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".

Tupac Shakur.

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.

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).

Alicante, July 2011 *José Alfonso Aguilar Calderón*

Contents

Part I

Summary

1

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 II presenta el conjunto de artículos publicados que forman el núcleo de la tesis. Finalmente, la Parte III consiste en un apéndice donde se presentan dos trabajos que se encuentran actualmente en proceso de revisión.

Asímismo, 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 Estrátregicas.

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 2

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 3

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 capítuo anterior, se realizó una revisión sistemática de la literatura para estudiar las técnicas ingenieríles 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 4

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 carácteristicas 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 análizar 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 5

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 A

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 2.

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 3 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: XXXXXXX.*

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: XXXXXXX.*

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 carácteristicas 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 carácteristicas 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 carácteristicas 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 especifico 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 estratégicas 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 tpo de fuente utilizada en el texto, el color, etc.
- **Modelo de personalización**. Este modelo es utilizado para la especificación de estratégias 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 3 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 *stakeholder0's* [1] (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 A)

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 *Meands-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**.

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

Elemento	Descripción
Actor ()	Es una entidad que lleva a cabo acciones para cumplir con sus objetivos. Se relaciona con varios elementos intencionales (objetivo, tarea o recurso).
Goal ()	Representa una condición o estado que el actor le gustaría alcanzar.
Task ()	Representa una manera particular de hacer algo.
Resource ()	Es una entidad que debe estar disponible para su uso.
Means-ends ()	Son asociaciones que describen cómo se alcanzan los objetivos, es decir, los posibles caminos para satisfacer un objetivo.
Decomposition ()	Son asociaciones que definen elementos adicionales necesarios para llevar a cabo una tarea.

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:

[1] ....

- **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*).

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

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 especifico, 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 especifico 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 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 maquinas 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 carácteristicas 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*).

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

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 del 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 accesar 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.

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).

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

- ***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 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 Personaliza tion2NavClass, 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 permité reflejar desde las etapas iniciales del proceso de desarrollo los objetivos, tareas y relaciones de los stakeholder's. 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 consideré 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*.

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 suficientes para aplicaciones Web sofisticadas. Asímismo, investigación empirica demuestra que, incluso los desarrolladores más experimentados deducen un análisis de impacto incompleto [11].

Para paliar esta limitante, se ha definido un algoritmo para 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 cumpliento con el objetivo (*Goal*). A continuación se presenta un ejemplo de la aplicación del algoritmo.

El cliente ha solicitado la implementación de una aplicación Web que permita la aplicación de encuestas *on-line*. En primer lugar es necesario que el diseñador especifique los requisitos de la aplicación Web, para esta demostración, solo se ha definido el actor que representa a la aplicación Web (Figura 1.10).

Fig. 1.10. El modelo de requisitos de la aplicación *"Survey WebApp"*.

El actor "*Survey WebApp*" tiene que cumplir con el objetivo "*Survey be performed*", para lograrlo dispone de dos vías, por medio de la aplicación de la encuesta pública "*Public survey*" y la privada "*Private survey*", ambos requisitos de navegación. Cada una de las vías necesita de uno o más requisitos para cumplirse, tal es el caso del requisito "*Private survey*" debido a que necesita de los requisitos de navegación "*Make private survey*" y "*Viewsurvey*" así como del requisito de contenido "*Registered users*". Algunos de los requisitos afectan positiva o negativamente a los requisitos no-funcionales especificados, por ejemplo, el requisito navegacional "*Private survey*" afecta positivame al requisito no-funcional "*Security*". El tipo de contribuciones que realizan los requisitos funcionales a los requisitos no-funcionales es relevante para la ejecución del algoritmo ya que son utilizadas para decidir qué requisitos funcionales será necesario implementar. En este ejemplo, solo los requisitos relacionados con el requisito de navegación "*Public survey*" se encuentran implementados en los modelos conceptuales.

Ahora bien, supongamos el siguiente escenario: el desarrollador Web elimina un elemento del modelo de dominio A-OOH, gracias al soporte para trazabilidad de A-OOH es posible conocer el requisito afectado por la clase del modelo de dominio que ha sido eliminada, la clase corresponde con el requisito navegacional "*Public survey*". Por tanto, el objetivo "*Survey be performed*" no se puede cumplir. Por tal motivo, es necesario conocer:

- ¿Cuáles requisitos son afectados por este cambio en el modelo de requisitos?
- ¿Cuáles elementos de los modelos conceptuales son afectados?

Por último, es necesario encontrar un camino alternativo en el modelo de requisitos (si es que existe) para continuar cumpliendo con el objetivo "*Survey be performed*".

El algoritmo inicia una vez que se han cumplido un conjunto de pre-condiciones, las cuales pueden consultarse con detalle en el Capítulo 4. El primer paso del algoritmo consiste en realizar un listado de todos aquellos requisitos funcionales que se encuentren implementados o no (en este caso, reflejados en el modelo de dominio), y que además realicen alguna contribución positiva o negativa a cualquier requisito no-funcional. De tal forma que el listado quedará como lo muestra la Tabla 1.2. El requisito afectado se muestra resaltado en negritas. Con el listado generado y el soporte para trazabilidad de A-OOH es posible saber qué elementos del modelo conceptual de dominio resultan afectados por la eliminación del requisito navegacional "*Public survey*".

Table 1.2. La contribución de los requisitos funcionales a cada requisito no-funcional.

Requirements	*"Process easier"*	*"Complete info"*	*"Security"*
"Public Survey"	**Help**	**-**	**Hurt**
"Make Survey"	-	Break	-
"Private Survey"	Some-	-	Some+
"Send Survey"	-	-	Some-
"View Surveys"	-	Make	-
"Send Private Survey"	-	-	Help

El siguiente paso consiste en determinar un camino alternativo para la satisfacción del objetivo de la aplicación ("*Survey be performed*"). Por cada requisito no-funcional que recibe una contribución por parte del requisito a remover (Tabla 1.2) es necesario buscar un requisito no implementado del cual su contribución compense la eliminación del requisito a remover. En este caso el requisito navegacional *"Private Survey"* realiza dos contribuciones a los requisitos no-funcionales *"Process easier"* y *"Security"*, por lo tanto se puede implementar. Para poder determinar si el requisito se puede implementar o no, es necesario aplicar un conjunto de heurísticas (Capítulo 4). El requisito navegacional *"Private Survey"* se puede implementar gracias a las heurísticas número 2 y 3. Este paso es iterativo y finaliza cuando no hay más requisitos (no implementados) que realicen contribuciones a los requisitos no-funcionales. En este caso finaliza cuando es detectado el requisito navegacional *"Send Private Survey"*.

Finalmente, es necesario aplicar una post-condición, la cual establece que si los requisitos a implementar (*"Private Survey"* y *"Send Private Survey"*) tienen uno o más requisitos asociados, estos requisitos deben ser implementadas de forma automática. Por tanto, los requisitos *"View Surveys"* y *"Registered Users"* deben de ser implementados. En la Figura 1.11 se muestran los requisitos a implementar.

Fig. 1.11. Los nuevos requisitos a implementar en el modelo de dominio.

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

Como se ha motivado anteriormente, las aplicaciones Web tienen una audiencia amplia y heterogénea, por eso es indispensable que las metodologías de diseño brinden soporte al diseñador en la etapa de requisitos y le permitan seleccionar distintas opciones de diseño en base la audiencia de la aplicación Web.

En A-OOH se ha implementado el algoritmo Frontera de Pareto para proveer al diseñador con un conjunto de alternativas de diseño basadas en la prioridad de los requisitos no-funcionales. Para esto, el diseñador establecerá una lista de requisitos no-funcionales, a continuación, el algoritmo obtendrá todas las posibles configuraciones de entre las cuales el diseñador seleccionará aquella que maximice o balancee los requisitos no-funcionales establecidos previamente.

1.4 Web 2.0: *Rich Internet Applications*

1.5 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 1.12 muestra una captura de pantalla donde puede apreciarse la implementación del metamodelo en Eclipse.

Fig. 1.12. 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 diagrma los requisitos funcionales y no-funcionales de la aplicación Web. A continuación se describen las carácteristicas 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. 1.13. Metamodelo de Navegación de A-OOH.

Módulo Transformaciones

Este modulo tiene como objetivo implementar todas las transformaciones definidas formalmente en QVT. La opción elegida para la implementación de las transformaciones es el *ATLAS Transformation Language* (ATL) [10]. ATL es un lenguaje para transformaciones entre modelos que agiliza el proceso de implementación de las reglas especificadas formalmente con el lenguaje QVT. Utilizar ATL para implementar transformaciones QVT es factible debido a que ATL y QVT están alineados [19].

Caso de estudio

En este apartado se presenta un caso de estudio para demostrar la aplicabilidad de nuestra propuesta. El caso de estudio es acerca de una empresa que se dedica a la aplicación de encuestas *on-line*. Para esto la empresa requiere que su aplicación Web permita la aplicación de encuestas para usuarios registrados y para el público en general (quien tendrá ciertas restricciones). Para efectos demostrativos, en este caso de estudio nos enfocaremos únicamente a la obtención del modelo de dominio a partir de la especificación de requisitos. El primer paso consiste en la especificación de los requisitos para la aplicación Web "Survey WebApp". En la figura 1, por cuestiones de espacio, solo se presenta un extracto de la especificación de requisitos en el contexto del método Web A-OOH. El actor "Survey WebApp" tiene como objetivo "Survey be performed", para alcanzar este objetivo es necesario seleccionar el requisito Navigational "Public Survey" o "Private Survey", cada uno de estos requisitos necesita de otros para poder realizarse, por ejemplo, el requisito "Public Survey" necesita del requisito Navigational "Make Survey" y este a su vez, necesita del requisito Service "Send Survey" y, finalmente, el requisito "Send Survey" necesita del requisito Content "Survey Repository" para poder llevarse a cabo. Finalmente, es importante resaltar que cada requisito contribuye de forma positiva o negativa a las Softgoals, por ejemplo, el requisito "Private Survey" lo hace de forma positiva (Some +) a la Softgoal "Security". Para fines prácticos, en la figura 5 se presenta el modelo de requisitos A-OOH en el ambiente Eclipse Modeling Framework (EMF). En la parte superior de la figura 5 se puede ver el modelo de requisitos representado en un diagrama ECORE de EMF; en la parte inferior de la figura se pueden observar las propiedades de cada elemento de i*, en este caso el elemento Means-end el cual indica el camino hacia el requisito "Public Survey". La finalidad de la representación del modelo de requisitos en ECORE es facilitar la ejecución de las reglas de transformación.

Fig. 1.14. Editor gráfico para la especificación de requisitos Web con *i**.

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.

1.6 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. Bohner. 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 (IWWOST)*, 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/atl/`.
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. Abrah ao, 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 Reenginering.* PhD thesis, University of Toronto, Canada, 1995.

Part II

PhD Thesis as a Collection of Papers

2

Web Engineering Approaches for Requirements Analysis - A Systematic Literature Review

The content of this chapter corresponds with the following paper: ***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.***

This chapter describes a standard and integrated model-driven framework with which to design each component of a data warehouse. Once this framework is defined, the focus is on the multidimensional modeling of the data warehouse repository. This chapter specifically defines a conceptual multidimensional model and how it is translated to a relational-based logical representation by using a model-driven approach. The advantages of using the model-driven development to design data warehouses are also enumerated. The content of this chapter corresponds with the part of the approach shaded in the figure below.

The content of this chapter is a paper published in *Decision Support Systems*. This journal focuses on contributions to the concepts and operational basis for decision support systems and techniques for implementing and evaluating decision support systems. This journal specifically encourages the following topics: artificial intelligence, data base management, decision theory, economics, linguistics, management science, mathematical modeling, amongst others. The common thread of articles published in the journal is their relevance to theoretical and technical issues for decision support systems. As data warehousing has been widely accepted as a key technology through which organizations can improve their abilities in data analysis and decision support, this journal is an important forum of publication for research in the data warehouse domain. Finally, it should be mentioned that this journal had an *impact factor* of *1.119* in 2007, according to the *Thomson's Science Citation Index (SCI)* (`http://www.isiwebofknowledge.com/`).

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

3

An MDA Approach for Goal-oriented Requirement Analysis in Web Engineering

The content of this chapter corresponds with the following publication: ***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).***

Since the previous chapter has defined how to obtain a relational-based implementation of the multidimensional conceptual model, this chapter focuses on obtaining a logical representation directly based on multidimensional technology, thus showing the full potential of applying formal transformations within a model-driven approach. The content of this chapter corresponds with the part of the approach shaded in the figure below.

This chapter was published in the *International Conference on Data Warehousing and Knowledge Discovery (DaWaK)*. This conference has become one of the most important international scientific events through which to bring together researchers, developers and practitioners to discuss the latest research issues and experiences in developing and deploying data warehousing and knowledge discovery systems, applications, and solutions. Each year, *DaWaK* seeks to introduce innovative principles, methods, algorithms and solutions to challenging problems faced in the development of data warehousing, knowledge discovery and data mining applications. *DaWaK* is, therefore, a leading international forum for the presentation and discussion of current research and applications in which the major emphasis is on data warehousing. The high quality of this conference can be demonstrated through the following two facts: the *acceptance rate* of this conference is usually around *30%*, and the *Estimated Impact of Conference (EIC)* is *0.86* according to *The Computer Science Conference Ranking Website* (http://www.cs-conference-ranking.org/home.html).

.png

\AM@currentdocname .png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

4

Impact Analysis of Goal-Oriented Requirements in Web Engineering

The content of this chapter corresponds with the following publication: ***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.***

The previous chapters begin by defining multidimensional models at the conceptual level. However successful data warehouse design needs to be based upon a requirement analysis phase if it is to adequately represent the information needs of decision makers. Moreover, since the data warehouse integrates the information provided by data sources, it is also crucial to take these sources into account throughout the development process in order to obtain a consistent reconciliation of data sources and information needs. In this chapter, a requirement analysis approach for multidimensional modeling of data warehouses and the corresponding reconciliation process is developed. The content of this chapter corresponds with the part of the approach shaded in the figure below.

The content of this chapter corresponds with a paper published in the *Data & Knowledge Engineering (DKE)* journal. This reaches a world-wide audience of researchers, designers, managers and users. The major aim of the journal is to identify, investigate and analyze the underlying principles in the design and effective use of database and knowledgebase systems. This journal publishes original research results, technical advances and news items concerning data engineering, knowledge engineering, and the interface of these two fields. It should also be mentioned that this journal had an *impact factor* of *1.144* in 2007, according to the *Thomson's Science Citation Index (SCI)* (`http://www.isiwebofknowledge.com/`).

Finally, it is worth mentioning that this paper has been ranked by *ScienceDirect* (`http://www.sciencedirect.com/`) as one of the *25 hottest articles* published in DKE during last quarter of 2007 (`http://top25.sciencedirect.com`).

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

5

A Goal-Oriented Approach for Optimizing Non-Functional Requirements in Web Applications

The content of this chapter corresponds with the following paper:
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.

The development of a data warehouse requires an in-depth analysis of data sources. In the previous chapter, it is assumed that documentation of the data sources is available. However, this is not always true, since in real scenarios data sources are, in reality, legacy systems and their manual analysis may be extremely difficult. In order to overcome these problems, this chapter considers the development of a data warehouse as a modernization scenario which addresses the analysis of the available data sources, thus discovering multidimensional structures with which to either derive a data-driven conceptual multidimensional model or to reconcile a requirement-driven conceptual multidimensional model with data sources. The content of this chapter corresponds with the part of the approach shaded in the figure below.

The content of this chapter was published in the *International Conference on Conceptual Modeling (ER)*. This is one of the most important conferences in data and process modeling, database technology, and database applications. This conference is a wide forum for researchers and industrial experts interested in all aspects of database and information systems design and usage. Topics of interest include data warehousing and business intelligence. *ER* is a top-ranking conference, since it has an *Estimated Impact of Conference (EIC)* value of *0.91* according to *The Computer Science Conference Ranking Website* (`http://www.cs-conference-ranking.org/home.html`). The *acceptance rate* of this conference is usually around *20%.*

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

\AM@currentdocname .png

.png

Part III

Appendix: Papers Already Submitted

A

Requirements in Web engineering: a systematic literature review.

