



Universidad Rey Juan Carlos
Escuela Técnica Superior de Ingeniería de Telecomunicación
Departamento de Lenguajes y Sistemas Informáticos

**APROXIMACIÓN MDA PARA EL DESARROLLO
ORIENTADO A SERVICIOS DE SISTEMAS DE
INFORMACIÓN WEB: DEL MODELO DE NEGOCIO AL
MODELO DE COMPOSICIÓN DE SERVICIOS WEB**

Memoria de Tesis Doctoral

Autora: **María Valeria de Castro**
Directora de Tesis: **Esperanza Marcos Martínez**

Móstoles (Madrid), Marzo de 2007

Detrás de cada línea de llegada, hay una de partida.

Detrás de cada logro, hay otro desafío.

Madre Teresa de Calcuta

RESUMEN

En los últimos años, las nuevas propuestas tecnológicas para la Web, tales como XML, servicios Web, automatización de procesos de negocios, B2B, etc., han favorecido el surgimiento de un nuevo paradigma para el desarrollo de aplicaciones conocido como *Computación Orientada a Servicios (Service Oriented Computing, SOC)* que propone la utilización de *servicios* como elementos básicos para la construcción de aplicaciones.

La computación orientada a servicios es, hoy en día, una de las principales líneas de investigación en el área de desarrollo de software, y ha promovido una evolución tanto en los Sistemas de Información (SI), como en la forma en que éstos pueden ser desarrollados. En la actualidad, la mayoría de las grandes empresas están tendiendo a sustituir sus sistemas, tradicionalmente centralizados, por redes de negocio en las que cada uno de los participantes provee a los demás de servicios especializados. Por ello, muchos de los SI actuales, principalmente los SI Web (SIW), se conciben como un medio para ofrecer servicios a través de Internet, involucrando a menudo simples o complejos procesos de negocios en los que colaboran varios participantes.

Como una consecuencia de este cambio en los SI, y de la misma manera que ha sucedido en su momento con la computación orientada a objetos y el posterior surgimiento de las metodologías orientadas a objetos, también en el campo de la ingeniería del software surge la necesidad de definir metodologías de desarrollo que faciliten la construcción de SI en base al nuevo paradigma SOC. De este modo, se permite a los desarrolladores de software aprovechar al máximo los beneficios de las nuevas tecnologías disponibles actualmente para este paradigma, tales como los *servicios Web*.

Para hacer frente a esta necesidad existen, en principio, dos opciones:

- La *adaptación de las metodologías de desarrollo de aplicaciones existentes*. Por ejemplo, realizando algunas modificaciones a sus propuestas de desarrollo originales para la representación de servicios Web o procesos de negocios. En general, dichas modificaciones sólo afectan a la interfaz de la aplicación.
- La *definición de un nuevo enfoque orientado a servicios* para el desarrollo de SI, teniendo en cuenta los *servicios* como elementos principales para la construcción de los SI.

La propuesta que se presenta en esta Tesis Doctoral, llamada SOD-M (*Service-Oriented Development Method*), está relacionada con la segunda opción y, a diferencia de otras aproximaciones existentes actualmente, define *un enfoque orientado a servicios para el desarrollo de SI*. Dicho enfoque se aplica aquí al desarrollo específico de SIW, aunque se presenta como un enfoque para el modelado de sistemas software en general, por lo que podría ser aplicado para el desarrollo de otros tipos de SI. Así, SOD-M propone el desarrollo de SIW en base a los servicios que se ofrecen a los usuarios o clientes a través de la Web. Dichos servicios, se identifican a partir del modelo de negocio de la organización, y se conciben como una composición, lógica y ordenada, de servicios más simples que son proporcionados por los distintos miembros que colaboran en los procesos de negocio de la organización.

Además, SOD-M se integra en una *arquitectura MDA (Model Driven Architecture)*, proporcionando todas las ventajas que ésta ofrece. SOD-M define un proceso dirigido por modelos que incluye: *modelos independientes de computación*, que permiten comprender los procesos de negocio de la organización y la identificación de los servicios que se ofrecen a los usuarios o consumidores del SIW; *modelos independientes de plataforma*, que permiten el modelado del sistema como un conjunto de servicios; y, dado que SOD-M surge con el objetivo de facilitar la implementación de los SIW utilizando las tecnologías disponibles para el paradigma SOC, *modelos específicos para plataformas basadas en servicios Web*. Además, SOD-M define *transformaciones entre los modelos de los distintos niveles de abstracción*, permitiendo de este modo, la alineación de procesos de negocio de alto nivel con tecnologías de la información basadas en el paradigma SOC.

SOD-M proporciona también, un *perfil UML* que define todos los elementos de modelado necesarios para la construcción de un SIW desde un enfoque orientado a servicios.

