framework*) (CITA). *Eclipse* puede extenderse por medio de *plugins* con el fin de añadir más características y nuevas funcionalidades. Se ha desarrollado un *plugin* que da soporte a cada parte de la propuesta. Este nuevo plugin contiene los siguientes módulos:

#### Módulo *Computational Independent Model*

Este modulo implementa los conceptos del marco de modelado  $i^*$  y la clasificación de requisitos presentada en [4] para proveer al diseñador con un editor gráfico para la especificación de requisitos Web. Concretamente, el metamodelo de requisitos implementado en la tesis doctoral ha sido extendido para incorporar los tipos de requisitos de la taxonomía presentada en la sección 1.3.1, la Figura 1.10 muestra una captura de pantalla donde puede apreciarse la implementación del metamodelo en Eclipse.

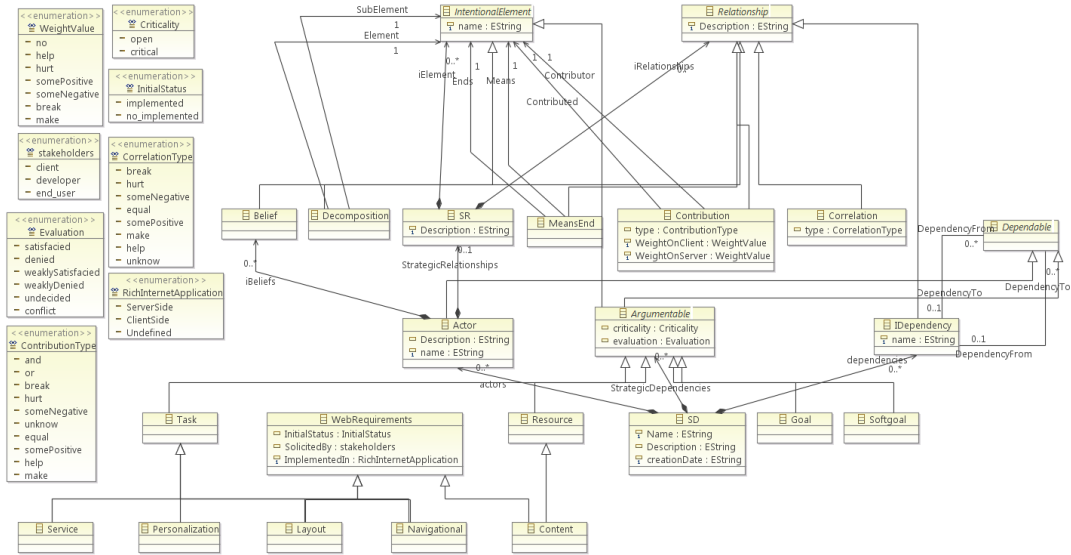


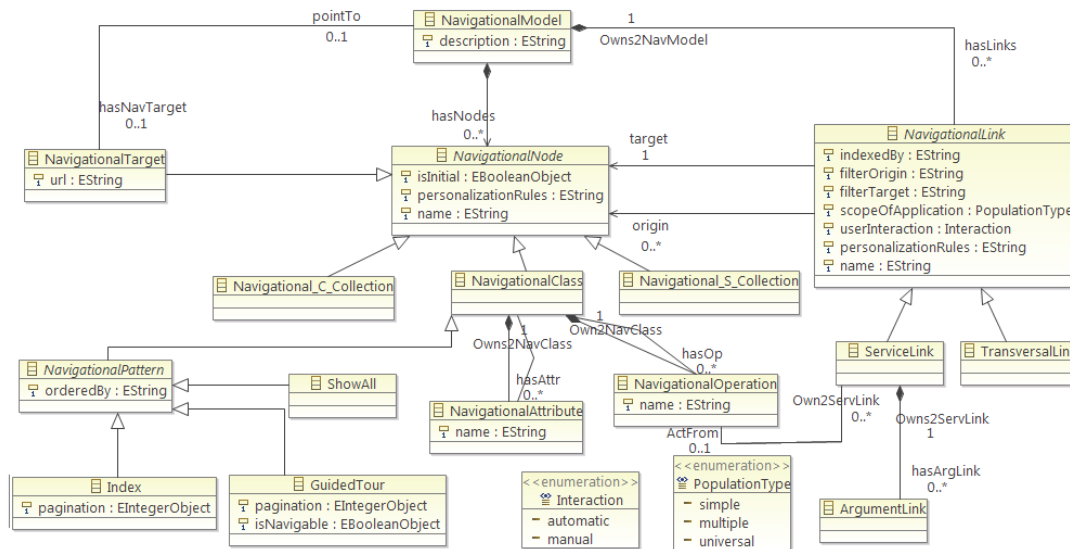
Fig. 1.10. Metamodelo para la especificación de requisitos Web con  $i^*$ .