The content of this appendix has been submitted to XXXXX *[IF2007: 1.144].*

Requirements in Web engineering: a systematic literature review

José Alfonso Aguilar[a], Irene Garrigós[a], Jose-Norberto Mazón[a]

[a]*Lucentia Research Group, Department of Software and Computing Systems, University of Alicante, San Vicente del Raspeig, 03080, Spain*

Abstract

Context: Requirements engineering is one of the most crucial stages in software development in which designers can attempt to fully satisfy users' needs, and an effective definition of requirements both contributes towards making the right design decisions and helps to support change management, thus improving the quality of the final software product. As a subdiscipline of software engineering, Web engineering methodologies should consider a requirement engineering stage that suits the Web's large heterogeneous user community.

Objective: The objective of this article is to classify the literature with regard to approaches for the analysis and specification of requirements in Web engineering, with which has allowed us to formally obtain the current state-of-the-art.

Method: Based on the systematic literature review method proposed by Barbara Kitchenham, we reviewed publications from ACM, IEEE Computer Society Digital Library, Science Direct, DBLP and Google Scholar. From a population of 3059 papers, we identified 43 primary studies that provide information about requirements in Web engineering.

Results: The results of this systematic literature review have shown that Web engineering methods were not designed to properly address design through the analysis of Web engineering requirements. What is more, they simply place less relevance on requirements traceability, or its corresponding techniques are poorly applied.

Conclusion: Web applications can no longer be considered as common software systems owing to the diversity of users who have access to these applications. It is therefore necessary to provide solutions in the requirements engineering field in order to specify and develop this type of applications by considering the huge heterogeneous user community, and their needs and goals.

Keywords: Web Engineering, Requirements Engineering, Requirements Analysis, Systematic Literature Review

1. Introduction

Web applications differ from traditional stand-alone software in that their user community is both large and heterogeneous, since respectively (i) Web applications are designed to be used by everybody, and (ii) the users of Web applications may have different needs, goals and preferences. Several special requirements must consequently be considered: what information the Web application should offer (content requirements), which scenarios should be defined in the Web application in order to offer this information (navigational requirements), and how the user or groups of users should be provided with this information (functional requirements). Web engineering methodologies [13, 14, 16, 18, 43] have therefore provided different mechanisms with which to consider these new requirements while developing Web applications [26].

Bearing these considerations in mind, this paper presents a systematic literature review [40] based on our previous work [5] with which to formally obtain the current state-of-the-art with regard to requirements analysis and specifcation in Web engineering in order to reveal their advantages and disadvantages. Roughly speaking, a systematic literature review is useful for formally defining, among other things, (i) the problem to be addressed, (ii) the information sources, (iii) keywords to search for information, (iv) the inclusion and exclusion criteria to be applied to the papers found in the searches; and (v) the templates used to order and classify the information collected from the papers found. This research technique originated in the field of med-

*ja.aguilar@dlsi.ua.es

Email addresses: ja.aguilar@dlsi.ua.es (José Alfonso Aguilar), igarrigos@dlsi.ua.es (Irene Garrigós), jnmazon@dlsi.ua.es (Jose-Norberto Mazón)

ical research and was successfully adapted to software engineering by Barbara Kitchenham [40][1].

Finally, the aim of this review is to: (i) structure the conceptual basis of requirements engineering approaches for Web applications, and (ii) shed light on potential avenues for future research in this area.

The remainder of this paper is structured as follows: Section 2 presents those requirements engineering and Web engineering concepts which are relevant to the context of this paper. The systematic literature review is detailed in Section 3. The analysis and discussion of this work and our suggestions for future research are presented in Section 4 and Section 5, respectively. Finally, our conclusions are provided in Section 6.

2. Requirements and Web engineering concepts

Requirements engineering is the process of discovering, analysing, documenting and verifying the services that should be provided by a software system, along with its operational constraints. According to [55], a requirements engineering process can be divided into five steps:

- **Requirements elicitation.** This is the initial step in requirements engineering whose objective is to discover what problem needs to be solved. The main goals of this step are the identification of the *stakeholders*, the objectives a software system must meet, the tasks that users currently perform and those that they might wish to perform. Requirements elicitation is often carried out through the application of various techniques, such as (i) traditional techniques (questionnaires, interviews, etc.), (ii) group elicitation techniques (brainstorming or focus groups) [51], (iii) prototyping (e.g., provoking discussion in a group elicitation), (iv) modelling techniques (e.g., goal-based methods such as *i** [71]), (v) cognitive techniques (specially developed for knowledge acquisition) [62], and (vi) contextual techniques (e.g. the use of ethnographic techniques such as participant observation) [69].

- **Requirements analysis.** This step includes the creation of conceptual models and prototypes with which to achieve the completeness of the requirements. Requirements analysis deals with understanding an organization's structure, its business rules, and the goals, tasks and responsibilities of its members and the data that is needed. This stage allows designers to detect and resolve conflicts that may occur among different *stakeholders* [63].

- **Requirements specification.** This step is an integral description of the behavior of the system to be developed. This description aims to facilitate an effective communication of requirements among different *stakeholders*. The most widely used techniques for requirements specification are, among others, templates, scenarios, use case modelling, and natural language [8].

- **Requirements validation.** This step aims to establish whether the requirements and models elicited provide an accurate representation of the actual *stakeholder* requirements. Some techniques for requirements validation are reviews, audit and traceability matrices. A requirement must be traceable, since it is defined throughout the development process. A traceable requirement ensures that change management is able to assess the impact of a change on the rest of the application. It is therefore necessary for *stakeholders* to be able to see that the final *work product*[2] is useful for achieving their requirements, and that each requirement has been derived within a particular work product. To do this, requirements engineering needs to ensure traceability. Requirements traceability refers to the ability to describe and follow the life of a requirement, in both a forward and a backward direction [37]. While forward traceability is related to following the requirement to its final implementation, backward traceability refers to following the *work product* to its source requirement (i.e., the associated requirement that originated it). Interestingly, traceability between requirements and *work products* is a norm recommended by CMMI (Capability Maturity Model Integration)[3], specifically in CMMI's Level 2 (within the requirements management process area).

- **Requirements management.** This step consists of recognizing changes through continual requirements elicitation, the re-evaluation of risks, and

[1] It is worth noting that the systematic literature review presented in this paper has been inspired by others such as [52], [64], [39], [41], [54], [50] or [42]

[2] A *work product* represents all the documents and models produced through a software development process.

[3] CMMI provides organizations with the essential elements for the effective improvement of a process. CMMI is a trademark owned by the Software Engineering Institute of Carnegie Mellon University.

the evaluation of systems in their operational environment. Requirement management includes techniques and tools for configuration management and version control [28].

After an overview of requirements engineering concepts, it is worth noting that the development of Web applications have some particular requirements that differ from the traditional requirements. These new requirements are defined in the seminal work of Escalona and Koch [26]. In this work, the authors put forward the argument that functional requirements for Web engineering are related to three main features of Web applications: navigational structure, user interface and personalization capability, and that the data structures required by the Web application should also be specified. An overview of each kind of requirement for Web engineering is described below:

- **Content Requirements.** These define the information that is useful for the Web application which should be presented to users. For example, in an online book store some examples might be "book title" or "book category".
- **Service Requirements.** This refers to the internal functionality that the Web application should provide to its users. Following the online book store example, service requirements might be: "register a new client" or"add book to shopping cart".
- **Navigational Requirements.** A Web application must also define the navigational paths which are available. In this respect, some examples are: the user navigation from "index page" to different options such as"consult products by category" or "consult shopping cart".
- **Layout Requirements.** Requirements can also define the visual interface for users, such as "present a colour style", "multimedia support", or "user interaction", among others.
- **Personalization Requirements.** The designer should specify the desired personalization actions to be performed in the final Web application (e.g., "show recommendations based on interest in previously acquired books", "adapt font for visually impaired users", etc.).
- **Non-Functional Requirements.** These are related to quality criteria that the intended Web application should achieve and that may be affected by other requirements. Some examples might be "good browsing experience", "attract more users", "improve efficiency", etc.

This classification of requirements for Web engineering is used throughout this systematic literature review for the sake of understandability and completeness.

These types of requirements are considered in the Web engineering methodologies for the definition of several models. Web engineering approaches commonly use five main models to define a Web application [33], namely:

- **Domain model (DM).** This encapsulates the structure and functionality required of the relevant concepts of the application domain and reflects the static part of the Web application.
- **Navigation model (NM).** This model aims to specify the structure and behavior of the navigation view over the domain data defined. It defines each path on which users can navigate through the Web application.
- **Presentation model (PM).** This model defines the layout of the Web application, i. e. style, font color, etc.
- **Personalization model (PM).** Personalization strategies are specified in this model.
- **User model (UM).** This allows users to be described in terms of personal information, their relations with a particular application domain and the navigational actions performed at execution-time. The structure of information needed for personalization is also described in this model.

3. A systematic literature review for requirements in Web engineering

A systematic literature review is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, topic area or phenomenon of interest. Although, there are several reasons for performing a systematic literature review, the most common are [40]:

- To sum up the remaining data (information) concerning a treatment of technology.
- To identify any gaps in research in progress, thus highlighting areas of potential interest for investigation.

- To serve as a framework in which to properly place new research activities.

A systematic literature review consists of three main phases, namely:

- **Planning the review.** This aims to develop a protocol that specifies the plan that the systematic literature review will follow to identify, assess and accumulate evidence.
- **Conducting the review.** This phase is responsible for achieving the protocol planned in the previous phase.
- **Reporting the review.** This phase is responsible for extending the review to the research community. It culminates with the elaboration of a review report.

Each of these phases is made up of a sequence of steps in order to be applied iteratively, thus simultaneously refining them. These steps are as follows:

- Planning the review.
  - Identifying the need for a systematic literature review.
  - Constructing the research questions that will guide the analysis of the research.
- Conducting the review.
  - Conducting an extensive search and selection of primary studies.
  - Evaluating the quality of the primary studies found.
  - Extracting data from primary studies.
  - Summarizing and synthesizing the results of the primary studies (i.e., meta-analysis).
- Reporting the review.
  - Elaborating a review report (extending the results to others).

3.1. Research questions

As mentioned in Section 1, the goal of this systematic literature review is to summarize the information concerning how requirements engineering techniques are applied in Web engineering, thus detecting avenues for future research. This has been done by defining the following research questions:

1. **Which Web engineering approaches that cover requirements engineering exist? Which of them support requirements traceability?** There are several Web engineering approaches for the development of Web applications, but not all of them necessarily cover requirements engineering and traceability in the same way. It is therefore necessary to detect to what extent different Web engineering approaches cover the requirements engineering stage.
2. **What type of requirements are proposed in Web engineering approaches? Which techniques are used?** This research question refers to detecting the type of requirements defined for Web engineering approaches. The main features, advantages and drawbacks of requirements engineering techniques defined for or adapted to the Web engineering field should also be studied.
3. **Which tools have been developed to analyze and specify requirements in Web engineering?** The main idea of this research question is to analyze those Web engineering approaches that offer tools in order to verify which requirements engineering techniques are supported.

The next step in the systematic literature review is to determine and follow a search strategy. The objective of this search strategy is to create the right search statements with which to answer the research questions. The search strategy defined in this systematic literature review is described as follows.

3.2. Search strategy

The search strategy should be systematic, comprehensive and explicit in order to make a formal search of primary studies with a predetermined order. According to [40], in this phase it is necessary to use the search engines by applying a combination of search terms (keywords) extracted from the research questions. Various experts should then verify and review the search results. Once the steps to be followed in the search process have been defined, it is necessary to define the resources that are available to conduc the review of primary studies. In this systematic literature review the research sources used are as follows:

- **eJournals (repositories with restricted access):** ACM[4], IEEE Computer Society Digital Library[5], Science Direct[6].

[4]http://portal.acm.org
[5]http://ieeexplore.ieee.org
[6]http://www.sciencedirect.com

- **Digital library of scientific literature:** DBLP Computer Science Bibliography[7].

- **Research resources in the World Wide Web:** Google Scholar[8].

These sources were chosen because they provide search engines which facilitate searches in the most relevant conferences and journals in the software engineering field (see Appendix A).

After defining the research questions and the sources to be consulted, the strings to be used in the search were then defined. Based on the structure of the research questions, some keywords were extracted and used to search the primary study sources. We initially had the following keywords: *Web*, *engineering*, *requirements*, *analysis*, *specification*, *methods* and *tools*. However, in order to obtain more concrete and specific results in the field, we decided to link Web with the keywords *engineering* and *requirements*, *requirements* with the keyword *engineering*, and the keyword *Web* with the keywords *engineering* and *methods*. Nevertheless, other words were also added to increase the size of potential relevant studies: *system*, *techniques*, *website*, *design*, *model* and *support*. These words were linked with the keywords *Web*, *requirements*, *methods*, *engineering* and *tool*.

Thus, the search string was thus constructed by using the boolean operators (AND and OR): (*Web Engineering*) OR ((*Web requirements*) OR (*Web system*) OR (*Requirements Engineering*)) AND (*Web Engineering Methods* AND (*techniques*)) AND (*Web site design*) OR (*Requirements model*) OR (*Requirements modeling*)) AND (*Requirements Traceability*) AND (*Tool support*). However, the results from this search string were too imprecise in the case of research sources such as IEEE and ACM. Bearing this consideration in mind, we decided to separate the search strings by search source. This was done by developing a specific type of string for each search source, making the search as accurate and comprehensive as possible. In this systematic literature review the search strings used are as follows:

- **ACM**. *(Web Engineering) OR ((Web requirements) OR (requirements Engineering)) AND (Web engineering methods) AND (techniques) OR (requirements model) AND (requirements traceability) AND (tool support)*

- **IEEE**. *(Web engineering) OR ((Web requirements) OR (Web system) OR (requirements engineering)) AND ((Web engineering methods) OR (techniques)) AND ((Web site design) OR (requirements model)) OR (requirements modeling) AND (requirements traceability) AND (tool support)*

- **Science Direct**. *Web engineering OR Web requirements OR requirements engineering AND Web engineering methods AND techniques AND Web site design OR requirements model OR requirements modeling AND requirements traceability AND tool support*

- **DBLP**. *Requirements AND Web*

- **World Wide Web (Google scholar)**. *Web +engineering +requirements +phase -stage +specification —requirements analysis +techniques —methods —approaches +tool support +model +traceability support +applications*

3.3. Study selection

In this phase, the inclusion and exclusion criteria were defined. These criteria were used, respectively, to both identify potentially relevant studies and exclude non-relevant studies from the primary study search results.

Three inclusion criteria were defined as follows: the first consisted of reading the title and abstract of the primary studies. Documents were considered to be sufficiently relevant if some of the terms used in the search strings appeared in the title or the abstract.

The second inclusion criterion consisted of reading the introduction and conclusion to those primary studies dealing with a specific requirements engineering technique. Lastly, the third criterion involved reading the whole document to discover those primary studies related to requirements traceability.

Basically, this systematic literature review included any primary studies related to the analysis, specification and traceability of requirements in the Web engineering field. In this respect, we deemed that at least the part related to requirements specification in Web engineering must be present in each primary study, since we assumed that not all approaches implement traceability.

With regard to the exclusion criteria, those primary studies whose topic did not match with the analysis, specification and traceability of requirements in Web engineering were excluded, along with duplicated documents of the same study, short papers and tutorials for the development of Web applications, since they would have made a limited impact and contribution.

[7]http://www.informatik.uni-trier.de/ ley/db/index.html
[8]http://scholar.google.com

The search for primary studies was conducted by using the search strategy defined in Section 3.2. In this phase, the *Title*, *Author*, and *Year* of the paper were used to keep track of the information extraction after the first exclusion. Table 1 shows the results of of the search.

Table 1: Search results by source and exclusion criteria.