Como principales aportaciones de este trabajo de Tesis Doctoral se pueden destacar los siguientes puntos: la definición de un *nuevo enfoque orientado a servicios para el desarrollo completo de SIW*; la *integración del método propuesto en el marco de una arquitectura MDA* proporcionando todos los beneficios que ésta aporta; y, la definición de un *perfil UML* que incluye todos los elementos de modelado necesarios para la construcción de los modelos propuestos.

ABSTRACT

Over the last few years, new Web technological proposals such as XML, Web services, business process automation, B2B, etc., have favoured the emergence of a new paradigm for the development of applications. This is known as Service Oriented Computing (SOC) and it proposes the use of *services* as basic elements for the construction of applications.

SOC is, currently, one of the major research topics in the field of software development and has brought about an evolution in Information Systems (IS) themselves, as well as in how they can be developed. Currently, fully integrated enterprises are being replaced by business networks in which each participant provides the others with specialized services. So, many IS, mainly those for the Web, are created as a means of offering services over the Internet, usually involving simple or complex business processes in which various participants collaborate.

As a consequence of this change in IS and, as occurred with object-oriented computing in its day and with the later emergence of object-oriented methodologies, in the field of software engineering the need arises to define development methodologies that make the construction of IS based on the new SOC paradigm possible. This lets software developers get the maximum advantage from the benefits provided by the existing technologies available for this paradigm, such as Web services. To face this need there are two options, initially:

- *The adaptation of existing applications development methodologies.* This includes, for example, carrying out some modifications to their original development proposals for the representation of Web services or business processes. In general, these modifications only affect the interface of the application.
- *The definition of a new service-oriented approach for IS development,* taking into account the services as principal elements for the construction of the IS.

The proposal presented in this Doctoral Thesis, called SOD-M (*Service-Oriented Development Method*), is related to the second option and, in contrast with other existing approaches, defines *a service-oriented approach for the development of SI*. This approach is applied here to the specific development of Web IS (WIS), although it is introduced as an approach for modelling software systems in general, so it could be applied to the development of other IS types.

Thus, SOD-M proposes the development of WIS based on the services offered to users or clients by means of the Web. These services are identified from the business model of the organisation, and can be seen as a logical and ordered composition of simpler services provided by the different members that cooperate in the business processes of the organisation.

In addition, SOD-M is integrated into a *Model Driven Architecture*, providing all of the advantages offered by that. SOD-M defines a model driven process, that includes: *computational independent models*, which allow us to understand the business processes of the organisation and the identification of the services offered to users and consumers of the WIS, *platform independent models*, which permit the modelling of the system as a set of services and, finally, *platform specific models* for web services based technologies, given that SOD-M comes about with the aim of making it easier to implement the WIS by using the technologies available for the SOC paradigm. Moreover, SOD-M defines *mappings between the models of the different levels of abstraction*, thus allowing the alignment of high-level business processes with information technologies based on the SOC paradigm.

SOD-M also provides a *UML profile*. It is defined over behavioral modeling elements of UML 2.0 and defines all of the modelling elements needed for the construction of a WIS from a service-oriented approach.

The following points can be emphasized as main contributions of this Doctoral Thesis: the definition of *a new service-oriented approach for the complete development of WIS*; the *integration of the proposed method in the framework of an MDA architecture*, providing all the benefits offered by that; and the definition of a *UML profile* that includes all of the modelling elements needed for the construction of the proposed models.

AGRADECIMIENTOS

Quisiera agradecer en primer lugar y de manera muy especial a mi mentora y directora de tesis Esperanza Marcos, a quien agradezco en especial sus enseñanzas, su apoyo y constante motivación. Muchas Gracias Cuca, por haber confiado en mí más que yo misma, y sobre todo por demostrarme que siempre estás ahí. He aprendido muchísimo de ti y espero seguir haciéndolo.

Agradezco también a todas las personas que con sus comentarios han contribuido a la mejora de esta Tesis Doctoral. En especial a Roel Wieringa, por sus siempre acertados comentarios, y por darme la posibilidad de realizar parte de este trabajo de investigación en la Universidad de Twente en Holanda. Gracias a los miembros del Departamento de Ingeniería en Sistemas de Información de la Regional Resistencia de la Universidad Tecnológica Nacional en Argentina, por haberme invitado a realizar una estancia allí. Y gracias también a Francisco Curbera por sus importantes comentarios, de primera mano, sobre BPEL y el mundo de los Servicios Web.

Gracias a Juan Antonio Hernández y David Ríos por haber prestado sus aplicaciones para la validación de SOD-M; y, a Marcos, Jesús, Guillermo y a mis alumnos de proyectos fin de carrera por haber experimentado con el.