Por otra parte, con el editor, el diseñador de la aplicación Web es capaz de especificar en un diagrama los requisitos funcionales y no-funcionales de la aplicación Web. A continuación se describen las características principales del editor gráfico:

- El editor implementa el metamodelo de requisitos Web  $i^*$  de tal forma que, los modelos especificados utilizando el editor son conformes al metamodelo.
- El editor gráfico permite la creación, modificación y actualización de la especificación de requisitos Web (diagramas).
- Las propiedades de los elementos del marco de modelado  $i^*$  pueden ser modificadas seleccionando cada elemento en la vista de propiedades.
- Además de poder especificar requisitos Web (*Content*, *Navigational*, *Personalization*, *Service* y *Layout*), el editor permite la creación de diagramas (modelos) con los elementos clásicos (*Goal*, *Task*, *Resource* y *Softgoal* de  $i^*$ ).

Módulo *Platform Independent Model*

El metamodelo de navegación en A-OOH se implementa en este módulo. El metamodelo definido formalmente en la Figura 1.3 e implementado en la Figura X define la sintaxis para la representación conceptual en un PIM del modelo de navegación de A-OOH.



**Fig. 1.11.** Metamodelo de Navegación de A-OOH.

## Módulo Transformaciones

Tras probar varios motores de transformaciones para implementar las reglas QVT definidas (como *mediniQVT* o *smartQVT*), finalmente se eligió el motor del *ATLAS Transformation Language* (ATL) [10]. Por tanto, este módulo aprovecha este motor para implementar y ejecutar todas las transformaciones definidas como parte de la propuesta guiada por modelos.

## Caso de estudio

A continuación, con el fin de ejemplificar el uso de esta herramienta, se presenta un caso de estudio basado en el Plan Estratégico de Formación de la Universidad de Alicante (<http://www.ua.es/es/presentacion/pe/psec/formacion/index.html>). Este plan determina los ejes, los objetivos y las acciones necesarias para articular una oferta formativa de calidad. En concreto, tras el análisis de este plan, el caso de estudio se centra en el desarrollo de un AD que apoye la toma de decisiones en el proceso de evaluación (*“Assessment”*) en la Universidad de Alicante. En este proceso está implicado un actor principal, el *“education manager”*, mediante el objetivo estratégico *“provide a good education program”*. A partir de este objetivo se obtienen tres objetivos decisionales diferentes: *“adapt education program to demand”*, *“achieve international recognition”* y *“having renowned program”*. Los objetivos informacionales derivados de estos objetivos decisionales son: *“evaluate environment demand”*, *“analyze international impact”* y *“study student performance”*. A partir de estos objetivos informacionales se pueden obtener los requisitos de información como tareas: *“percentage of students per city and province”*, *“percentage of foreign students”*, *“average of passed examination sessions by subject, degree and department”*. Una vez hecho esto, se debe determinar las medidas y los contextos de análisis y asociarlos a los requisitos de información como recursos: la única medida es *“examination session”* y los elementos que representan contextos de análisis son *“student”*, *“city”*,