	ACM	IEEE	Science Direct	DBLP	WWW	Total
Found	416	234	294	44	2071	3059
Duplicates	19	0	5	1	45	70
1st Exclusion	19	11	6	6	35	77
2nd Exclusion	11	5	3	3	21	43
3rd Exclusion	5	2	3	0	3	13

The selection of primary studies was addressed by using the exclusion and inclusion criteria. Table 1 shows the number of publications retrieved after applying the three exclusion criteria, divided by the search source. The first search (without any exclusion criteria) returned a total of 3059 documents. The publications retrieved after applying the first and second exclusion criteria are shown in Tables 2, 3, 4, 5 and 6, separated by search sources (ACM, IEEE, Science Direct, DBLP and WWW). The publications selected when the third exclusion criterion was applied are highlighted in bold type.

Table 2: Search results in ACM after applying first and second exclusion criteria. Final relevant documents (after applying third exclusion criteria) are highlighted in bold type.

ACM		
Title	**Author**	**Year**
A Model-Driven Architecture Approach Using Explicit Stakeholder Quality Requirement Models for Building Dependable Information Systems	Stefan Biffl, Richard Mordinyi, Alexander Schatten	2007
The object-oriented hypermedia design model.	**Daniel Schwabe,Gustavo Rossi**	**1995**
A Requirement Analysis Approach for Using *i in Web Engineering**	**Irene Garrigos, Jose-Norberto Mazon, Juan Trujillo**	**2009**
Model-driven in reverse. The practical experience of the AQUA project	M.J. Escalona, J.J. Gutirrez	2009
On the generation of requirements specifications from software engineering models: A systematic literature review	Joaqun Nicolas, Ambrosio Toval	2009
Transformation techniques in the model-driven development process of UWE	**Nora Koch**	**2006**
A Conceptual Modeling Approach for the Design of Web Applications based on Services	Ricardo Quintero, Victoria Torres, Marta Ruiz, Vicente Pelechano	2006
Requirements traceability in model-driven development: Applying model and transformation conformance	Joao Paulo A. Almeida, Maria-Eugenia Iacob	2007
Supporting Requirements in a Traceability Approach between Business Process and User Interfaces	Kenia Sousa, Hildeberto Mendona, Jean Vanderdonckt	2008
Model Transformations from Requirements to Web System Design	**Nora Koch, Gefei Zhang, María José Escalona**	**2006**
Introducing requirements traceability support in model-driven development of web applications	**Pedro Valderas, Vicente Pelechano**	**2008**

Table 3: Search results in IEEE after applying first and second exclusion criteria. Final relevant documents (after applying third exclusion criteria) are highlighted in bold type.

IEEE		
Title	**Author**	**Year**
An analysis of the requirements traceability problem	Orlena C. 2. Gotel. Anthony C. W. Finkelstein	1994
A navigational Web requirements validation Trhough Animation	Joumana Dargham, Rima Semaan	2008
From Task-Oriented to Goal-Oriented Web Requirements Analysis.	**Davide Bolchini , John Mylopoulos**	**2003**
NDT. A Model-Driven Approach for Web Requirements	**María José Escalona**, Gustavo Aragn	**2008**
Impact Analysis from Multiple Perspectives: Evaluation of Traceability Techniques	Salma Imtiaz, Naveed Ikram, Saima Imtiaz	2008

Table 4: Search results in Science Direct after applying first and second exclusion criteria. Final relevant documents (after applying third exclusion criteria) are highlighted in bold type.

Science Direct		
Title	**Author**	**Year**
A scoped approach to traceability management	Patricia Lago, Henry Muccini, Hans van Vliet	2008
Using established Web Engineering knowledge in model-driven approaches	Ralf Gitzel, Axel Korthaus, Martin Schader	2006
Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems	**Fernando Molina, Ambrosio Toval**	**2009**

Table 5: Search results in DBLP after applying first and second exclusion criteria. Final relevant documents (after applying third exclusion criteria) are highlighted in bold type.

DBLP		
Title	**Author**	**Year**
Conceptual Modeling for System Requirements Enhancement	Eric Le Pors, Olivier Grisvard	2009
From Conceptual Modeling to Requirements Engineering	Colette Rolland	2006
Requirements engineering for web applications-a comparative study.	**María José Escalona, Nora Koch**	**2004**

The primary studies selected according to the search performed are shown in Table 7. A total of 13 relevant documents were found.

3.4. Data extraction

The goal of this phase is to design data extraction forms with which to accurately record the information obtained from the primary studies. The data extraction form must be designed to collect all the information required in order to fully address the research questions [40]. In this phase, the form shown in Table 8 was used to store the information extracted from the search results. This form contained the title of the publication, the journal or workshop in which the paper was published, the publication date, the main author, requirements engineering techniques related to each approach, traceability support offered by each approach, shortcomings with regard to the requirements engineering techniques used, and finally, tool support.

Table 6: Search results in Google after applying first and second exclusion criteria. Final relevant documents (after applying third exclusion criteria) are highlighted in bold type.

WWW (Google Scholar)		
Title	**Author**	**Year**
A Model Driven Approach for the Integration of External Functionality in Web Applications. The Travel Agency System	Nora Koch, Antonio Vallecillo, Gustavo Rossi	2005
Automatic Support for Traceability in a Generic Model Management Framework	Artur Boronat, Jos . Cars, and Isidro Ramos	2005
Hera: Development of semantic web information systems	**Geert-Jan Houben , Peter Barna , Flavius Frasincar , Richard Vdovjak**	**2003**
Metamodeling the Requirements of Web Systems	**Nora Koch, María José Escalona**	**2006**
Requirement Engineering with URN: Integrating Goals and Scenarios	Jean-Franois Roy	2007
Requirements Traceability and Transformation Conformance in Model-Driven Development	Joao Paulo Almeida, Pascal van Eck, Maria-Eugenia Iacob	2006
Applying Transformations to Model Driven Development of Web Applications	Santiago Meli and Jaime Gmez	2006
An MDA Approach for the Development of Web Applications	Santiago Meli Beigbeder and Cristina Cachero Castro	2004
Supporting Stakeholders in the MDA Process	Keith Phalp, Sheridan Jeary, Jonathan Vincent	2007
A Transformational Approach to Model Driven Engineering of Data-intensive Web Applications	Davide Di Ruscio, Henry Muccini, and Alfonso Pierantonio	2008
A Survey on Web Modeling Approaches for Ubiquitous Web Applications	Andrea Schauerhuber, Wieland Schwinger, Werner Retschitzegger, Manuel Wimmer	2002
Model driven architecture: Principles and practice	Alan W. Brown	2004
Weaving Business Requirements into Model Transformations	Ying Zou, Hua Xiao, Brian Chan	2007
ADM: Método de diseño para al generación de prototipos Web rápidos a partir de modelos	Susana Montero, Paloma Daz, Ignacio Aedo y Laura Montells	2006
From requirements to implementations: a model-driven approach for web development	Susana Montero,Paloma Diaz, Ignacio Aedo	2007
WSDM:a user centered design method for Web sites	**O. M. F. De Troyer, C. J. Leune**	**1998**
Web Modeling Language (WebML):a modeling language for designing Web sites	**Stefano Ceri, Piero Fraternali, Aldo Bongio**	**2000**
Extending UML for modeling Web applications	Luciano Baresi, Franca Garzotto, and Paolo Paolini	2001
THESIS A Requirements Engineering Approach for the Development of Web Applications	Pedro J. Valderas Aranda	2007
THESIS Model Driven Language Engineering	Octavian Patrascoiu	2005
THESIS Modelling Adaptive Web Applications in OOWS	Gonzalo Eduardo Rojas Durn	2008

Table 7: Primary studies resulting after applying the third exclusion criteria.

Title
Metamodeling the requirements of Web systems [27].
Model transformations from requirements to web system design [48].
Requirements engineering for Web Applications-a comparative study [26].
Introducing requirements traceability support in model-driven development of web applications [66].
The object-oriented hypermedia design model. [61].
Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems [53].
From task-oriented to goal-oriented Web requirements analysis [10].
Transformation techniques in the Model-Driven Development Process of UWE [44].
NDT. A Model-Driven Approach for Web requirements [25].
A requirement Analysis Approach for Using i* in Web Engineering [33].
Web Modeling Language (WebML): a modeling language for designing Web sites [17].
WSDM: a user centered design method for Web sites [20].
Hera: Development of semantic web information systems [38].

Table 8: Data extraction form used in this systematic review.

Data Extraction Form	
Title	
Journal/Conference	
Date	
Main author	
Requirements engineering techniques	
Traceability support	
Requirements engineering shortcomings	
Tool support	

At this stage of the systematic literature review, the data extraction of primary studies was performed by applying the different inclusion and exclusion criteria. Quality assessment was then performed separately to verify the information extracted. It is better to perform data extraction and quality assessment separately owing to the large quantity of items extracted.

Data synthesis was then performed. Data synthesis involves collating and summarizing the results of the primary studies included. The synthesis is descriptive (non-quantitative) [40] and is carried out by answering the research questions, as in this review (see questions presented in Section 3.1).

4. Analysis of the primary studies

This section presents and analyses the results obtained after performing the extraction and data synthesis phases on the primary studies. Several features (derived from the research questions) of Web Engineering approaches in the primary studies are described. These features are related to requirements engineering techniques, traceability and tool support. The different types of requirements used by each approach are also discussed (according to the taxonomy described in Section 2).

4.1. UML-based Web Engineering (UWE)

UWE [47] is a methodological approach for the development of Web applications which is entirely based on UML. It covers the whole life cycle of the development of Web applications and focuses on adaptive applications.

Development process: UWE is based on Model-Driven Architecture (MDA) [45]. Within MDA, requirements are considered in the CIM (Computational Independent Model) in order to show what the Web application is expected to do without showing details of how it is implemented. Through this requirements model, UWE can generate content, navigation, presentation and process models. Generation is done by clearly and formally establishing a set of Query/View/Transformation (QVT) transformations[9] to be automatically performed [46]. The authors claim that the main benefit of MDA is that less time and effort are needed to develop the whole Web application, thus improving productivity.

Requirements engineering techniques: UWE proposes the use of interviews, questionnaires and checklists as requirements elicitation mechanisms [26]. With regard to requirements specification, UWE models requirements with UML use case diagrams together with textual descriptions of use cases and UML activity diagrams. Use case diagrams are used to represent an overview of the functional requirements, while activity diagrams provide a more detailed view. In recent works [48], UWE has been extended in order to support the UML profile for Web requirements WebRE (Web REquirements metamodel) [27]. This profile has been defined by UWE's authors in a joint work with the authors of NDT (presented below) [27]. Interestingly, UWE considers any kind of requirements according to the taxonomy presented in Section 2: content, service, navigational, layout, personalization, and non-functional requirements.

Requirements traceability: This methodological approach does not support traceability.

Tool support: A plugin called MagicUWE was developed to be used with the CASE tool MagicDraw[10] [12], which allows UWE to provide a CASE (Computer-Aided Software Engineering) tool. Unfortunately, requirements analysis is missing in this tool. It is worth mentioning that in [49] the author proposes an approach called UWE4JSF for the automatic generation of Web applications in JSF (Java Server Faces) derived from UWE models.

Approach limitations: Despite the fact that UWE provides its own tool, it does not provide support for the analysis of requirements, and provides only superficial support for the automatic transformations (related to traceability issues). Moreover, it may be complicated to elaborate use cases when the designer is modelling the navigation of more complex Web applications.

4.2. Navigational Development Techniques (NDT)

NDT [25] is a model-driven approach that covers the requirements analysis phase in the development of Web applications.

Development process: The development process of NDT is divided into three phases:

- Requirements treatment, in which Web application requirements are collected.
- Analysis, which consists of the derivation of the conceptual and navigational models from the requirements.
- Prototyping, in order to develop Web prototypes from the analysis phase.

Requirements engineering techniques: requirements analysis consists of three phases (capture, definition, and validation) which permit the derivation of three conceptual models (content, navigational and abstract interface). To this aim, NDT applies use case diagrams and a set of textual templates for the requirements description [25]. With regard to the types of requirements, NDT considers content, service, navigational, layout, personalization, and non-functional requirements. In [27], the authors of NDT and UWE have jointly developed a UML-Profile for Web requirements called WebRE. This metamodel is useful for specifying a set of QVT transformations in order to obtain several conceptual models (content, navigational and abstract interface) from requirements specification.

Requirements traceability: NDT supports requirements traceability by means of a requirements matrix. This technique is applied to observe the correspondence between the requirements and artefacts (documents in the requirements analysis phase) that satisfy it [24].

Tool support: NDT has tool support for the requirements analysis phase [24]: NDT-Suite. This is actually

[9] http://www.omg.org/spec/QVT/1.0
[10] http://www.magicdraw.com

a set of tools composed of NDT-Profile, NDT-Driver, NDT Quality and NDT Report[11].

- NDT-Profile. This is a profile defined in Enterprise Architect[12].
- NDT-Driver. This uses NDT-Profile to allow model-driven transformations to be executed in order to obtain analysis artefacts.
- NDT Quality. This tool verifies whether NDT constraints and relations have been followed correctly. Traceability errors between phases are also numbered and grouped by phases.
- NDT Report. This tool obtains the requirements, the analysis, the design and the test documents.

Approach limitations: NDT is not a complete methodology for developing Web applications since it needs to be combined with UWE and WebRE. Importantly, NDT does not consider design and implementation phases, and focuses solely on requirements analysis. NDT hampers the development of a complex Web application, since templates are difficult to complete as they require intensive interviews [27]. Furthermore, requirements are difficult to maintain in NDT owing to the use of textual templates for specification. Solving these drawbacks was one of the reasons for developing WebRE metamodel.

4.3. Web Modeling Language (WebML)

WebML [17] is a visual language for specifying the content structure of a Web application and the organization and presentation of such content in a hypertext.

Development process: WebML consists of different phases that must be applied in an iterative and incremental manner. The process has several cycles, each of which produces a prototype or a partial version of the application which permits conducting, testing and evaluation to take place during the early development phases[13]. The development process of this Web engineering approach is divided into six phases:

- Requirements analysis to collect requirements.
- Conceptual modelling to represent the Web system independently of any technological detail.
- Implementation to derive conceptual models in a specific technology.
- Testing and evaluation of the Web application.
- Deployment of the Web application in a specific architecture.
- Maintenance and evolution.

Requirements engineering techniques: Although a requirements specification phase is considered in WebML, a requirements analysis phase is missing. Requirements are specified by using UML use case and activity diagrams. WebML considers six types of requirements,namely content, service, navigational, layout, personalization and non-functional requirements [11].

Requirements traceability: This approach does not support requirements traceability.

Tool support: WebML has tool support: WebRatio[14] [2]. The support that WebRatio offers to WebML is mainly focused on providing facilities for a conceptual modeling phase and for the automatic generation of J2EE code (as opposed to traditional prototyping tools, which generate application mockups). In this context, WebML models are implicitly defined within WebRatio with a document type definition (DTD) and with a UML 2.0 profile. WebML models can therefore be used in conjunction with other notations and can ensure interoperability with other tools.

Approach limitations: The main limitation of this Web engineering approach is that WebML uses use cases as a technique for requirements specifications without an explicit extension to support Web application requirements. Moreover, this approach has no clearly defined techniques for requirements analysis, in addition to which the requirements specification is not supported by WebRatio. Finally, this approach does not support traceability.

4.4. Web Site Design Method (WSDM)

The WSDM method is a user-centered approach [20] since the Web application is defined according to the requirements of the groups of users.

Development process: The development process of this method consists of the following phases:

- Mission statement, in which the purpose of the Web application must be expressed.
- User modelling, in which the users of the Web applications are classified and grouped.

[11] http://www.iwt2.org
[12] http://www.sparxsystems.com/
[13] *http://www.webml.org*
[14] *http://www.webratio.com*

- Conceptual design, in which a navigational model and a class diagram are designed for the navigational paths.

- Implementation design, whose aim is to create a consistent, pleasing and efficient look and feel for the conceptual design made in the previous phase.

- Implementation of the Web application tailored to a chosen implementation environment.