Muchísimas gracias a los “Kybelitos” (todos), porque es para mi un placer tenerlos como compañeros de trabajo, amigos y familia adoptiva ;-). Gracias a todos por su apoyo incondicional, por sus valiosos consejos, y por saber escuchar mis lamentos. Gracias también a todos por hacerse cargo de mis marroncillos durante estancias, y en especial durante el último tiempo de la Tesis. De manera particular, gracias a Paloma, Juancho, Marcos, Belén y César que han trabajado conmigo en parte de la investigación de este trabajo de Tesis Doctoral aportando lo mucho que saben; gracias a Josemari, por estar siempre disponible para ayudarme a resolver mis grandes problemas; y gracias también a los Kybelitos más nuevos, Carlos, Diana, Vero, Mariano y Sandra, por estar siempre dispuestos a echar una mano.

Gracias a mis amigos de Argentina, por su apoyo, y por sus constantes cuestionamientos acerca de qué estoy haciendo y para qué sirve todo esto, porque de alguna manera han ayudado a plantear el lado práctico de esta contribución.

A ti César te agradezco todo... en especial, y sólo por nombrar algo, gracias por ser mi gran apoyo en la vida, por el amor que me demuestras día a día, y

también por ser un excelente amigo y compañero de trabajo. Gracias también por tu enorme apoyo y comprensión durante los últimos momentos de la Tesis.

Finalmente, agradezco de todo corazón a mi familia, a la que amo y extraño muchísimo: mis hermanos, mis adorables sobrinitos, mis abuelos, tíos, primos; pero fundamentalmente a mis padres, por su gran amor y por hacerme sentir que están tan cerca, gracias a los dos porque sin sus constantes muestras de apoyo y cariño nada de esto hubiera sido posible.

Gracias, Gracias, Gracias... Mil Gracias a todos!!!

ÍNDICE GENERAL

1	INTRODUCCIÓN.....	3
1.1	PLANTEAMIENTO Y JUSTIFICACIÓN DEL TRABAJO	3
1.2	HIPÓTESIS Y OBJETIVOS.....	10
1.3	MARCO DE TRABAJO.....	12
1.3.1	<i>Proyectos de Investigación</i>	<i>13</i>
1.3.2	<i>Estancias en Centros Extranjeros</i>	<i>14</i>
1.4	MÉTODO DE INVESTIGACIÓN	15
1.4.1	<i>Método de Resolución y Validación</i>	<i>16</i>
1.5	ORGANIZACIÓN DE LA MEMORIA.....	21
2	ESTADO DEL ARTE	25
2.1	MODEL DRIVEN ARCHITECTURE – MDA.....	25
2.1.1	<i>Los Modelos en MDA.....</i>	<i>26</i>
2.1.2	<i>Transformaciones de modelos en MDA.....</i>	<i>27</i>
2.2	METODOLOGÍAS PARA EL DESARROLLO DE SIW.....	29
2.2.1	<i>OOHDM: Object Oriented Hypermedia Design Method.....</i>	<i>32</i>
2.2.2	<i>UWE: UML Based Web Engineering.....</i>	<i>34</i>
2.2.3	<i>WebML: Web Modeling Language</i>	<i>36</i>
2.2.4	<i>OOWS: Object Oriented Web-Solutions.....</i>	<i>38</i>
2.2.5	<i>OO-H: Object-Oriented Hipermedia Method.....</i>	<i>39</i>
2.2.6	<i>WSDM: Web Site Design Method.....</i>	<i>41</i>
2.2.7	<i>Autoweb: Una metodología para el desarrollo de aplicaciones Web</i>	<i>42</i>
2.2.8	<i>W2000: Un marco para el diseño de aplicaciones Web</i>	<i>43</i>
2.2.9	<i>Otras metodologías de Desarrollo de SIW.....</i>	<i>44</i>
2.2.10	<i>Comparación de Metodologías para el Desarrollo de SIW.....</i>	<i>45</i>
2.3	PROPUESTAS PARA EL DESARROLLO ORIENTADO A SERVICIOS	50
2.3.1	<i>Método e³-value.....</i>	<i>51</i>
2.3.2	<i>Método de modelado WIED</i>	<i>52</i>
2.3.3	<i>Business Process Modeling Notation - BPMN</i>	<i>52</i>
2.3.4	<i>Perfil UML para Business Process Execution Language - UML4BPEL....</i>	<i>53</i>
2.3.5	<i>Perfil UML para Web Services Description Languages - UML4WSDL....</i>	<i>54</i>
2.3.6	<i>Método para el desarrollo dirigido por modelos de servicios Web - MDWSD (Model-Driven Web Service Development).....</i>	<i>54</i>
2.3.7	<i>Service Components: Un marco para el desarrollo de la composición de servicios Web.....</i>	<i>55</i>
2.3.8	<i>Resumen de las principales propuestas relacionadas con el desarrollo orientado a servicios</i>	<i>56</i>