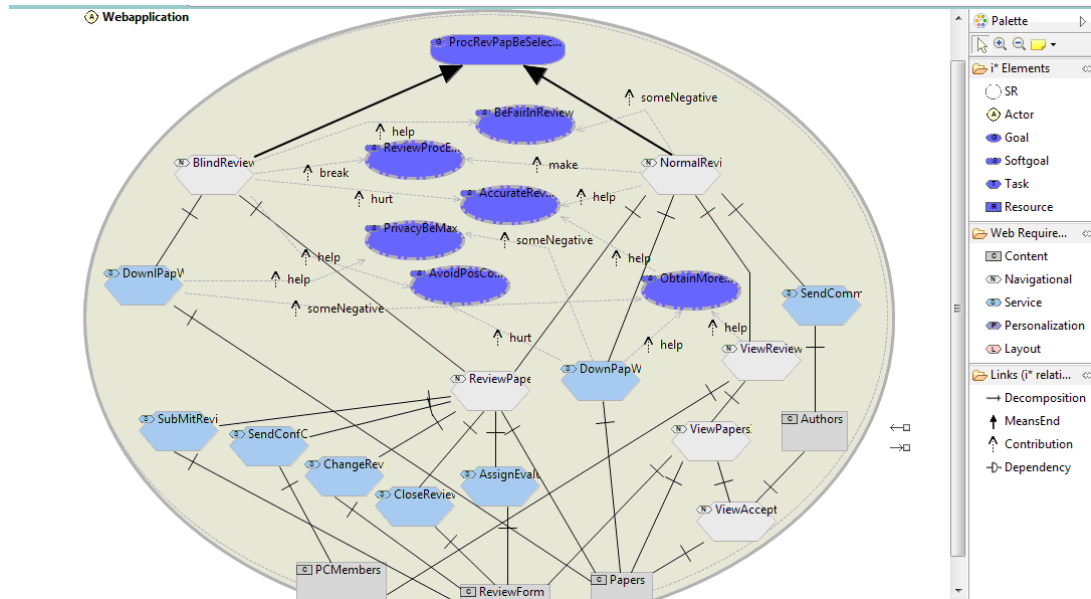


Fig. 1.12. Editor gráfico para la especificación de requisitos Web con  $i^*$ .

“province”, “country”, “subject”, “degree” y “department”. Parte de estos contextos de análisis (“student” y “subject”) pueden agregarse, por lo que se relacionan con otros contextos.

Cada uno de estos elementos se define en un CIM mediante el *profile* UML de  $i^*$  (Fig. ??).

De manera resumida, para definir apropiadamente un CIM con  $i^*$  se deben realizar los siguientes pasos: (i) descubrir los actores (usuarios del AD), (ii) descubrir sus objetivos (estratégicos, decisionales e informacionales), (iii) derivar requisitos de información de los objetivos informacionales y (iv) obtener las medidas y el contexto de análisis relacionados con los requisitos de información.

Con el fin de automatizar el paso del CIM al PIM, se ha desarrollado una serie de reglas de transformación QVT [?]. Esta transformación tiene como entrada el CIM y crea como salida un PIM con los elementos MDs correspondientes (tal y como se muestra en la Fig. ??): se crea una clase *Fact Assessment* con un *FactAttribute* llamado *ExaminationSession* (Fig. ??) y, además, se crean dos clases *Dimension* y sus jerarquías de clases *Base* según los contextos de análisis definidos en el CIM (ver Fig. ?? y Fig. ??).

El próximo paso es la obtención de un modelo de fuentes de datos y su marcado con conceptos MDs (tales como hecho, dimensión, etc.). En este caso de estudio existe una implementación de las fuentes de datos en *Oracle*. El proceso de derivación de un modelo relacional a partir del diccionario de datos de *Oracle* se ha implementado mediante *Java* dentro del entorno *Eclipse*. En concreto se ha usado la interfaz `java.sql.Connection` para realizar la conexión a la base de datos *Oracle* y ejecutar las sentencias SQL requeridas para obtener los metadatos del diccionario. Después de obtener los metadatos necesarios, se deriva el modelo correspondiente mediante el uso de *Eclipse* y el metamodelo relacional de CWM. Una vez que se tiene este modelo, se marca cada uno de sus elementos con conceptos MDs. La figura ?? muestra el modelo de las fuentes de datos del caso de estudio.