Requirements engineering techniques: The initial two phases of WSDM (denominated as mission statement and user modeling) are responsible for managing requirements through techniques such as concept maps (of roles and activities) and the data dictionary for the definition of functional and security requirements. The requirements in this approach are specified in a textual form and are considered to be content, service, navigational, personalization and non-functional requirements [20, 65].

Requirements traceability: This is not supported by WSDM.

Tool support: No tools are publicly available for WSDM. There have been some implementations to support parts of the method but these were never integrated, i.e. a code generation tool called WSDMtool [70].

Approach limitations: This approach presents the same limitation as NDT: the definition of textual requirements in a complex Web application development process is difficult to maintain owing to the use of text for requirements specification. Moreover, WSDM does not have a fully-supported tool and lacks traceability support.

4.5. *An Object-Oriented Approach for Web Solutions Modeling (OOWS)*

OOWS [30] is an object-oriented software production method that provides conceptual modeling extensions (in terms of models and abstraction primitives) to facilitate the Web application specification [56]. OOWS extends the OO-Method [57] by adding navigational and presentation characteristics.

Development process: OOWS has a model-driven development process [58] in which there are two main steps: conceptual modeling and solution development [19].

- Conceptual Modeling (i.e., problem space). The aim of this step is to obtain the system specification. It is divided into three sub-steps:

  1. Functional requirements elicitation: techniques based on use cases and scenarios are applied to build a conceptual schema.
  2. Classic conceptual modeling: structural, functional and dynamic models are used to capture the system structure and its behavior.
  3. Navigation and presentation modeling: a navigational model is built in order to specify navigational requirements from the class diagram. Once the navigational model has been built, presentation requirements are specified by using a presentation model which is strongly based on the navigational model.

- Solution development (i.e., solution space). Some patterns are applied to obtain the Web application from the system specification.

Requirements engineering techniques: The requirements analysis phase is carried out through a set of strategies, namely: (i) FRT (Functional Refinement Tree), (ii) use cases, and (iii) tasks, task specification and data description diagrams for navigation requirements. Finally, OOWS considers six types of requirements: content, service, navigational, layout, personalization and non-functional requirements [30, 29, 1, 22].

Requirements traceability: With regard to requirements traceability, this approach focuses on tracing requirements by using the navigational model in a model-driven process through a set of transformation rules defined by graph theory, thus omitting traceability from requirements to the other models involved in the development process [66].

Tool support: This approach has a development framework called OOWS-Suite [67], which is integrated with the OlivaNova tool[15] to provide support for requirements engineering.

Approach limitations: Task analysis is a technique used in the requirements specification which may be time-consuming, complex and depends largely on the experience of the analyst for its correct implementation. Moreover, according to [10],the users' needs are not necessarily well defined within their own mind so as to be defined as tasks. From the designer's perspective, the user goals are ill-defined (no good criterion is defined to be satisfied), i.e., they are not easily reducible to specific tasks that can be easily mapped onto interface features. Finally, the last limitation of this approach concerns requirements traceability, which is only from

[15]http://www.care-t.com/

the requirements to the navigational model, signifying that the other conceptual models involved in the Web application development process are ignored.

4.6. Adaptive Object Oriented Hypermedia (A-OOH)

The A-OOH method [33], [32] is an extension of the Object-Oriented Hypermedia (OO-H) modelling method [36]. The main difference between A-OOH and OO-H are the personalization capabilities [31] which support the modelling of adaptive Web sites.

Development process: The A-OOH design process is based on MDA.

- Computational Independent Model (CIM). At this level, the requirements are specified by the Web application designer. This is done by means of a graphic editor that allows the specification of the Web application requirements through the *i** modelling framework, considering the user goals, needs and the *softgoals*.
- Platform Independent Model (PIM). There are several models at this level: domain, navigational, presentation, user and personalization models.These models are generated from the CIM, and can be further completed.
- Platform Specific Model (PSM). This represents the implementation phase of the Web application.

In this approach the generation of conceptual models from requirements bridges the gap between user needs and Web design, since it is ensured (verification) that the conceptual models conform to the requirements specification (model).

Requirements engineering techniques: A-OOH uses the *i** framework [71] by means of a UML profile in order to create a requirements model. Conceptual models are then generated by means of model transformations, which are applied automatically. A-OOH applies the taxonomy presented in Section 2 to classify the types of requirements used (content, service, navigational, layout, personalization and non-functional) [33].

Requirements traceability: Traceability in A-OOH is partially supported [4], since the authors only provide support for traceability from the CIM level (requirements) to the PIM level [3] by means of weaving models [6] and QVT (Query View Transformation) transformations.

Tool support: The development process is supported by the Eclipse[16] development platform [23].

[16]http://www.eclipse.org/

Approach limitations: One limitation of this approach is that traceability in A-OOH is partially supported, since traceability is only from the CIM to the PIM level and not from CIM to PSM or code level. Moreover, A-OOH does not have a complete supported model-driven tool since it only supports the code generation from PIM level.

4.7. Object-Oriented Hypermedia Design Method (OOHDM)

The OOHDM [34, 61] is an approach for the development of Web applications based on HDM [35], one of the first methods designed for hypertext or hypermedia applications. This approach was the first to propose the separation of aspects, an idea that has been applied by different Web approaches to date.

Development process: The OOHDM development method considers five phases:

- Requirements gathering. *Stakeholder* requirements are captured in this phase.
- Conceptual design. The OOHDM approach models the application domain using object oriented modelling constructs and some design patterns.
- Navigational design. In this phase the OOHDM approach takes into account the user profile and the task, and builds the navigational structure of the Web application.
- Abstract interface design. OOHDM describes the interface for navigational objects and defines the layout of interface objects (responses to external events).
- Implementation. The implementation of interface objects.

Requirements engineering techniques: The first step is to gather the *stakeholders'* requirements. To achieve this, it is first necessary to identify the actors (*stakeholders*) and the tasks they must perform. Next, scenarios are collected (or drafted), for each task and type of actor. The scenarios are then collected to form a use case, which is represented by using user interaction diagrams. These diagrams provide a concise graphical representation of the interaction between the user and the system during the execution of a task. This approach presents guidelines to define conceptual and navigational schemas by means of rules described in natural language, and these rules indicate how the conceptual and navigational schemas can be defined from user interaction diagrams. OOHDM only considers three types of requirements: conceptual, navigational and layout requirements.

Requirements traceability: OOHDM does not support requirements traceability.

Tool support: This approach has an environment called OOHDM-Web. OOHDM-Web allows the implementation of hypermedia applications such as CGI scripts which produce dynamically generated pages, whose contents are fed from a database and integrated with pre-defined templates [60]. This tool can be downloaded from the OOHDM wiki[17].

Approach limitations: This approach does not place much importance on the requirements phase. Moreover, the navigational structure is captured narrowly and is therefore captured individually for each use case from its associated User Interaction Diagram. Navigation is not therefore considered as a part of the whole (the big picture) of the web application. What is more, OOHDM has neither tool support for RE nor tool support for traceability.

*4.8. **Hera***

Hera [68] is a model-driven methodology that supports the design of Web information systems. This approach focuses on the processes of integration, data retrieval, and presentation generation. These processes lead to data transformations based on RDF (Resource Description Framework) models. Hera is principally focused on the development of Web semantic information systems. This approach has been extended and called Hera-S, which is a Web design method that combines the strength of Sesame (a popular open source RDF framework) and the rich modeling capabilities of Hera [15].

Development process: As with other Web engineering methodologies, Hera includes a phase in which the hypermedia navigation is specified. The Hera methodology has the following phases [68, 38]:

- The integration and data retrieval phase, which generates a conceptual model from a user query. This model contains the data for which the application will generate a presentation (here this is considered to be the navigation).

- The presentation generation phase considers how to select and obtain the data from the storage part of the application. The handling of the interaction from users is also specified (querying, navigation, or application-specific user interaction).

[17] http://www.tecweb.inf.puc-rio.b

Requirements engineering techniques: Hera does not have a well-defined requirements analysis phase, but in [7], the authors propose a specification method and a procedure for the automatic generation of a target model from a workflow model. The workflow model describes the business process in the system and the collaboration with users and external systems by means of a task model, in which the actors' tasks only contain activities providing interaction with the system process. This approach considers requirements of the content, service, navigational, layout and personalization types.

Requirements Traceability: This approach does not support requirements traceability.

Tool support: In the Presentation generation phase, the Hera method uses Saxon 7.0[18] (XPath 2.0, XSLT 2.0). For the purpose of data retrieval, Hera uses the RDF query language RQL[19] and its Java-based interpreter called Sesame[20].

Approach limitations: This approach does not have a well-defined requirements phase and lacks a tool to support requirement engineering. What is more, this approach does not have support for traceability.

5. Discussion

This section provides an analysis of the approaches studied above, with the objective of answering the research questions presented in Section 3.1. This analysis is principally focused on the techniques applied to each approach in the requirements analysis phase, requirements terminology, requirements traceability, tool support and approach applications.

*5.1. **Requirements terminology***

The research addressing requirements in Web engineering has produced a heterogeneous terminology for requirements which hinders further progress. A unifyied vocabulary proposed by [26] (introduced in Section 2) is therefore provided to shed light on (i) the expressivity of current approaches when considering requirements in Web engineering, and (ii) the correspondences between the types of requirements used by each approach with the aforementioned taxonomy. An overview of this is shown in Table 9.

In the same order of ideas, of all the approaches presented in Section 4, only A-OOH [33], UWE [9], WebML [11], OOWS [30], and NDT [25] cover every

[18] http://saxon.sourceforge.net
[19] http://www2002.org/CDROM/refereed/329
[20] http://www.openrdf.org

Web Approach	Type of Requirement					
	Content	**Service**	**Navigational**	**Layout**	**Personalization**	**Non-Functional**
UWE	Content	Process	Navigation	Presentation	Adaptation	Non-Functional
NDT	Storage Information	Functional	Interaction	Interaction	Actor	Non-Functional
WebML	Content	Service	Navigational	Presentation	Personalization	Non-Functional
WSDM	Content	Functional	Navigational	X	Personalization	Security and Usability
OOWS	Functional	Functional	Navigational	Presentational	Presentation	Non-Functional
OOHDM	Content	X	Navigational	Layout	X	X
HERA	Content	Service	Navigational	Presentation	Personalization	X
A-OOH	Content	Service	Navigational	Layout	Personalization	Non-Functional

Table 9: Types of requirements and terminology used by each Web engineering approach.

type of requirements relevant to the classification introduced in Section 2. However, these approaches use their own terminology to name each type of requirement, with the exception of A-OOH which applies the classification directly.

Although the approaches presented in Table 9 share, in a few cases, a term with the same name as regards the taxonomy, some types of requirements are used to consider extra functionality, i.e., NDT includes Navigational and Layout requirements in the concept of Interaction requirements and OOWS uses Content and Service requirements in the terms used for Functional requirements [22, 29, 1].

Finally, Non-Functional requirements are considered in a very general manner by almost all the approaches, as can be noted in Table 9. The only approach that considers Non-Functional requirements in a more detailed form is WSDM, which details the Security and Usability Non-Functional requirements to use. [65].

5.2. *Requirements analysis*

In Web engineering, almost all the methodological approaches studied in this systematic literature review (OOWS, NDT, OOHDM, A-OOH, UWE, WSDM, WebML, HERA) consider a requirements engineering process. Hera is the exception, because it does not have an explicit requirements analysis phase, regardless of its being an engineering method for the development of Web applications. In this context, the UWE, NDT and A-OOH approaches are those which have placed greater importance on the requirements analysis phase by defining a set of formal guidelines to be used. On the one hand, A-OOH has a development process based on MDA. This process considers the CIM model in MDA as the requirements model, in the same way that UWE does. Both approaches consider the requirements from the early stages of the Web application development process, thus making the development and maintenance of Web applications easier, whilst fulfilling the project budget. On the other hand, NDT focuses solely on the requirements analysis phase and uses the UWE models to cover the entire Web application development process.

5.3. *Techniques applied*

The techniques applied to each approach in the requirements analysis phase involve a specific set of technologies. Table 10 shows the requirements engineering techniques applied by each Web engineering approach for the analysis and specification of Web requirements. This table shows (i) a trend towards the application of use cases, since this technique is used by OOWS, WebML, NDT, UWE and OOHDM for requirements specifications, and (ii) the persistence of the technique of UML profiles, which is applied by UWE and A-OOH respectively.

Use cases are widely accepted and used in traditional software engineering. Iit is not therefore surprising that Web engineering methodologies also use them to model the possible scenarios that may occur when the user interacts with the Web application. The second technique applied is **UML profiles**, which is used to provide a generic extension mechanism with which to customize UML models for particular domains. The UML profiles technique is, in this respect, applied by UWE and A-OOH since both have an MDA-based development process, UML profiles can therefore be used to adapt particular concepts from a particular domain to its development process, i.e. A-OOH adapts the *i** framework to the Web engineering field in order to use it as a technique for requirements analysis and specification. Another UML technique which is successfully applied in the Web engineering field is the **Activity Diagram**. This technique (applied by WebML and UWE) is a complementary technique for use case diagrams to model the logic captured by a single use case or usage scenario.

The ***i** framework** is a goal-oriented technique that is applied to requirements analysis in A-OOH in order to describe and evaluate design alternatives and their relationships to the organizational objectives. This framework is used in order to explicitly analyse and model relationships among multiple *stakeholders* (actors in the i* notation). The *i** Framework has proved to be useful for representing: (i) the *stakeholders'* intentions, i.e. their motivations and goals, (ii) dependencies between *stakeholders* to achieve their goals, and (ii) the (positive or negative) effects of these goals on each other in order to be able to select alternative designs for the system, thus maximizing goal fulfilment.

NDT and WSDM, meanwhile, apply **Textual Templates** as a technique for requirements specification with the effort involved to do it this manner. The Textual Template technique is only applicable when the Web application project is not very large, otherwise, textual descriptions will grow significantly, making the textual descriptions difficult to maintain and analyze. This technique can be applied in combination with others, such as in the description of a use case diagram, which, as in the traditional software engineering field, is helpful to the developer, depending, of course, on the level of granularity of the diagram in question.

The OOWS approach uses **Task Analysis** as a requirements specification technique. The term *Task* is frequently applied to activity or process. Task Analysis is a hierarchical representation of which steps have to be performed by a task to achieve a goal. Task analysis is often performed by professionals, and therefore, depends in most cases on the analyst's experience. However, the authors of this approach are currently working on a technique for the specification of requirements through ontologies in order to solve this drawback.

Table 10: RE techniques by each Web engineering approach.

RE Technique	**Approach**
Use Cases	OOWS, NDT, UWE, WebML, OOHDM
Data Dictionary	WSDM
Conceptual Maps	WSDM
Functional Refinement Tree (FRT)	OOWS
UML-Profiles	A-OOH, UWE
Task Diagrams	OOWS
Textual Templates	NDT, WSDM
*i** Framework	A-OOH
Activity Diagrams	WebML, UWE

Functional Refinement Tree (FRT) is a technique applied by OOWS to represent a hierarchical decomposition of business functions of a system which is independent of the actual system structure (it is combined with Task Diagrams).

The requirements specification techniques used by WSDM are **Conceptual Maps of Roles and Activities** and **Data Dictionary**. These techniques are difficult to maintain and analyse owing to the fact that the requirements are basically defined in textual form. In this respect, describing navigational requirements by means of textual descriptions is not an easy task owing to the description of alternative navigation paths through the Web application.

The last technique is the **User Interaction Diagram (UID)**. This is used by OOHDM to specify the interaction described in a use case. This technique is basically used to support the communication between the designer and users and to validate use cases.

The techniques presented in this section have advantages and disadvantages, i. e., the use of text for requirements specifications in a complex Web application development process is a disadvantage because it is difficult to maintain. This technique could, nevertheless, be extremely useful and comprehensible in the development of simple Web applications. In the same context, task analysis is a set of techniques which is intended to provide a researcher with a complete understanding of what tasks a user really performs, what is needed to complete those tasks, and what tasks a user should be doing. On the other hand, a task analysis can be an extremely time and resource consuming affair.