2.4	CONCLUSIONES DEL CAPÍTULO	58
3	SOD-M: UNA APROXIMACIÓN MDA PARA EL DESARROLLO ORIENTADO	
	A SERVICIOS DE SIW	63
3.1	CARACTERÍSTICAS PRINCIPALES Y OBJETIVOS DE SOD-M	63
3.2	FUNDAMENTOS DE SOD-M	65
3.3	SOD-M EN EL MARCO DE MDA	71
3.4	MODELOS DE SOD-M	74
3.4.1	<i>Modelos Independientes de Computación</i>	<i>74</i>
3.4.1.1	Modelo de Valor	75
3.4.1.2	Modelo de Proceso de Negocio	78
3.4.2	<i>Modelos Independientes de Plataforma</i>	<i>78</i>
3.4.2.1	Modelo de Casos de Uso.....	79
3.4.2.2	Modelo de Casos de Uso Extendido	80
3.4.2.3	Modelo de Proceso de Servicio	82
3.4.2.4	Modelo de Composición de Servicio	83
3.4.3	<i>Modelos Específicos de Plataforma.....</i>	<i>85</i>
3.4.3.1	Modelo de Interfaz de Servicio Web	86
3.4.3.2	Modelo de Composición de Servicio Extendido	91
3.5	PERFIL UML SOD-M	93
3.6	PROCESO DE SOD-M	95
3.6.1	<i>Actividades de Generación de Modelos.....</i>	<i>97</i>
3.6.1.1	Generación del Modelo de Valor	98
3.6.1.2	Generación del Modelo de Proceso de Negocio	99
3.6.1.3	Generación de la Lista de Servicios de Negocio	99
3.6.1.4	Generación del Modelo de Casos de Uso	100
3.6.1.5	Generación del Modelo de Casos de Uso Extendido	101
3.6.1.6	Generación del Modelo de Proceso de Servicio	103
3.6.1.7	Generación del Modelo de Composición de Servicio	103
3.6.1.8	Generación del Modelo de Composición de Servicio Extendido..	104
3.6.1.9	Generación del Modelo de Interfaz de Servicio Web	105
3.6.2	<i>Transformación de Modelos en SOD-M.....</i>	<i>107</i>
3.6.2.1	Guías de Transformación.....	107
3.6.2.2	Reglas de Transformación: Lenguaje Natural	108
3.6.2.3	Reglas de Transformación: Gramáticas de Grafos	111
3.7	INTEGRACIÓN DE SOD-M EN LA ARQUITECTURA DE MIDAS	113
3.7.1	<i>Modelado del Hipertexto de MIDAS.....</i>	<i>114</i>
3.7.1.1	Modelo de Fragmentos Extendido	115
3.7.1.2	Modelo de Navegación Extendido	116

3.7.2	<i>Obtención de Modelos del Hipertexto a partir de Modelos Propuestos por SOD-M.....</i>	<i>117</i>
4	VALIDACIÓN.....	125
4.1	MÉTODO DE VALIDACIÓN	125
4.2	DESCRIPCIÓN DE LOS CASOS DE ESTUDIO DESARROLLADOS	130
4.2.1	<i>Casos de Estudio de Laboratorio</i>	<i>130</i>
4.2.2	<i>Casos de Estudio Reales.....</i>	<i>132</i>
4.3	APLICACIÓN A UN CASO DE ESTUDIO REAL	136
4.3.1	<i>Descripción del SIW GesIMED.....</i>	<i>137</i>
4.3.2	<i>CIM: Modelado de Negocio</i>	<i>139</i>
4.3.3	<i>PIM: Modelado del Comportamiento</i>	<i>141</i>
4.3.4	<i>PSM: Modelado del Comportamiento.....</i>	<i>146</i>
4.3.5	<i>Implementación.....</i>	<i>148</i>
5	CONCLUSIONES Y TRABAJOS FUTUROS	155
5.1	ANÁLISIS DE LA CONSECUCCIÓN DE OBJETIVOS	155
5.2	PRINCIPALES APORTACIONES	158
5.3	CONTRASTACIÓN DE RESULTADOS	160
5.4	LÍNEAS DE INVESTIGACIÓN ABIERTAS	165
	BIBLIOGRAFÍA	171
	LUGARES DE INTERNET	183
A	APÉNDICE A: SIGLAS.....	187
B	APÉNDICE B: CASOS DE ESTUDIO.....	193
	<i>Caso 0: Servicio Web de Información de Vuelos.....</i>	<i>193</i>
	<i>Caso 1: WebConference: Sistema para la Gestión del Envío de Trabajos a Conferencias</i>	<i>195</i>
	<i>Caso 3: PARBUD: Sistema para la Gestión de Presupuestos Participativos ...</i>	<i>198</i>
	<i>Caso 4: Atento: Sistema de Ventanilla Única para la Administración Pública.</i>	<i>202</i>
	<i>Caso 5: ISLA: Sistema para la Gestión de una Autoescuela.....</i>	<i>207</i>
C	APÉNDICE C: REGLAS DE TRANSFORMACIÓN	219