Una vez que se tiene el modelo de las fuentes y el PIM inicial, se debe proceder a realizar su reconciliación. Esta reconciliación se ha implementado en tres pasos siguiendo las formas normales multidimensionales. Primero, con el fin de asegurar que el PIM es fidedigno, para cada una de sus dependencias funcionales (FD) se comprueba que existe una FD equivalente en el modelo de las fuentes de datos, es decir, las FDs del modelo MD deben ser un subconjunto de aquellas observadas en las fuentes de datos. La opción “*Required Annotation*” (ver Fig. ??) ejecuta una serie de reglas para hallar las FDs del PIM y comprobar que esas mismas FDs ocurren en el modelo de las fuentes de datos. Si esta comprobación falla, entonces el estado



(*status*) de los elementos involucrados se etiqueta como *required* y se colorean en *rojo* para indicar que estos elementos aparecen en el PIM inicial pero no tienen equivalente en el modelo de fuentes de datos (es decir, son requeridos por el usuario pero las fuentes de datos no los suministran). Por ejemplo, en la Fig. ??, *ExaminationSession*, *Country* y la asociación *Rolls-upTo* entre *Degree* y *Department* están en color rojo porque no existen elementos equivalentes en el modelo de fuentes de datos.

Una vez se comprueba que el modelo es fidedigno mediante la anotación de elementos requeridos, el segundo paso es asegurar la completitud del modelo. Se debe realizar una comprobación de las siguientes condiciones:

- Las FDs entre niveles de dimensión que aparezcan en las fuentes de datos deben representarse como asociaciones *Rolls-upTo* en el modelo MD. De lo contrario, el potencial de análisis se pierde (*completitud de agregación*).
- Las FDs entre conjuntos de medidas que se encuentran en las fuentes de datos deben representarse mediante fórmulas de derivación. De otra manera, se perderían estas relaciones en el modelo MD (*completitud de derivación*).
- Cada medida (*FactAttribute*) se debe asignar a un hecho (clase *Fact*) de tal manera que los niveles terminales de dimensión (clase *Dimension*) determinen funcionalmente y sin dependencias transitivas a la medida. De otro modo, la medida se guarda de manera redundante en un nivel de detalle “erróneo” (*eliminación de redundancias*).

Estas condiciones se comprueban mediante la ejecución de la transformación “*Supplied Annotation*” (ver Fig. ??), la cual comprueba que las FDs del modelo de fuentes de datos tienen su equivalente en el PIM. Cuando esta comprobación falla, el estado (*status*) de los elementos involucrados se etiqueta como *supplied* y se colorean de *azul* para indicar que estos elementos aparecen en las fuentes de datos pero no en el PIM inicial (es decir, son suministrados por las fuentes de datos pero el usuario no los ha tenido en cuenta, bien porque no los necesitaba o bien porque no sabía de su existencia). Por ejemplo, en la Fig. ?? y en la Fig. ??, se muestran nuevos elementos en azul que forman parte de las fuentes de datos pero no aparecen en el PIM inicial.

Hasta ahora cada elemento del PIM inicial se ha etiquetado con cierto estado (*status*) como *required* o *supplied*, o bien se ha dejado sin etiquetar (estado *none*). La tarea del diseñador es comprobar el modelo resultante y cambiar el estado de los elementos a *none* (ver Fig. ??) si (i) un elemento requerido (*required*) puede ser suministrado por alguna fuente de datos externa o (ii) un elemento suministrado (*supplied*) puede ser útil para el usuario. Obviamente, en el siguiente paso, sólo se tienen en cuenta los elementos cuyo estado es *none* para poder derivar el PIM híbrido (ver Fig. ?? y Fig. ??).

Una vez que se obtiene el PIM híbrido, el siguiente paso es obtener un PSM acorde a una tecnología específica. En este ejemplo, se deriva un PSM según la representación lógica más común de un modelo MD: el *esquema estrella* [?]. Éste es un esquema relacional que consiste en una tabla de hechos central con una clave primaria compuesta cuyos elementos forman una clave ajena a cada una de las tablas de dimensión (las cuales poseen una clave primaria única). Este esquema se muestra en la Fig. ??.

Finalmente, se obtiene el código que implementa el modelo MD en una herramienta comercial. El código SQL para el esquema estrella se obtiene del PSM (Fig. ??), mientras que del PIM híbrido se obtiene directamente el código que se usará en una herramienta de análisis de datos (Fig. ??).