5.4. ***Requirements traceability***

Traceability support is specifically studied with regard to traceability from requirements to the work products involved in the Web application development process. Traceability is a highly important success factor in software engineering for the reason that, during the design phase, requirements traceability allows us to keep track of what happens when a change request is implemented before a system is redesigned. Traceability can also provide information about the justifications, important decisions and assumptions behind requirements [59]. However, one common problem in the Web applications development process is the absence of requirements traceability in most of the approaches studied in this SLR (Table 11). Only NDT and OOWS cover the traceability issue to some extent. While NDT offers traceability support by means of a requirements matrix [24], OOWS focuses on tracing requirements throughout the navigational model by using a set of transformation rules defined by graph theory [66]. However, both approaches omit traceability from requirements to the other models involved in the development process.

Table 11: Requirement engineering phase and traceability support by Web engineering approach.

Approach	RE Support	Traceability Support
NDT	+	+/-
WebML	+	-
WSDM	+	-
UWE	+	-
OOWS	+	+/-
OOHDM	+	-
Hera	-	-
A-OOH	+	-

Finally, but of no less importance, some approaches maintain unidirectional traceability between the models generated at the initiation of the development process and the rest of the models involved, but not from requirements specification to the rest of the models involved in a bidirectional manner. We believe that bidirectional traceability could be very useful in a Web engineering development process, particularly during validation with customers.

5.5. *Tool support*

This section provides an analysis of the tools offered by each approach with regard to (i) approach support, (ii) requirements analysis and (iii) traceability. Table 12, shows the approaches studied in this SLR as regards tool support for the methodology, RA phase and traceability. All the approaches have a tool support for the methodology. NDT its supported by NDT-Tool, WSDM has a code generation tool called WSDMtool, WebML is supported by WebRatio, UWE by ArgoUWE, the OOWS approach uses OlivaNova, OOHDM has a tool support called OOHDM-WEB and Hera uses two tools, Saxon 7.0 and Sesame.

Table 12: Tool support for the method, requirements analysis phase and traceability for each Web engineering approach.

Approach	Support	Requirement Analysis	Traceability
NDT	NDT-Tool	NDT-Tool	NDT-Suite
WebML	WebRatio	No	No
WSDM	WSDMtool	No	No
UWE	ArgoUWE	MagicUWE, ArgoUWE	No
OOWS	OlivaNova	OlivaNova, OOWS-Suite	AGG, TaskTracer
OOHDM	OOHDM-WEB	No	No
Hera	Saxon 7.0, Sesame	No	No
A-OOH	Eclipse plugins	Eclipse plugins	No

With regard to the requirements analysis phase, only NDT, UWE, OOWS and A-OOH have tool support. NDT does this by means of the NDT-Tool[21], UWE provides a tool that consists of a MagidDraw[22] plugin, the so-called MagicUWE, OOWS combines OOWS-Suite with OlivaNova tool[23], and finally, A-OOH has requirements analysis support by means of an Eclipse plugin [23].

In terms of tool support for requirements traceability, the two approaches that have implemented traceability have tool support. NDT does this by means of the NDT-Suite. The OOWS approach uses two tools, the first of which is the open source tool called AGG (Attributed Graph Grammar System) and the second of which is TaskTracer[24] developed by the authors to generate traceability reports. These reports are helpful to study aspects such as whether all the requirements are supported, the impact of changing a requirement, or how requirements are modelled [66].

5.6. *The range or diffusion of methods for the development of Web applications*

Figure 1 shows the number of publications by each Web engineering approach ordered by year with regard to requirements engineering and tool support. This figure shows a period from 1998 to 2009 based on the relevant documents for this systematic literature review, and also shows the conference, journal or workshop in which each study was published. OOWS has the greatest number of publications: three from 2002 to 2009, followed by two publications concerning UWE. According to this research, the most important conference in the Web Engineering field for the main topics in this systematic literature review is ICWE (*International Conference on Web Engineering*). The conferences, journals and workshops in which the publications were presented can be consulted in Appendix A. This appendix may serve as a future reference for the publication of research work in this area.

We shall conclude this analysis by mentioning the divulgation of the different approaches, which is a highly important issue since it assists in the realisation of important advances in the standardization of Web application development. The approaches must therefore be well-known in both the academic world and the software industry. In this respect, it is worth mentioning the support offered by NDT, WSDM, UWE and WebML through their websites. They all offer examples, published papers and their respective tools to everyone who visits their website, with the exception of WSDM, which only offers the downloading of published papers because the tool's licence is not free. In the particular case of UWE and WebML, it is necessary to mention that their websites provide guided step

[21] http://www.iwt2.org
[22] http://www.magicdraw.com
[23] http://www.care-t.com/
[24] http://eecs.oregonstate.edu/TaskTracer/

by step examples with which to study and practice the development of Web applications using their respective support tool. This confirms why these two approaches are those most frequently used in academic (university) projects.

5.7. *Suggestions for future research*

The results of this systematic literature review show that Web engineering methods were not specifically developed to address the analysis of Web engineering requirements, and new adjustments are therefore necessary since:

- Those Web engineering approaches that have a requirements analysis phase do not provide a systematic method with which to design models from requirements since they consider the requirements analysis phase without real user goals and needs. This gap must be covered, because the successful development of a system depends on satisfying *stakeholders'* needs.

- Virtually none of the Web engineering approaches do not support requirements traceability: they simply place less importance on traceability, do not include it at all, or its corresponding techniques are poorly applied in the Web engineering field. This deciency must be solved, since the main purpose of traceability is to facilitate the understanding of the product under development and the ability to manage any changes that occur during the development process.

None of the approaches studied has a guided methodology for the analysis of Web engineering requirements that considers real user goals and needs from the beginning of the Web application development process. Empirical studies [55] demonstrate that efforts made to provide a detailed business modelling with which to capture the *stakeholders'* requirements in the system to be built considerably reduce drawbacks in later phases of the development process.

Web engineering approaches should therefore adopt a model-driven development process since this offers important advantages, i.e., it improves the development process of a Web application and reduces the development time, thereby reducing the cost of the project. Model-driven development is applied successfully by several Web engineering approaches, but its use should be generalized.

Moreover, although traceability is a very important success factor and quality criterion in software engineering, a common problem in Web development processes is the lack of requirements traceability support throughout different phases of the software development life cycle and its multiple levels of abstraction (down to the code level). In this context, the authors of A-OOH are currently working to provide RT support from requirements model to domain and navigational models by applying *weaving models* [21]. This work is based on the idea that requirements specified in a requirements model expressed in the *i** Framework will satisfy real user expectations of the website, owing this framework's success in modelling user goals and needs in Intelligent Agents Technology [3]. Moreover, with RT support, this approach will produce an impact analysis (verify the impact resulting from the change in any model).

To sum up, the suggestions proposed for future research are:

- Explicitly consider the specification of real user goals and needs from very early stages of the Web application development process in a systematic manner.

- Consider the non-functional requirements from the requirements specification phase in a structured and comprehensive manner.

- Full support for requirements traceability (bidirectional) from requirements specification throughout the different phases of the Web application development process.

6. Conclusion

This work presents the results obtained after carrying out a systematic literature review. The aim was to create a comprehensive review and synthesis of the current state-of-the-art in the literature related to the analysis, specification and traceability of Web requirements. To do this, a total of 3059 papers published in literature and extracted from the most relevant scientific sources were considered, of which 43 were eventually analysed in depth in accordance with the systematic review process adopted.

The results of this systematic literature review have shown that Web engineering methods were not designed to properly address design through the analysis of Web engineering requirements. What is more, they simply place less relevance on requirements traceability, or its corresponding techniques are poorly applied.

In this context, most of the approaches explored in this systematic literature review do not consider the real

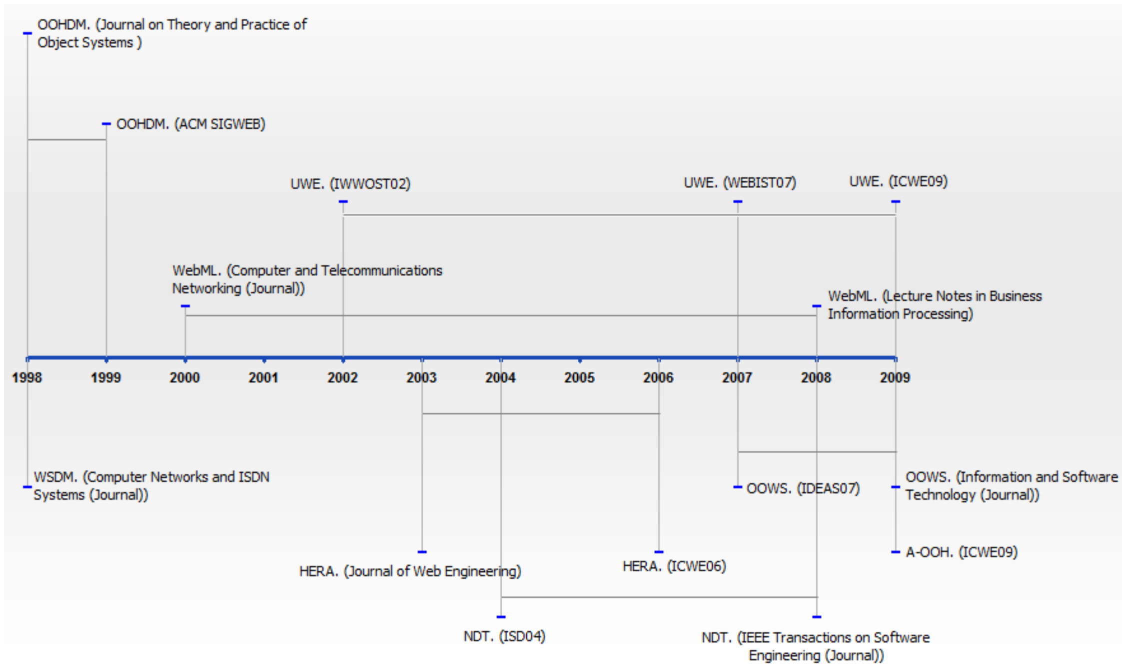

Figure 1: Publications for each Web engineering approach ordered by year (refers only to RA phase).

user expectations of the website or *stakeholders* from the early requirements analysis phase. A-OOH is the exception, as it considers these expectations through the *i** Framework, in which requirements are modelled based on user goals and objectives, thus avoiding the specification of requirements in textual form, and based on Task Analysis, which implies certain limitations.

Web applications can no longer be considered as common software systems owing to the diversity of users who have access to these applications. It is therefore necessary to provide solutions in the Web engineering field in order to specify and develop this type of applications by considering the huge heterogeneous user community, and their needs and goals. Our future work will therefore be a model-driven design approach guided by the requirements analysis in Web engineering. This proposal will be entirely based on the A-OOH method. In order to solve the drawback of the requirements analysis phase without real user goals and needs, the *i** Framework models will be used for the requirements specification. Moreover, with regard to traceability support, the designer will be able to consult the bidirectional tracing of the requirements within the conceptual models derived from requirements specification. This will be helpful in enhancing communication among *stakeholders*, designers and costumers. It is thus necessary to integrate techniques and tools for the automated generation of Web systems. This proposal will therefore be supported by an open source tool for requirements specification. The Web designer will be able to use this tool to obtain the Web application conceptual models from requirements, along with support for bidirectional traceability between requirements and conceptual models.

7. Acknowledgements

This work has been partially supported by the MANTRA project (GRE09-17) from the University of Alicante, and by the MESOLAP (TIN2010-14860) from the Spanish Ministry of Education and Science. José Alfonso Aguilar is funded by CONACYT (Consejo Nacional de Ciencia y Tecnología) Mexico and University of Sinaloa, Mexico.

Appendix A. Journals and conferences

Appendix A shows a list of the conferences, workshops and journals in which the authors of the Web engineering approaches studied in this systematic literature review have presented their work. This list is simply a reference of research publications which may be of interest to readers of this work in the future.

Conferences.

- International Conference on Information Systems Development (ISD).
- International Conference on Web Engineering (ICWE).
- International Conference on Web Information Systems and Technologies (WEBIST).
- Conference on Hypertext and Hypermedia (HT).
- International Conference on Advanced Information Systems Engineering (CAiSE).
- International Conference on Electronic Commerce and Web Technologies (EC-Web).
- Conferencia de Ingeniera de Requisitos y Ambientes de Software (IDEAS).
- International Conference on Conceptual Modeling (ER).

Workshops.

- International Workshop on Web-Oriented Software Technologies (IWWOST).
- Taller sobre Desarrollo de Software Dirigido por Modelos (DSDM) en XV Jornadas de Ingeniera del Software y Bases de Datos(JISBD).

Journals.

- Computer Networks: The International Journal of Computer and Telecommunications Networking[25]
- IEEE Transactions On Software Engineering[26].
- Information and Software Technology[27]
- Journal of Web Engineering[28].
- ACM Special Interest Group on Hypertext, Hypermedia and the Web[29] (SIGWEB) (Newsletter).
- Communications of the ACM[30].

References

[25] http://www.elsevier.com
[26] http://www.computer.org/portal/web/tse/
[27] http://www.elsevier.com
[28] http://www.rintonpress.com/journals/jwe/
[29] http://www.sigweb.org
[30] http://portal.acm.org

References

[1] Abraho, S., Fons, J., Gonzlez, M., Pastor, O., 2006. Conceptual modeling of personalized web applications. In: De Bra, P., Brusilovsky, P., Conejo, R. (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems. Vol. 2347 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 358–362.

[2] Acerbis, R., Bongio, A., Brambilla, M., Butti, S., Ceri, S., Fraternali, P., 2008. Web applications design and development with webml and webratio 5.0. In: Aalst, W., Mylopoulos, J., Sadeh, N. M., Shaw, M. J., Szyperski, C., Paige, R. F., Meyer, B. (Eds.), Objects, Components, Models and Patterns. Vol. 11 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, pp. 392–411.

[3] Aguilar, J. A., Garrigós, I., Mazón, J.-N., 2010. Modelos de weaving para trazabilidad de requisitos web en a-ooh. In: DSDM: Actas del VII Taller sobre Desarrollo de Software Dirigido por Modelos, JISBD, Congreso Espanol de Informatica (CEDI). SISTEDES, Valencia, Espana, pp. 146–155.

[4] Aguilar, J. A., Garrigós, I., Mazón, J. N., Trujillo, J., 2010. An mda approach for goal-oriented requirement analysis in web engineering. Journal of Universal Computer Science JUCS 16 (17), 2475–2494, `http://www.jucs.org/jucs_16_17/an_mda_approach_for`.

[5] Aguilar, J. A., Garrigós, I., Mazón, J. N., Trujillo, J., 2010. Web Engineering approaches for requirement analysis- A Systematic Literature Review. In: 6th Web Information Systems and Technologies (WEBIST). Vol. 2. SciTePress Digital Library, Valencia, Spain, pp. 187–190.

[6] Barbero, M., Del Fabro, M., Bézivin, J., 2007. Traceability and provenance issues in global model management. In: 3rd ECMDA-Traceability Workshop. Citeseer.

[7] Barna, P., Frasincar, F., Houben, G.-J., 2006. A workflow-driven design of web information systems. In: ICWE 06: Proceedings of the 6th international conference on Web engineering. ACM, New York, NY, USA, pp. 321–328.

[8] Bass, L., Merson, P., Clements, P., Bergey, J., Ozkaya, I., Sangwan, R., 2006. A comparison of requirements specification methods from a software architecture perspective. Tech. rep., Software Engineering Institute, Carnegie Mellon University.

[9] Baumeister, H., Koch, N., Zhang, G., 2005. Modelling adaptivity with aspects. In: International Conference on Web Engineering (ICWE 2005. Springer Verlag, pp. 406–416.

[10] Bolchini, D., Mylopoulos, J., 2003. From task-oriented to goal-oriented web requirements analysis. In: WISE '03: Proceedings of the Fourth International Conference on Web Information Systems Engineering. IEEE Computer Society, Washington, DC, USA, p. 166.