ÍNDICE DE FIGURAS

Figura 1.1. Arquitectura de MIDAS.....	10
Figura 1.2. Marco de Trabajo de la Tesis Doctoral	13
Figura 1.3. Método de Investigación.....	16
Figura 1.4. Etapa de Resolución y Validación	18
Figura 1.5. Aplicación del método de Investigación en Acción	21
Figura 2.1. Metamodelo de MDA	28
Figura 3.1. Conceptos de SOD-M	66
Figura 3.2. Modelos de SOD-M en el Marco de MDA	73
Figura 3.3. Metamodelo del modelo de valor de e ³ value (Gordijn y Akkermans, 2003)	76
Figura 3.4. Metamodelo de Casos de Uso.....	80
Figura 3.5. Metamodelo de Casos de Uso Extendido.....	81
Figura 3.6. Metamodelo de Proceso de Servicio	83
Figura 3.7. Metamodelo de Composición de Servicio	84
Figura 3.8. Metamodelo del estándar WSDL Versión 2.0 Core Language	87
Figura 3.9. Metamodelo de Interfaz de Servicio Web	88
Figura 3.10. Metamodelo de Composición de Servicio Extendido	92
Figura 3.11. Dependencias de los elementos de modelado de SOD-M respecto de los paquetes UML para modelado del comportamiento y modelado estructural	94
Figura 3.12. Perfil UML SOD-M	95
Figura 3.13. Proceso de SOD-M	96
Figura 3.14. Transformación de <i>Servicios de Negocios y Consumidor Final</i> del metamodelo de Casos de Uso a <i>Casos de Uso Básicos y Compuestos</i> en el metamodelo de Casos de Uso Extendido.....	113
Figura 3.15. Arquitectura de Modelos de MIDAS	114
Figura 3.16. Proceso para el Modelado del Hipertexto de MIDAS.....	117
Figura 3.17. Transformación de los <i>Casos de Uso Básicos</i> del metamodelo de Casos de Uso Extendido al metamodelo de Fragmentos Extendido	121
Figura 3.18. Transformación de los atributos de las clases del metamodelo de Datos Conceptual al metamodelo de Fragmentos Extendido	121
Figura 4.1. Proceso de Validación de este trabajo de Tesis Doctoral	126
Figura 4.2. Escenario del SIW GesIMED	138
Figura 4.3. Modelo de valor para el caso GesIMED	139
Figura 4.4. Modelo de proceso de negocio para el caso GesIMED	140
Figura 4.5. Modelo de casos de uso del SIW GesIMED	142

Figura 4.6. Modelo de casos de uso extendido para el servicio de negocio "realizar procesamiento de imágenes"	143
Figura 4.7. Modelo de casos de uso extendido para el servicio de negocio "realizar visualizaciones de imágenes"	143
Figura 4.8. Modelo de casos de uso extendido para el servicio de negocio "realizar consultas"	143
Figura 4.9. Modelo de proceso de servicio para el servicio de negocio "realizar procesamiento de imágenes"	144
Figura 4.10. Modelo de proceso de servicio para el servicio de negocio "realizar visualizaciones de imágenes"	144
Figura 4.11. Modelo de proceso de servicio para el servicio de negocio "realizar consultas"	144
Figura 4.12. Modelo de composición de servicio para el servicio de negocio "realizar procesamiento de imágenes"	145
Figura 4.13. Modelo de composición de servicio para el servicio de negocio "realizar visualizaciones de imágenes"	145
Figura 4.14. Modelo de composición de servicio para el servicio de negocio "realizar consultas"	146
Figura 4.15. Modelo de composición de servicio extendido para el servicio de negocio "realizar procesamiento de imágenes"	146
Figura 4.16. Modelo de composición de servicio extendido para el servicio de negocio "realizar visualizaciones de imágenes"	147
Figura 4.17. Modelo de composición de servicio extendido para el servicio de negocio "realizar consultas"	147
Figura 4.18. Modelo de interfaz de servicio Web para el servicio Web "validar tarjeta de crédito"	148
Figura 4.19. Arquitectura el SIW GesIMED	148
Figura 4.20. Interfaz del SIW GesIMED	149
Figura B.1. Modelo de interfaz del servicio Web "ServicioVuelo"	194
Figura B.2. Representación de la interfaz del servicio Web "ServicioVuelo" en WSDL 1.2	194
Figura B.3. Modelo de casos de uso del SIW WebConference	196
Figura B.4. Modelo de casos de uso extendido del SIW WebConference	196
Figura B.5. Modelo de proceso de servicio para los servicios de negocio "enviar un artículo", "editar los datos del autor" y "ver los artículos enviados"	197
Figura B.6. Modelo de fragmentos extendido para del SIW WebConference	197
Figura B.7. Modelo de navegación extendido para del SIW WebConference	198