## 1.4 Conclusiones

Un AD es una colección integrada de datos históricos en apoyo a la toma de decisiones. Según esta definición, en el desarrollo de un modelo MD para un AD no es sólo importante considerar las necesidades de información de los usuarios (propuestas guiadas por requisitos), sino también las fuentes de datos existentes que poblarán el AD (propuestas guiadas por datos). Por tanto, se requiere de mecanismos formales para integrar estos dos puntos de vista en una propuesta híbrida. Además, el modelado MD del AD se asemeja a los métodos de diseño de bases de datos tradicionales [?] en cuanto a que debe estructurarse en varios pasos durante los cuales se desarrolla una fase de diseño conceptual, cuyos resultados se transforman en un modelo de datos lógico como base de la implementación del esquema. Esta manera de proceder posibilita la automatización de las transformaciones entre estas fases.

Para tratar con estas cuestiones, en esta tesis doctoral, se ha presentado una propuesta dirigida por modelos que permite (i) la especificación de un modelo MD híbrido a nivel conceptual de manera integral, sistemática y bien estructurada y (ii) la derivación automática de su representación lógica. Por lo tanto, esta propuesta permite a los diseñadores decrementar la complejidad del desarrollo de un AD, ahorrando tiempo y esfuerzo. Posteriormente, se ha añadido a esta propuesta dirigida por modelos un proceso de normalización con el fin de asegurar la sumarizabilidad de las estructuras MD complejas, como las jerarquías de dimensión y las relaciones hecho-dimensión. También, se ha desarrollado una herramienta basada en *Eclipse* que apoya cada parte de esta propuesta.

Finalmente, cabe destacar que esta tesis doctoral representa la primera propuesta de un proceso híbrido para el desarrollo del AD teniendo en cuenta requisitos de información y fuentes de datos, a la vez que se evitan los problemas de sumarizabilidad.

## References

1. R. Arnold and S. Böhner. Impact analysis-towards a framework for comparison. In *Software Maintenance ,1993. CSM-93, Proceedings., Conference on*, pages 292–301, sep 1993.
2. Eclipse. <http://www.eclipse.org/>, 2010.
3. M. Escalona, M. Mejías, and J. Torres. Developing systems with NDT & NDT-Tool. In *13th International conference on information systems development: methods and tools, theory and practice, Vilna, Lithuania*, pages 149–59, 2004.
4. M. J. Escalona and N. Koch. Requirements engineering for web applications - a comparative study. *J. Web Eng.*, 2(3):193–212, 2004.
5. I. Garrigós, J.-N. Mazón, and J. Trujillo. A requirement analysis approach for using i\* in web engineering. In *ICWE*, pages 151–165, 2009.
6. J. Gómez, C. Cachero, and O. Pastor. Extending a conceptual modelling approach to web application design. In *CAiSE '00: Proceedings of the 12th International Conference on Advanced Information Systems Engineering*, pages 79–93, London, UK, 2000. Springer-Verlag.
7. N. Koch. The expressive power of uml-based web engineering. In *International Workshop on Web-oriented Software Technology (IWWOOST)*, pages 40–41, 2002.
8. N. Koch, A. Knapp, G. Zhang, and H. Baumeister. Uml-based web engineering. In *Web Engineering: Modelling and Implementing Web Applications*, Human-Computer Interaction Series, pages 157–191. Springer London, 2008.
9. N. Koch, G. Zhang, and M. J. E. Cuaresma. Model transformations from requirements to web system design. In *ICWE*, pages 281–288, 2006.
10. A. T. Language. <http://www.eclipse.org/m2m/at1/>.
11. M. Lindvall and K. Sandahl. How well do experienced software developers predict software change? *J. Syst. Softw.*, 43:19–27, October 1998.
12. Model Driven Architecture. <http://www.omg.org/mda/>.
13. Object Management Group. <http://www.omg.org>.
14. O. Pastor, S. M. Abrahão, and J. Fons. An object-oriented approach to automate web applications development. In *EC-Web 2001: Proceedings of the Second International Conference on Electronic Commerce and Web Technologies*, pages 16–28, London, UK, 2001. Springer-Verlag.

15. R. Quintero, V. Pelechano, O. Pastor, and J. Fons. Aplicación de MDA al Desarrollo de Aplicaciones Web en OOWS. *Jornadas de Ingeniería de Software y Base de Datos (JISBD)*, VIII, pages 84–668, 2003.
16. Unified Modeling Language. <http://www.uml.org>.
17. E. Yu. *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, University of Toronto, Canada, 1995.