[11] Bongio, A., Milano, P. D., Fraternali, P., Maurino, A., Ceri, S., 2000. Modeling data entry and operations in webml. In: The World Wide Web and Databases (WebDB, Selected Papers. pp. 201–214.

[12] Busch, M., Koch, N., 2009. Magicuwe: A case tool plugin for modeling web applications. In: ICWE '9: Proceedings of the 9th International Conference on Web Engineering. Springer-Verlag, Berlin, Heidelberg, pp. 505–508.

[13] Cachero, C., Gómez, J., 2002. Advanced conceptual modeling of web applications: Embedding operation. In: Interfaces in Navigation Design. 21th International Conference on Conceptual Modeling. El Escorial.

[14] Casteleyn, S., Garrigós, I., Troyer, O. D., 2005. Automatic runtime validation and correction of the navigational design of web sites. In: APWeb. pp. 453–463.

[15] Casteleyn, S., Van Woensel, W., Houben, G.-J., 2007. A semantics-based aspect-oriented approach to adaptation in web

engineering. In: HT '07: Proceedings of the eighteenth conference on Hypertext and hypermedia. ACM, New York, NY, USA, pp. 189–198.
[16] Casteleyn, S., Woensel, W. V., Houben, G.-J., 2007. A semantics-based aspect-oriented approach to adaptation in web engineering. In: Hypertext. pp. 189–198.
[17] Ceri, S., Fraternali, P., Bongio, A., 2000. Web modeling language (webml): a modeling language for designing web sites. The International Journal of Computer and Telecommunications Networking 33 (1-6), 137–157.
[18] Ceri, S., Manolescu, I., 2003. Constructing and integrating datacentric web applications: Methods, tools, and techniques. In: VLDB. p. 1151.
[19] Cors, J. F., 2008. OOWS: Un Metode Dirigit per Models per al Desenvolupament d Aplicacions Web. Tesis doctoral en informtica, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia.
[20] De Troyer, O. M. F., Leune, C. J., 1998. Wsdm: a user centered design method for web sites. Comput. Netw. ISDN Syst. 30 (1-7), 85–94.
[21] Del Fabro, M., Bézivin, J., Valduriez, P., 2006. Weaving Models with the Eclipse AMW plugin. In: Eclipse Modeling Symposium, Eclipse Summit Europe. Vol. 2006. Citeseer.
[22] Durán, G. E. R., 2008. Modelling adaptative web applications in oows. Ph.D. thesis, Technical University of Valencia, Spain.
[23] Eclipse, 2010. `http://www.eclipse.org/`.
[24] Escalona, M., Mejías, M., Torres, J., 2004. Developing systems with NDT & NDT-Tool. In: 13th International conference on information systems development: methods and tools, theory and practice, Vilna, Lithuania. pp. 149–59.
[25] Escalona, M. J., Aragón, G., 2008. Ndt. a model-driven approach for web requirements. IEEE Transactions on Software Engineering 34 (3), 377–390.
[26] Escalona, M. J., Koch, N., 2004. Requirements engineering for web applications - a comparative study. J. Web Eng. 2 (3), 193–212.
[27] Escalona, M. J., Koch, N., 2007. Metamodeling the requirements of web systems. In: Aalst, W., Mylopoulos, J., Sadeh, N. M., Shaw, M. J., Szyperski, C., Filipe, J., Cordeiro, J., Pedrosa, V. (Eds.), Web Information Systems and Technologies (WEBIST). Vol. 1 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, pp. 267–280.
[28] Estublier, J., 2000. Software configuration management: a roadmap. In: Proceedings of the Conference on The Future of Software Engineering. ICSE '00. ACM, New York, NY, USA, pp. 279–289.
[29] Fons, J., Garca, F., Pelechano, V., Pastor, O., 2003. User profiling capabilities in oows. In: Lovelle, J., Rodrguez, B., Gayo, J., del Puerto Paule Ruiz, M., Aguilar, L. (Eds.), Web Engineering. Vol. 2722 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 457–461.
[30] Fons, J., Valderas, P., Ruiz, M., Rojas, G., Pastor, O., 2003. Oows: A method to develop web applications from web-oriented conceptual models. In: International Workshop on Web Oriented Software Technology (IWWOST). (2003) 6570. pp. 65–70.
[31] Garrigós, I., 2008. A-ooh: Extending web application design with dynamic personalization. Ph.D. thesis, University of Alicante, Spain.
[32] Garrigós, I., Gomez, J., Houben, G.-J., 2010. Specification of personalization in web application design. Inf. Softw. Technol. 52 (9), 991–1010.
[33] Garrigós, I., Mazón, J.-N., Trujillo, J., 2009. A requirement analysis approach for using i* in web engineering. In: ICWE. pp. 151–165.
[34] Garzotto, F., Paolini, P., Schwabe, D., 1993. Hdm—a model-based approach to hypertext application design. ACM Transactions on Information Systems (TOIS) 11 (1), 1–26.
[35] Garzotto, F., Paolini, P., Schwabe, D., January 1993. Hdm a model-based approach to hypertext application design. ACM Trans. Inf. Syst. 11, 1–26.
[36] Gómez, J., Cachero, C., Pastor, O., 2000. Extending a conceptual modelling approach to web application design. In: CAiSE '00: Proceedings of the 12th International Conference on Advanced Information Systems Engineering. Springer-Verlag, London, UK, pp. 79–93.
[37] Gotel, O., Finkelstein, A., 1994. An analysis of the requirements traceability problem. In: Proceedings of the First International Conference on Requirements Engineering. pp. 94–101.
[38] Houben, G., Barna, P., Frasincar, F., Vdovjak, R., 2003. Hera: Development of semantic web information systems. Web Engineering, 529–538.
[39] Jorgensen, M., Shepperd, M., 2007. A systematic review of software development cost estimation studies. Software Engineering, IEEE Transactions on 33 (1), 33 –53.
[40] Kitchenham, B., 2004. Procedures for performing systematic reviews. Tech. rep., Keele University and NICTA.
[41] Kitchenham, B., Mendes, E., Travassos, G., May 2007. Cross versus within-company cost estimation studies: A systematic review. Software Engineering, IEEE Transactions on 33 (5), 316 –329.
[42] Kitchenham, B., Pretorius, R., Budgen, D., Pearl Brereton, O., Turner, M., Niazi, M., Linkman, S., August 2010. Systematic literature reviews in software engineering - a tertiary study. Inf. Softw. Technol. 52, 792–805.
[43] Koch, N., 2002. The expressive power of uml-based web engineering. In: International Workshop on Web-oriented Software Technology (IWWOST). pp. 40–41.
[44] Koch, N., 2006. Transformation techniques in the model-driven development process of uwe. In: ICWE '06: Workshop proceedings of the sixth international conference on Web engineering. ACM, New York, NY, USA, p. 3.
[45] Koch, N., 2006. Transformation techniques in the model-driven development process of UWE. In: Workshop proceedings of the sixth international conference on Web engineering. ACM, p. 3.
[46] Koch, N., Knapp, A., Zhang, G., Baumeister, H., 2008. Uml-based web engineering. In: Web Engineering: Modelling and Implementing Web Applications. Human-Computer Interaction Series. Springer London, pp. 157–191.
[47] Koch, N., Kraus, A., 2002. The expressive power of uml-based web engineering. In: IWWOST 02: Second Int. Worskhop on Web-oriented Software Technology.
[48] Koch, N., Zhang, G., Escalona, M. J., 2006. Model transformations from requirements to web system design. In: ICWE '06: Proceedings of the 6th international conference on Web engineering. ACM, New York, NY, USA, pp. 281–288.
[49] Kroiss, C., Koch, N., Knapp, A., 2009. Uwe4jsf: A model-driven generation approach for web applications. In: Gaedke, M., Grossniklaus, M., Daz, O. (Eds.), Web Engineering. Vol. 5648 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 493–496.
[50] Lisboa, L., Garcia, V., Lucrédio, D., de Almeida, E., de Lemos Meira, S., de Mattos Fortes, R., 2010. A systematic review of domain analysis tools. Information and Software Technology 52 (1), 1–13.
[51] Maiden, N., Rugg, G., 1996. Acre: selecting methods for requirements acquisition. Software Engineering Journal 11 (3), 183 –192.
[52] Mendes, E., 2005. A systematic review of Web engineering research. In: Empirical Software Engineering, 2005. 2005 Inter-

national Symposium on. IEEE, p. 10.
[53] Molina, F., Toval, A., 2009. Integrating usability requirements that can be evaluated in design time into model driven engineering of web information systems. Adv. Eng. Softw. 40 (12), 1306–1317.
[54] Nicolás, J., Toval, A., 2009. On the generation of requirements specifications from software engineering models: A systematic literature review. Information and Software Technology 51 (9), 1291–1307.
[55] Nuseibeh, B., Easterbrook, S. M., 2000. Requirements engineering: a roadmap. In: ICSE - Future of SE Track. pp. 35–46.
[56] Pastor, O., Abrah ao, S. M., Fons, J., 2001. 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. Springer-Verlag, London, UK, pp. 16–28.
[57] Pastor, O., Pelechano, V., Insfrn, E., Gmez, J., 2004. From object oriented conceptual modeling to automated programming in java. In: Ling, T. W., Ram, S., Lee, M. L. (Eds.), Conceptual Modeling ER 98. Vol. 1507 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 183–196.
[58] Quintero, R., Pelechano, V., Pastor, O., Fons, J., 2003. Aplicación de MDA al Desarrollo de Aplicaciones Web en OOWS. Jornadas de Ingeniería de Software y Base de Datos (JISBD), VIII, 84–668.
[59] Ramesh, B., Jarke, M., 2001. Toward reference models for requirements traceability. IEEE Transactions on Software Engineering 27, 58–93.
[60] Schwabe, D., de Almeida Pontes, R., Moura, I., 1999. Oohdm-web: an environment for implementation of hypermedia applications in the www. ACM SIGWEB Newsletter 8 (2), 18–34.
[61] Schwabe, D., Rossi, G., 1995. The object-oriented hypermedia design model. Communications of the ACM 38 (8), 45–46.
[62] Shaw, M., Gaines, B., 1996. Requirements acquisition. Software Engineering Journal 11 (3), 149 –165.
[63] Sommerville, I., 1995. Software engineering (5th ed.). Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
[64] Staples, M., Niazi, M., 2007. Experiences using systematic review guidelines. The Journal of Systems and Software 80, 1425–1437.
[65] Troyer, O. D., Casteleyn, S., 2001. The conference review system with wsdm. In: IWWOST01. pp. 30–98.
[66] Valderas, P., Pelechano, V., 2009. Introducing requirements traceability support in model-driven development of web applications. Inf. Softw. Technol. 51 (4), 749–768.
[67] Valverde, F., Valderas, P., Fons, J., 2007. OOWS Suite: Un Entorno de desarrollo para Aplicaciones Web basado en MDA. IDEAS, Venezuela.
[68] Vdovjak, R., Frasincar, F., Houben, G., Barna, P., 2003. Engineering semantic web information systems in hera. Journal of Web Engineering 2, 3–26.
[69] Viller, S., Sommerville, I., 1999. Social analysis in the requirements engineering process: from ethnography to method. In: Requirements Engineering, 1999. Proceedings. IEEE International Symposium on. pp. 6 –13.
[70] Wilder, K. V., 2009. Implementation generation for wsdm using web applications framework. Ph.D. thesis, University of Brussel.
[71] Yu, E., 1995. Modelling strategic relationships for process reenginering. Ph.D. thesis, University of Toronto, Canada.

B

A Goal-Oriented Requirements Engineering Approach to Distribute Functionality in RIAs

The content of this appendix has been submitted to 12th International Conference on Web Information System Engineering (WISE 2011)

A Goal-Oriented Requirements Engineering Approach to Distribute Functionality in RIAs

José Alfonso Aguilar[1,3], Sven Casteleyn[2], Irene Garrigós[1], and Jose-Norberto Mazón[1]

[1]Department of Software and Computing Systems
University of Alicante, Spain
[2]Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València, Spain
[3]Computer Science Faculty
University of Sinaloa, Mexico
{ja.aguilar,igarrigos,jnmazon}@dlsi.ua.es
sven.casteleyn@upv.es

Abstract. The challenges and complexity involved in the design and development of Rich Internet Applications are well recognized, and most existing Web 1.0 design methodologies have been augmented to tackle these. However, the focus is mostly on technological and engineering challenges, while only providing limited support for the requirements analysis. This paper addresses this hiatus by extending an existing goal-oriented Web requirements analysis approach to support RIAs. Specifically, we present an approach (based on the Pareto efficiency) to evaluate and select desirable client/server distribution of functionality for RIAs, and guide the designer towards an optimal solution that provides a balanced maximization of the non-functional requirements. We illustrate our approach by using an excerpt of a real-world case study of a bioinformatics company.

Keywords: Web Engineering, Rich Internet Applications, Goal-Oriented Requirements Analysis

1 Introduction

Due to the Web's highly heterogeneous user base and its constant evolution, requirements specification is an important yet difficult task in the Web engineering field. Importantly, a key factor in a Web application's success is to ensure that the functionality required by the users is fulfilled, while at the time facilitating their browsing session [10]. This is tightly related to both (i) functional requirements (FRs) that describe the system services, behavior or functions, e.g., "download a sales report"; and (ii) non-functional requirements (NFRs) which specify a constraint over that functionality, e.g., "usability" [17]. Therefore, Web methodologies should be able to provide a requirements analysis phase in which functional and non-functional requirements are considered.

Unfortunately, both in industrial cases and in Web engineering methodologies, requirements are not sufficiently taken into account when developing a Web application [3]. For stakeholders[1], this results in a mismatch between their expectations and the resulting Web application, and for users in a Web application that poorly supports their requirements and does not fulfill their needs.

As a natural evolution of the Web, Rich Internet Applications (RIAs) enable client-side logic, offer better responsiveness and interactivity, and provide more elaborate and attractive presentation. Their main goal is to tackle the limitation imposed by traditional Web 1.0 technology [6], and provide the user with a richer experience, similar to that of desktop applications. While offering more possibilities to satisfy the user's expectations, RIAs are also considerably more complicated to design and develop. Existing Web design methodologies have been extended to support RIAs, but the proposed extensions mainly focus on presentation issues or interaction capabilities [13]. Requirements analysis is largely left to the discretion of the designer and no systematic, tailored support for RIAs is provided. Nevertheless, this is not a trivial task, as the designer needs to balance classical Web requirements, such as those functionalities (i.e., FR) related to navigation, and RIA specific NFR, such as "Responsiveness" or "Bandwidth reduction". Furthermore, the designer needs to take into account that the distribution of functionality between client and server influences the satisfaction of the NFRs; the choice where to place certain functionality is thus critical.

In our previous work [2, 11], we presented a GORE (Goal-Oriented Requirements Engineering) approach based on the *i** modeling framework to specify Web requirements. In this paper we extend that approach to support requirements engineering in RIAs. Specifically, we focus on studying the requirements distribution between client and server in order to provide a balanced maximization of the satisfaction of NFRs. For this purpose, we present a proposal that allows the designer to obtain and evaluate good client/server configurations (requirements distribution), i.e. those that maximize the NFRs. Our proposal is based on the Pareto efficiency, which is useful when there are multiple competing and conflicting objectives [5] that need to be balanced, as is the case here. In essence, a group of non-dominated optimal solutions (called Pareto front) is created, i.e., solutions that optimize the NFRs (corresponding to *softgoals* in our GORE framework) in such a way that no single NFR can still be improved, without negatively affecting another. From these balanced optimizations, the designer can select the final solution by taking into account the importance of the different NFRs. We present our results by using an excerpt of a real-world case study of a start-up bioinformatics company in order to effectively show how our approach can aid designers to obtain an adequate client/server distribution of functionality in RIAs.

This paper is structured as follows: Section 2 presents the related work. Section 3 describes our previous work on GORE for the Web 1.0. The contribution

[1] Individuals or organizations who affect or are affected directly or indirectly by the development project in a positive or negative form [17].

of this work is presented in Section 4. Section 5 describes a real world case study. Section 6 presents the ongoing work related to the implementation of our approach. Finally, the conclusion and future work are presented in Section 7.