Figura B.8. Modelo de proceso de negocio para la elaboración de presupuestos participativos	199
Figura B.9. Modelo de casos de uso para la etapa de “comunicación de preferencias” del SIW PARBUD	200
Figura B.10. Modelo de fragmentos extendido para el servicio de negocio “comunicación de preferencias” del SIW PARBUD	201
Figura B.11. Modelo de navegación extendido para el servicio de negocio “comunicación de preferencias” del SIW PARDUB	201
Figura B.12. Arquitectura del sistema Atento.....	202
Figura B.13. Modelo de casos de uso del SIW Atento.....	204
Figura B.14. Modelo de casos de uso extendido para el servicio de negocio “grabar interacción de un ciudadano” del SIW Atento	204
Figura B.15. Modelo de proceso de servicio para el servicio de negocio “grabar interacción de un ciudadano” del SIW Atento	205
Figura B.16. Modelo de composición de servicio extendido para el servicio de negocio “grabar interacción de un ciudadano” del SIW Atento	205
Figura B.17. Modelo de fragmentos extendido parcial para del SIW Atento	206
Figura B.18. Modelo de navegación extendido parcial para del SIW Atento.....	207
Figura B.19. Modelo de valor para la Autoescuela ISLA.....	209
Figura B.20. Modelo de valor: necesidad del alumno de obtener el carné de conducir	210
Figura B.21. Modelo de valor: necesidad del alumno de tomar clases prácticas...	210
Figura B.22. Modelo de valor: necesidad del alumno de comprar material.....	211
Figura B.23. Modelo de valor: necesidad del alumno de realizar un curso de reciclaje	211
Figura B.24. Modelo de casos de uso del SIW ISLA	212
Figura B.25. Modelo de casos de uso extendido para el servicio de negocio “obtener permiso de conducir”	213
Figura B.26. Modelo de casos de uso extendido para el servicio de negocio “tomar clases prácticas”	214
Figura B.27. Modelo de casos de uso extendido para el servicio de negocio “comprar material”	214
Figura B.28. Modelo de casos de uso extendido para el servicio de negocio “recuperar permiso de conducir”	215
Figura C.1. Transformación de <i>Servicios de Negocios y Consumidor Final</i> del metamodelo de Casos de Uso a <i>Casos de Uso Básicos y Compuestos</i> en el metamodelo de Casos de Uso Extendido.....	219

Figura C.2. Transformación de relaciones <i>extend</i> del metamodelo de Casos de Uso Extendido al metamodelo de Procesos de Servicio	220
Figura C.3. Transformación de relaciones <i>extend</i> (con varios orígenes) del metamodelo de Casos de Uso Extendido al metamodelo de Procesos de Servicio	220
Figura C.4. Transformación de relaciones <i>include</i> del metamodelo de Casos de Uso Extendido al metamodelo de Procesos de Servicio	221
Figura C.5. Transformación de relaciones <i>include</i> (con varios destinos) del metamodelo de Casos de Uso Extendido al metamodelo de Procesos de Servicio	221
Figura C.6. Transformación de <i>Actividades de Servicios</i> del metamodelo de Procesos de Servicio al metamodelo de Composición de Servicio	222
Figura C.7. Transformación de <i>Acciones</i> del metamodelo de Composición de Servicio a <i>Servicios Web</i> en el metamodelo de Composición de Servicio Extendido	222
Figura C.8. Transformación de <i>Servicios Web</i> del metamodelo de Composición de Servicio Extendido a elementos <i>Operation</i> en el metamodelo de Interfaz de Servicio Web.....	223
Figura C.9. Transformación de <i>Servicios Web</i> del metamodelo de Composición de Servicio Extendido a elementos <i>Message</i> en el metamodelo de Interfaz de Servicio Web.....	223