2 Related Work

Research in Web engineering has shown that RIA development is a difficult challenge which requires extending the traditional Web engineering approaches [14], i.e., to improve the development process and adapt the conceptual models in order to support richer presentation and interactive user interfaces.

Several Web 1.0 Web engineering approaches have been extended to support RIAs, i.e., (i) **WebML** (Web Modeling Language) [4], (ii) **OOWS** (Object-Oriented Approach for Web Solutions Modeling) [19], (iii) **OOHDM** (Object-Oriented Hypermedia Design Method) [18], and (iv) **UWE** (UML-based Web Engineering), which has three extensions for RIA development (UWE-Patterns [9], UWE-R [12] and UWE-RUX [15]). For a complete analysis of the techniques used for requirements analysis by the approaches described in this paragraph we refer the reader to [3].

Some extension have been created in order to be combined with Web engineering approaches as the ones aforementioned, these are: (i) the **RUX-Model** [16], solely for the design of UIs for RIAs over existing HTML-based Web Applications in order to give them multimedia support, and (ii) the **OOH4RIA** [13], which defines a model-driven development process to obtain a complete RIA for the GWT (Google Web Toolkit) framework, and is supported by a CASE tool.

On the other hand, new methods have been created exclusively for the development of RIAs. **IAML** (Internet Application Modelling Language) [21] is a modelling language that directly uses RIA concepts as first-class modelling citizens [20]. IAML has a CASE tool for the development of RIAs based on EMF (Eclipse Modeling Framework) and GMF (Graphical Modeling Framework). The **ADRIA** (Abstract Design of Rich Internet Applications) [7] method is an UML-based approach, whose design activities depart from an object-oriented analysis, and focus on the design of events triggered by user interactions.

In some of these approaches, e.g. OOH4RIA, IAML, RUX, the requirements engineering phase is not considered at all. In others, classical software engineering techniques are used, such as use cases in UWE, WebML, OOWS, OOHDM and ADRIA, activity diagrams in UWE, WebML and OOHDM, or task models in OOWS. These techniques are inherited from their respected Web 1.0 methods, and are not well-adapted to take RIA's requirements into account, or deal with RIA specificities (i.e., client/server distribution). Our approach complements these techniques, is method-independent, and allows to focus on satisfying user goals and objectives by means of our goal-oriented approach, while taking specificities of RIAs, in particular the client/server distribution, into account.

Finally, some interesting goal-oriented techniques for requirements analysis have been proposed for non-Web engineering approaches, i.e., [8]. This work evaluates i^* models based upon an analysis question (what-if analysis) and the human judgment. To this aim, this procedure uses a set of evaluation labels to represent the satisfaction or denial level of each element in the i^* model. Unfortunately, these general approaches have not been adapted to Web engineering.

3 Specifying Requirements in Web Engineering

In this section we summarize our requirements analysis approach for Web 1.0 applications based on the i^* modeling framework [1, 2, 11]. This approach is the starting point of our approach for analyzing the client/server distribution of functionality when RIA-specific requirements are considered.

The development of Web applications involves different kinds of stakeholders with different needs, which depend on each other to achieve their goals, perform several tasks or obtain some resources. For example, the Web administrator relies on new clients for obtaining data in order to create new accounts. The i^* framework [22] is a GORE framework used to explicitly analyze and model these relationships among multiple stakeholders (actors in the i^* notation), along with their goals, tasks and resources. Essentially, the i^* framework consists of two models: the strategic dependency (SD) model, to describe the dependency relationships (represented as) among various actors in an organizational context, and the strategic rationale (SR) model, used to describe actor goals and interests and how they might be achieved. Therefore, the SR model (represented as a dashed circle) provides a detailed way of modeling the intentions of each actor (represented as a circle), i.e., internal intentional elements and their relationships:

- A goal (elipse) represents an (intentional) desire of an actor. Interestingly, goals provide a rationale for requirements but they are not enough for describing how the goal will be satisfied. This can be described through means-end links () representing alternative ways for fulfilling goals.
- A task (hexagon) describes some work to be performed in a particular way. Decomposition links () are useful for representing the necessary intentional elements for a task to be performed.
- A resource (rectangle) represents some physical or informational entity required for the actor.
- A softgoal (eight-shape) is a goal whose satisfaction criteria is not clear-cut. How an intentional element contributes to the satisfaction or fulfillment of a softgoal is determined via contribution links ($\xrightarrow[\text{hurt}]{\text{help}}$). Possible labels for a contribution link are “make”, “some+”, “help”, “hurt”, “some-”, “break”, “unknown”, indicating the (positive, negative or unknown) strength of the contribution.

Our previous work [2] presented an approach for adapting the i^* framework for specifying requirements typically encountered in Web applications (this adaptation is supported by a metamodel described in Fig. 1):

- Service requirements refer to the internal functionality the system should provide to its users, e.g., “register a new client”.

– Navigational requirements aims to define the paths needed for the user to browse the Web applications, e.g., "consult products by category".
– Layout requirements for defining the visual interface for the users, e.g., "present certain text style for headers".
– Personalization requirements allow the designer to specify the desired adaptation functionality to be performed in the final Web application, e.g., "adapt font for visual impaired users".
– Content requirements define the content that the Web application presents to its users, e.g., "product information".
– Non-Functional requirements are related to quality criteria that the intended Web system should achieve and that can be affected by other requirements, e.g., "responsiveness".

An example of an $i*$ model for Web engineering can be seen in Fig. 3 with explanation in Sect. 5.

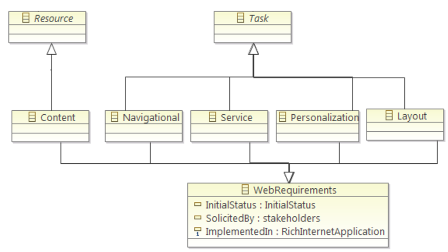

Fig. 1: An excerpt of the $i*$ requirements metamodel for Web domain.

Finally, it is worth noting that FRs (navigational, layout, service and personalization) are represented as tasks, while the requirements related to the data (content) are represented as resources. There is thus a clear separation between these two concerns; in this paper we focus on the (distribution of) FRs. Also, NFRs are represented as softgoals. For further details, we refer the reader to [2].

4 Optimizing RIA functionality distribution by using Pareto efficiency

A critical task in RIA requirements engineering is to determine a *good* distribution of functionality (client/server). In other words, for each FR the designer needs to decide where it will contribute best to the realization of the NFR or softgoals (according to our framework), i.e. on the client or on the server side. This is not a trivial decision, as each requirement influences different softgoals differently depending on whether it resides on the server or the client. In addition, maximizing one softgoal often leads to decreased support for another softgoal, so there is no notion of a single best client/server configuration and a tradeoff among softgoals should be made.

Pareto efficiency [5] is an ideal candidate to tackle this problem, as it is designed to balance competing concerns. It is a notion from economics widely applied to engineering, which is described as follows: given a set of alternative allocations (server or client) and a set of individuals (requirements), allocation A is an improvement over allocation B only if A can make at least one individual better (maximizing softgoals) than B, without making any other worse (weakening softgoals). Therefore, in our approach, a Pareto optimal (client/server) configuration is one so that no other configuration better satisfies one single softgoal, while satisfying the others equally. The set of Pareto optimal configurations are thus candidate *good* distributions. It includes the solutions that best satisfy each single softgoal, but also those that provide an optimal tradeoff between softgoals. As such, it forms a perfect basis to make a well-informed design decision for functionality client/server distribution.

More formally, finding the set of Pareto optimal configuration can be defined as the problem of finding a (decision) vector of decision variables X (i.e., a valid client/server distribution), which maximizes a vector of M objective functions $f_i(X)$, (i.e., the satisfaction of softgoal i in client/server distribution X, where $i \in \{1, ..., M\}$ (with M the amount of softgoals). To do so, the concept of domination between vectors is defined as follows: a decision vector X is said to dominate a decision vector Y ($X \succ Y$) if and only if their corresponding objective vectors of objective functions $f_i(X)$ and $f_j(X)$ satisfy: $\forall i \in \{1...M\} : f_i(X) \geq f_i(Y)$ and $\exists i \in \{1, ..., M\} : f_i(X) > f_i(Y)$. We then say that all decision vectors that are not dominated by any other decision vectors form the Pareto optimal set, while the corresponding objective vectors are said to form the Pareto front.

Following the Pareto efficiency, we defined the following steps to determine the Pareto optimal configuration for distributing FRs between client and server, while the softgoals are balanced and maximized (Figure 2). Our approach supports bith scenarios: (i) the evolution of an existing Web 1.0 specification to support RIAs, and (ii) from-scratch specification.

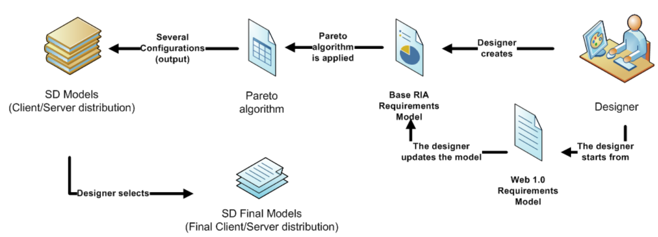

Fig. 2: An overview of our RIA requirement analysis approach.

Before defining each step of our approach for dealing with the distribution of functionality, it is worth noting that the distribution of data is also a relevant requirement when developing RIAs, and in fact, both are coupled. Indeed, when deciding to implement particular functionality on the client, the relevant data needed to realize this functionality needs to be duplicated on the client. A straightforward solution is thus to duplicate in the client any resource (content requirement) that is related to a task (navigation, layout, personalization or service requirement) located in the client. We are currently applying this solution, allowing us to focus only on functionality. However, it needs to be mentioned that the duplication of data and its communication to the client causes some additional effects, which might influence, both positively and negatively, the maximization of softgoals. For example, duplication and communication of data to the client side might cause slower initialization, but on the other hand, re-use of this data for different tasks will increase responsiveness. We are therefore considering in our future work to also adopt content requirements in our Pareto algorithm to include their effect on the maximization of softgoals.

Step 1.- Create the base RIA requirements model. When no Web 1.0 artifacts are available, a Web 1.0 requirements model is created (see Sect. 3), but additionally including RIA specific requirements, softgoals and the contribution links they induce. The RIA specific requirements are those that are only realizable using RIA technology (e.g., "Provide a graphical view", "Capture webcam"). The RIA specific softgoals arise from the fact that RIA technologies are used; they would otherwise not be present, possible or relevant. Examples include "Responsiveness" or "Low bandwidth". At this point, we do not yet take into account the distribution of FRs among client and server. Hence, the label of the contribution links (help, hurt, some+, some-, make and break) cannot yet be determined, as this is dependent on where each requirement is implemented. For example, the requirement "Browse result set" will "help" the softgoal "Responsiveness" if implemented on the client, but "hurt" it when implemented on the server. Therefore, we leave the label of the contribution links as "unknown" at this moment; later, when we take into account server/client distribution, we will assign the correct label to each of the contribution links. In some cases, the label of the contribution can be directly assigned[2]: if the contribution involves a requirement that can only be realized on server or client side (i.e., "Provide graphical view" needs client-side technology and is thus only realizable on the client side) or for softgoals to whom a contribution is client or server independent (i.e., "Provide proprietary format" always "hurts" "Compatibility" as it concerns a non-standard format).

In case a Web 1.0 requirement model was already available, the model is evolved as follows: the RIA specific requirements, softgoals and contribution links (with "unknown" label) are added to the existing model. For contributions that were already present in the Web 1.0 requirement model, their label is replaced by "unknown" if it may be influenced by the client/server distribution of the requirements.

Step 2.- Obtain the Pareto optimal client/server distribution. We provide a set of sub-steps based on the Pareto efficiency, to obtain a set of valid client/server configurations that balance the maximization of softgoals, rather than maximizing one single softgoal.

[2] Note that leaving these "unknown" in this step would yield the same final result; assigning them now only simplifies the following steps.

A) Obtain a decision vector for requirements distribution. Each possible distribution of N FRs between client and server is stored in a decision vector X_v: $\forall v : 0 \preceq v < 2^N$, $\forall i \in \{1...N\} : X_{v_i} = R_i$, where X_{v_i} is the ith element of X_v, $R_i = S$ if the FR i is allocated on the server and $R_i = C$ if the FR i is allocated on the client.

B) Create a weight matrix. Depending on where the FR is implemented (either on the client or on the server), the contribution of each requirement to each softgoal must be quantified. To this aim, the designer creates a matrix by using the following weights to each kind of contribution: $w = 0$ if the requirement does not contribute to any softgoal, $w = +1$ if there is a "Help" contribution link, $w = -1$ if there is a "Hurt" contribution link, $w = +2$ if there is a "Some +" contribution link, $w = -2$ if there is a "Some -" contribution link, $w = +4$ if there is a "Make" contribution link and $w = -4$ if there is a "Break" contribution link.

Therefore, the matrix is defined, so that each entry $W^k_{i_j}$ corresponds to the contribution of the ith functional requirement to the jth softgoal on the k (client or server) side: $\forall i \in \{1...N\}$, $\forall j \in \{1...M\}$, $\forall k \in \{C, S\}$: $W^k_{i_j} = w$, where N is the number of functional requirements and M is the number of softgoals, and w is defined as previously described.

C) Calculate the objective function. For each softgoal j the corresponding objective function F_j with respect to a decision vector X_v is calculated by summing the contributions of all requirements to each softgoal j taking into account the client/server distribution defined in x_v: $\forall j \in \{1, ..., M\}$, $\forall v \in \{0, ..., 2^N - 1\}$: $F_j(X_v) = \Sigma^M_{j=1} W^k_{i_j}$, where N is the number of FRs, M is the number of softgoals.

D) Calculate the overall fitness. The sum of all objective functions with respect to a decision vector X_v is computed to obtain the overall fitness of the decision vector X_v: $\forall j \in \{1...N\}$, $\forall v \in \{0, ..., 2^N - 1\}$: $\Sigma^M_{j=1} F_j(X_v)$, where N is the number of requirements and M is the number of softgoals.

E) Calculate the Pareto front. To do this, it is necessary to obtain all decision vectors that are not dominated by any other decision vectors, these forms the Pareto optimal set, while the corresponding objective vectors are said to form the Pareto front.

Step 3- Select the final client/server distribution. From the set of Pareto optimal client/server distributions, the designer now elects the distribution of choice. When making this selection, the designer needs to take into account the priorities of the stakeholder, i.e., which softgoals are most important to realize, and which negative impact on other softgoals is an acceptable consequence. The merit of our approach is that it gives the designer a good overview of the good possible configurations. Furthermore, by using the GORE approach, he also gets a good insight in the impact, both positive and negative, his decisions have on the realization of all softgoals, i.e. on the user satisfaction. Finally, the final RIA requirements model is created using the client/server distribution selected by the designer.

5 Case Study

In this section, we illustrate our approach by presenting an excerpt of a real-world case study of a bioinformatics company. The main goal of this company is to provide online services based on human genome analysis. The company will allow clients to upload a gene, subsequently analyze the gene and provide the user with personalized reports corresponding to the desire of the user. The first and only service that initially will be available is the reporting of potential illnesses the user is vulnerable to, based on the detection of variations of the user's supplied gene compared to the company's gene database. The company is committed to offer their services using a RIA, as they expect a modern Web application will be more attractive for visitors, and they want to study the potential benefit of distributing their functionality between the client and server side. In our excerpt of the case study, we focus (solely) on the reporting requirement to illustrate our approach.

Step 1.- Create the base RIA requirements model. According to our proposal, the first step is to create the initial (base) RIA requirements model. As no Web 1.0 was available, it is necessary to specify the base RIA requirements model from scratch. To do this, the designer has to specify the company's goals and requirements.

In Figure 3 the initial requirements model for this case study is shown, focusing solely on the reporting requirement. The goal of the WebApp (actor) is "Report to be provided". To achieve this goal, some FRs are needed: the navigational requirements "Provide variations by disease" and "Provide disease by variant", the layout requirement "Provide graphical view" and, the service requirements "Provide proprietary format document" and "Provide pdf". Some of these requirements directly or indirectly affect some softgoals, i.e., the service requirement "Provide pdf" affects directly the softgoal "Compatibility" by means of a "Help" contribution and also affects the "Security" softgoal indirectly by means of the goal "Report to be provided".

As explained in Sect. 4, at this point, we do not yet decide which FR will be implemented on the server or on the client side, and thus the contributions to the softgoals are initially labeled as "unknown", as their contribution depends on where they are implemented (e.g. client or server). For the "Compatibility" softgoal however, we see that the contributions are already stated, as those values do not depend on where the requirement is implemented. Contributions originating in "Provide graphical view" are also stated, because this requirement is mandatory implemented on the client, as it can only be realized using client-side technology.

Step 2.- Obtain the Pareto optimal client/server distribution. The aim is to obtain a set of good client/server (FR) distributions. To do this, the following sub-steps are needed.

A) Obtain a decision vector for requirements distribution. To calculate this distributions, we only need to consider those FRs that contribute to the softgoals in the base requirements model, as can be seen in Table 1. Consequently, there are sixteen possible configurations (decision vectors) for the

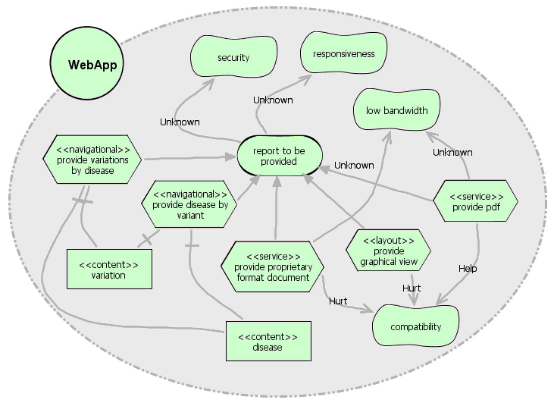

Fig. 3: Base RIA requirements model.

requirements distribution (using 2^4). Table 3 shows all possible decision vectors (column 2 to 5, all rows), in other words, all possible client/server configurations, where C represents Client and S represents Server. The requirement "Provide graphical view" is not considered as it is always allocated client side.

Table 1: *Softgoals* and FRs detected in the base RIA requirements model.

Softgoals	**Requirement**
S1.- security	R1.- provide variations by disease
S2.- responsiveness	R2.- provide disease by variant
S3.- compatibility	R3.- provide pdf
S4.- low bandwidth	R4.- provide proprietary format document

B) Create a weight matrix. Next, we create the weight matrices. For computing the objective functions in this case study, the following matrices are defined: Client matrix (Table 2a) and Server matrix (Table 2b). Each matrix contains the quantification of the contributions made by the FRs to each softgoal, both on the client and on the server side. In practice, the designer will take the base RIA requirements model, and concretize it twice: once filling in the contributions labels under the assumption that all FRs are located on the client, and once assuming they are all located on the server. The first model correspond to the Client matrix (Table 2a), the second to the Server matrix (Table 2b), using the weights defined in Section 4, 2B. As an example, row 3, (requirement "Provide pdf"), column 4 (softgoal "Responsiveness"), shows "+1" in the Client matrix, indicating a "help" contribution if the requirement is located on the client, and shows "-1" in the Server matrix, indicating a "hurt'" contribution if the requirement is located on the server side.

C) Calculate the objective function. The objective functions are calculated using the weight matrices from step 2.B, i.e., for the configuration X1 (Table 3): $F(S1) = -4$ is calculated summing the contributions made by the requirements to the softgoal S1 according to the client weight matrix (Table 2a) and $F(S2) = +4$ is calculated in the same form by using the server weight matrix

(Table 2b). For $F(S1) = -2$ from the configuration X2, the requirements R1, R2 and R3 are implemented in client side, so their contribution is taken from the client weight matrix (Table 2a): $\Sigma(R1, R2, R3) = -3$ and R4 is implemented in the server side, thus, $R4 = +1$ (taken from the server weight matrix (Table 2b), thus $F(S1) = -2$. For this case study, the results of the corresponding objective functions are shown in columns 6 to 9 from Table 3.

D) Calculate the overall fitness. All the objective functions are summed (Σ) with respect to each configuration (decision vector) to obtain the overall fitness (see column 10 from Table 3). For example, in the configuration X1 from Table 3, the overall fitness (Σ) is "+2", i.e. the sum of the softgoals $F(S1) + F(S2) + F(S3) + F(S4)$ for the decision vector X1.

Requirement	S1	S2	S3	S4
R1	-1	+1	0	0
R2	-1	+1	0	0
R3	-1	+1	+1	+1
R4	-1	+1	-1	+1

(a) Client weight matrix.

Requirement	S1	S2	S3	S4
R1	+1	-1	0	0
R2	+1	-1	0	0
R3	+1	-1	+1	-1
R4	+1	-1	-1	-1

(b) Server weight matrix.

Table 2: Weight matrices.

E) Calculate the Pareto front. The configurations (decision vectors) are Pareto front when they are not dominated by any other configurations. It is then said that these configurations form the Pareto optimal set, while the corresponding objective vectors (softgoals) are said to form the Pareto front. For example, the configuration X1 (decision vector X1) is Pareto front, because no other decision vector can be found that satisfies a particular softgoal better, without causing the satisfaction of other softgoals to decrease. Note that $F(S2)$, $F(S3)$ and $F(S4)$ are all maximized in this configuration. On the other hand, the configuration X2 is not Pareto front due to the configuration X5 that better satisfies $F(S4)$, and equally satisfies all other softgoals.

In Table 3, grey rows are the Pareto front from which we can select the final solution according to the priorities we establish over the softgoals. For example, the configuration X1, where every requirement is placed on the client side, is the best option if we do not mind that the satisfaction of softgoal S1 ("Security") is minimal (-4): all other softgoals are maximized, and the overall fitness is +2, indicating that overall, softgoals are well satisfied. On the other hand, if "Security" is the softgoal we want to prioritize, configuration X16, where every requirement is placed on the server side, is best suited. However, it comes with a high price with respect to other softgoals, and has an overall fitness of -2, indicated an overall poor satisfaction of softgoals. All the other solutions of the Pareto front are intermediate configurations that lead us to different tradeoffs such as X6 or X14. Both have an overall fitness of 0, indicating that overall, all softgoals are balanced.

Step 3.- Select the final client/server distribution. In this step, the final RIA requirements model is created using the client/server distribution selected by the designer. For the bioinformatics company, the configuration X14

Table 3: The possible functionality distribution among the server and the client by means of the softgoal tradeoff.

Configuration	R1	R2	R3	R4	F(S1)	F(S2)	F(S3)	F(S4)	Σ	Pareto Front
X1	C	C	C	C	-4	+4	0	+2	+2	Yes
X2	C	C	C	S	-2	+2	0	0	0	No because of X5
X3	C	C	S	C	-2	+2	0	0	0	No because of X5
X4	C	C	S	S	0	0	0	-2	-2	No because of X13
X5	C	S	C	C	-2	+2	0	+2	+2	Yes
X6	C	S	C	S	0	0	0	0	0	Yes
X7	C	S	S	C	0	0	0	0	0	Yes (as same as X6)
X8	C	S	S	S	+2	-2	0	-2	-2	No because of X14
X9	S	C	C	C	-2	+2	0	+2	+2	Yes (as same as X5)
X10	S	C	C	S	0	0	0	0	0	Yes (as same as X6)
X11	S	C	S	C	0	0	0	0	0	Yes (as same as X6)
X12	S	C	S	S	+2	-2	0	-2	-2	No because of X14
X13	S	S	C	C	0	0	0	+2	+2	Yes
X14	S	S	C	S	+2	-2	0	0	0	Yes
X15	S	S	S	C	+2	-2	0	0	0	Yes (as same as X14)
X16	S	S	S	S	+4	-4	0	-2	-2	Yes

(Figure 4) is selected as the final client/server distribution. In X14, the requirements R1, R2 and R4 are placed on the server and R2 on the client (see Table 1). The requirement "Provide graphical view" by default is always allocated in the client side (as mentioned in Step 2.A). This configuration was considered a good tradeoff, because security is deemed important, but other softgoals should not be totally neglected.

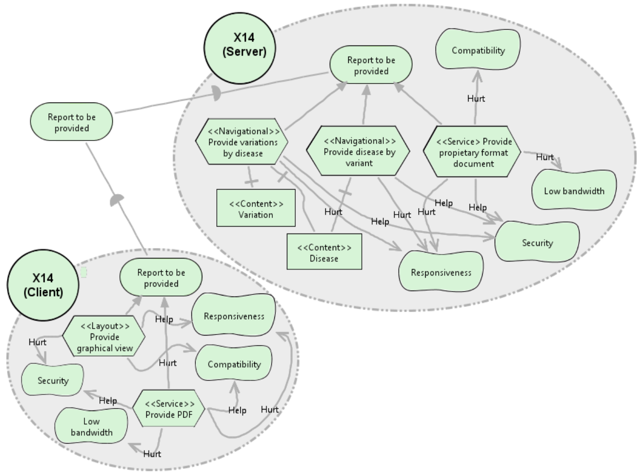

Fig. 4: X14 solution.

6 Implementation

Our approach for client/server distribution of RIA functionality has been implemented by using EMF (http://www.eclipse.org/emf/). EMF is a Java-based modeling framework and code generation facility for building tools and other applications based on a structured data model. In our case, the left-hand side of Fig. 5 shows a representation of the $i*$ requirements model of our case study by using the tree editor of EMF. The right-hand side shows how our Pareto-based approach has been implemented by means of such a tree editor. Furthermore, several Java classes have been developed to obtain a tree representation of the Pareto efficiency to be tailored to the $i*$ model that conforms to our metamodel for specifying Web requirements (previously described in Sect. 3): a *ParetoModel* represents all the solutions of our approach from a specific input requirements model (contained in the *ParetoModelRef*). Both, the *ParetoLink* and *ParetoLinkEnd* reference to each of the elements of this input model in order to be useful for computing the Pareto efficiency.

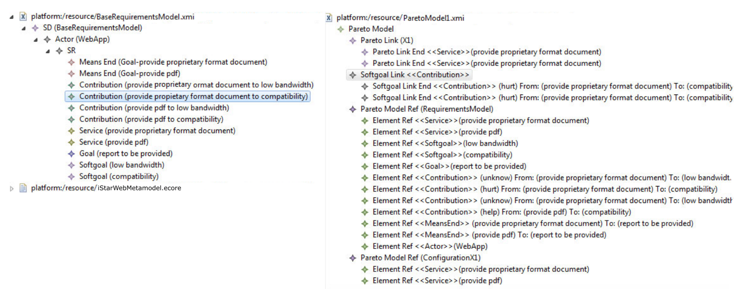

Fig. 5: The $i*$ base RIA requirements model (left) and the Pareto model (right).

7 Conclusions and Future Work

In this paper, we presented an extension to support RIAs to an existing $i*$-based goal-oriented requirements analysis method for Web 1.0 applications. Our approach goes out from the user's goals and expectations, and allows the designer to evaluate and decide on good client/server distributions for the functionality. To do so, we devised a set of steps based on the Pareto optimal approach that is particularly suited to balance and maximize the conflicting softgoals. Furthermore, it facilitates the evaluation of the obtained (Pareto) optimal solutions and selection of the final solution taking into account the priorities of softgoals. We exemplified our approach using an excerpt of a real-world case study we performed for a bio-informatics company. As future work, we plan to study and incorporate the data distribution between client and server in our Pareto algorithm. We would also like to integrate our approach in a RIA engineering method, and study the possibility to automatically generate design artifacts.

References

1. Aguilar, J.A., Garrigós, I., Mazón, J.N.: Impact Analysis of Goal-Oriented Requirements in Web Engineering. In: ICCSA. pp. 421–436. Springer, Santander, Spain (2011)

2. Aguilar, J.A., Garrigós, I., Mazón, J.N., Trujillo, J.: An MDA Approach for Goal-oriented Requirement Analysis in Web Engineering. Journal of Universal Computer Science (J. UCS) 16(17), 2475–2494 (2010)
3. Aguilar, J.A., Garrigós, I., Mazón, J.N., Trujillo, J.: Web Engineering Approaches for Requirement Analysis - A Systematic Literature Review. In: WEBIST. pp. 187–190 (2010)
4. Bozzon, A., Comai, S., Fraternali, P.: Current Research on the Design of Web 2.0 Applications Based on Model-Driven Approaches. ICWE pp. 25–31 (2008)
5. Collette, Y., Siarry, P.: Multiobjective optimization: principles and case studies. Springer (2003)
6. Cormode, G., Krishnamurthy, B.: Key differences between Web 1.0 and Web 2.0. First Monday 13(6), 2 (2008)
7. Dolog, P., Stage, J.: Designing interaction spaces for Rich Internet Applications with UML. In: Baresi, L., Fraternali, P., Houben, G.J. (eds.) Web Engineering. LNCS, vol. 4607, pp. 358–363. Springer (2007)
8. Horkoff, J., Yu, E.: Evaluating Goal Achievement in Enterprise Modeling–An Interactive Procedure and Experiences. The Practice of Enterprise Modeling pp. 145–160 (2009)
9. Koch, N., Pigerl, M., Zhang, G., Morozova, T.: Patterns for the model-based development of rias. In: ICWE. pp. 283–291. Springer, Berlin (2009)
10. Lowe, D.: Web system requirements: an overview. Requirements Engineering 8, 102–113 (2003)
11. Luna, E.R., Garrigós, I., Mazón, J.N., Trujillo, J., Rossi, G.: An i*-based Approach for Modeling and Testing Web Requirements. Journal of Web Engineering 9(4), 302–326 (2010)
12. Machado, L., Filho, O., Ribeiro, J.a.: UWE-R: an extension to a web engineering methodology for rich internet applications. WSEAS Trans. Info. Sci. and App. 6, 601–610 (April 2009)
13. Meliá, S., Gómez, J., Pérez, S., Díaz, O.: A model-driven development for GWT-based Rich Internet Applications with OOH4RIA. In: ICWE. pp. 13–23. IEEE (2008)
14. Preciado, J.C., Linaje, M., Sanchez, F., Comai, S.: Necessity of methodologies to model rich internet applications. In: WSE. pp. 7–13. IEEE, Washington, DC, USA (2005)
15. Preciado, J., Linaje, M., Comai, S., Sanchez-Figueroa, F.: Designing rich internet applications with web engineering methodologies. In: WSE. pp. 23–30. IEEE (Oct 2007)
16. Preciado, J., Linaje, M., Sanchez-Figueroa, F.: Adapting Web 1.0 User Interfaces to Web 2.0 Multidevice User Interfaces using RUX-Method. Journal of Universal Computer Science (J. UCS) 14(13), 2239–2254 (2008)
17. Sommerville, I.: Software Engineering. Addison-Wesley, 6th edn. (2001)
18. Urbieta, M., Rossi, G., Ginzburg, J., Schwabe, D.: Designing the interface of rich internet applications. In: LA-Web. pp. 144–153. IEEE (2007)
19. Valverde, F., Pastor, O.: Facing the Technological Challenges of Web 2.0: A RIA Model-Driven Engineering Approach. WISE pp. 131–144 (2009)
20. Wright, J.M., Dietrich, J.B.: Requirements for rich internet application design methodologies. In: WISE. pp. 106–119. Springer, Berlin (2008)
21. Wright, J.: A Modelling Language for Interactive Web Applications. In: ASE. pp. 689–692. IEEE (2010)
22. Yu, E.: Modelling Strategic Relationships for Process Reenginering. Ph.D. thesis, University of Toronto, Canada (1995)

References

1. R. Arnold and S. Bohner. 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 (IWWOST)*, 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/atl/`.
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. Abrah ao, 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 Reenginering.* PhD thesis, University of Toronto, Canada, 1995.