ÍNDICE DE TABLAS

Tabla 1.1. Casos de estudios utilizados en la validación de la Tesis Doctoral.....	20
Tabla 2.1. Resumen de las principales propuestas para el Desarrollo de SIW.....	49
Tabla 2.2. Resumen de propuestas relacionadas con el Desarrollo Orientado a Servicios	57
Tabla 2.3. Características Generales de SOD-M	59
Tabla 3.1. Conceptos, semántica y notación de los elementos del Modelo de Valor.	76
Tabla 3.2. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Casos de Uso	80
Tabla 3.3. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Casos de Uso Extendido	81
Tabla 3.4. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Proceso de Servicio	83
Tabla 3.5. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Composición de Servicio.....	85
Tabla 3.6. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Interfaz de Servicios Web.....	89
Tabla 3.7. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Composición de Servicio Extendido	92
Tabla 3.8. Guías de Transformación en SOD-M.	108
Tabla 3.9. Reglas de Transformación en SOD-M	109
Tabla 3.10. Reglas de transformación de las relaciones <i>extend</i> e <i>include</i> del Modelo de Casos de Uso Extendido al Modelo de Proceso de Servicio	110
Tabla 3.11. Reglas de Transformación entre los modelo de SOD-M y modelos del hipertexto.....	119
Tabla 4.1. Lista de servicios de negocio para el caso GesIMED	141
Tabla B.1. Lista de servicios de negocio para la Autoescuela ISLA.....	212

INTRODUCCIÓN

La presente Tesis Doctoral aborda la problemática del desarrollo orientado a servicios en el caso específico de Sistemas de Información Web. En este capítulo se presenta, a modo de introducción al trabajo de investigación que se ha llevado a cabo: el planteamiento y la justificación del trabajo; la hipótesis y objetivos; el marco de trabajo; el método de investigación; y, por último, la organización de esta memoria de Tesis Doctoral. Así, en la sección 1.1, se plantea el problema abordado y se justifica la necesidad de métodos específicos para el desarrollo orientado a servicios de SIW, describiendo además las ventajas de que dichos métodos se definan como parte de una arquitectura dirigida por modelos. En la sección 1.2 se presenta la hipótesis planteada al inicio de este trabajo de investigación, así como el objetivo principal y derivados del mismo. En la sección 1.3 se presenta el marco de trabajo, describiendo los principales proyectos de investigación en los que se integra y las estancias realizadas en centros extranjeros. En la sección 1.4 se presenta el método de investigación utilizado. Por último, en la sección 1.5 se describe la organización de los restantes capítulos de esta memoria de Tesis.

1.1 Planteamiento y Justificación del Trabajo

En los últimos años, las nuevas propuestas tecnológicas para la Web, tales como XML, servicios Web, automatización de procesos de negocios, B2B, etc., han favorecido el surgimiento de un nuevo paradigma para el desarrollo de aplicaciones conocido como **Computación Orientada a Servicios** (SOC, *Service Oriented Computing*) (Papazoglou y Georgakopoulos, 2003). SOC utiliza *servicios* como elementos para el desarrollo de aplicaciones, basándose para ello en una arquitectura orientada a servicios (SOA, *Service Oriented Architecture*).

Este nuevo paradigma en el desarrollo de aplicaciones, ha impulsado una evolución en los sistemas de información (SI) y en la forma en que éstos pueden ser desarrollados, principalmente, en lo que respecta al ámbito de los SI Web (SIW). Así, mientras que los primeros SIW se concebían como un medio para ofrecer información a los usuarios, y se construían fundamentalmente en base a páginas estáticas o dinámicas e hiperenlaces entre ellas, en la actualidad, muchos de los SIW se conciben como un medio para ofrecer servicios a través de Internet, y la mayoría de las grandes empresas están sustituyendo sus sistemas completamente centralizados por redes de negocio, en las que cada participante provee a los demás de servicios especializados (Curbera *et al.*, 2003). Los servicios

más comúnmente ofrecidos en la Web van desde aquellos más simples, como la posibilidad de comprar un libro, un billete de avión o reservar un coche, hasta aquellos que involucran procesos complejos, en los que pueden participar varias entidades, como obtener cotizaciones para una compra o participar en una subasta on-line.

Debido a este cambio de enfoque en el desarrollo de aplicaciones, y de la misma manera que sucedió en su momento con otros paradigmas de computación como el orientado a objetos, también en el campo de la ingeniería del software surge la necesidad de definir metodologías específicas que soporten el proceso de desarrollo de software en base a este nuevo paradigma.

Analizando la literatura actual relacionada con el desarrollo orientado a servicios, se puede comprobar que la mayoría de los autores tratan este tema desde un punto de vista tecnológico o de implementación (Curbera *et al.*, 2003; Papazoglou y Georgakopoulos, 2003; Bichler y Lin, 2006). Es decir, centrándose en las tecnologías disponibles para el desarrollo de aplicaciones basadas en servicios Web. Sin embargo, y a pesar del impacto que la computación orientada a servicios ha producido en el desarrollo de software y en cuanto a la manera en que los sistemas pueden ser contruidos, actualmente, no existen metodologías que faciliten el desarrollo de software en base a este nuevo paradigma.

Para hacer frente a esta necesidad, es decir, a la definición de metodologías que le permitan a los desarrolladores de software aprovechar al máximo los beneficios del paradigma de computación orientado a servicios, y, centrándonos ya en el ámbito específico de los SIW (ámbito en el que se aplica el método que se propone en esta Tesis Doctoral) se puede hablar de dos alternativas:

- La *adaptación de las metodologías de desarrollo de SIW existentes*. Por ejemplo, extendiendo sus modelos para dar soporte al desarrollo de nuevos conceptos tales como servicios Web, procesos de negocios, etc.
- La *definición de un nuevo enfoque orientado a servicios* para el desarrollo de SIW, que permita a los desarrolladores de software concebir un SIW como un conjunto de *servicios*.

Algunas de las metodologías más relevantes para el desarrollo de SIW, tales como UWE, WebML o OO-HDM han optado por la primera de las alternativas. Así, por ejemplo, han surgido extensiones para la representación de procesos de negocios en SIW, tales como las extensiones a WebML (Brambilla *et al.*, 2003a; Brambilla *et al.*, 2006), a UWE y OO-H (Koch *et al.*, 2004), a OOHDM (Schmid y

Rossi, 2004) y la extensión de WSDM (De Troyer y Casteleyn, 2003); como también nuevas extensiones para la representación de SIW que interactúan con servicios Web, como la extensión a WebML (Brambilla *et al.*, 2003b; Manolescu *et al.*, 2005).

Sin embargo, la computación orientada a servicios, como todo nuevo paradigma de computación, no sólo se limita a la utilización de servicios Web en el desarrollo de aplicaciones Web o a la representación de procesos de negocios en la lógica de la aplicación, sino que provee nuevas facilidades y nuevos retos en el desarrollo de este tipo de aplicaciones. Mike Papazoglou, uno de los impulsores de la computación orientada a servicios, destaca la importancia e implicaciones de este nuevo paradigma de computación, a la vez que afirma que “la ingeniería y desarrollo orientado a servicios, tales como el análisis orientado a servicios y las metodologías o técnicas de diseño y desarrollo orientados a servicios, son elementos cruciales para el desarrollo de servicios y especificaciones de procesos de negocios significativas” (Papazoglou *et al.*, 2006).

Así, atendiendo a este nuevo paradigma y las necesidades que plantea, en este trabajo de Tesis Doctoral se propone un *nuevo enfoque orientado a servicios* para el desarrollo de SIW. Dicho enfoque permite diferenciar al método propuesto de otras aproximaciones existentes actualmente para el desarrollo de SIW. Cabe aclarar en este punto, que el enfoque orientado a servicios que aquí se presenta se define como un enfoque para el modelado de sistemas software en general, aunque, en el caso concreto de esta Tesis Doctoral, se aplica al desarrollo específico de **sistemas de información para la Web**.

Para dar una definición más precisa sobre qué significa un *enfoque orientado a servicios*, es posible apoyarse en las dos formas más tradicionales de enfocar el modelado de sistemas para el desarrollo de software: el *enfoque estructurado*, en el que el bloque principal de construcción de todo del software es el procedimiento o función; y el *enfoque orientado a objetos*, en el que el bloque de construcción de todo el software es el objeto o clase (Booch *et al.*, 1999). Así, teniendo en cuenta estas definiciones y la idea central del paradigma de computación SOC, en el marco de esta Tesis Doctoral se define un **enfoque orientado a servicios** como *un enfoque para el modelado de sistema que se basa en la identificación de servicios como bloque principal para la construcción del software*.

Un **servicio** se define como “*un recurso abstracto que representa una capacidad de realización de tareas que forman una funcionalidad coherente desde el punto de vista de las entidades proveedoras y de las entidades solicitantes*”

(W3C, 2003a). Por tanto, cuando se habla de servicio, se habla de una funcionalidad, que es ofrecida por una entidad y que satisface una necesidad de una entidad solicitante, y no de la forma en que se implementa una funcionalidad en el sistema.

Según afirma (Papazoglou *et al.*, 2006), “una **metodología de desarrollo orientado a servicios** se centra en el desarrollo de procesos de negocios, los cuales se consideran elementos reutilizables que son independientes de las aplicaciones y de la plataforma en la que se implementen los mismos. Así, es posible concebir las aplicaciones como un conjunto de servicios, que se conectan a través de interfaces bien definidas, en el marco de una arquitectura orientada a servicios”. Enfoques tradicionales, como el desarrollo orientado a objetos y el desarrollo basado en componentes, pueden ser aplicados por desarrolladores de software para la construcción de aplicaciones basadas en servicios. Sin embargo, metodologías como las orientadas a objetos o a componentes sólo pueden abordar el desarrollo de parte de los requisitos de las mismas (Papazoglou *et al.*, 2006). Esto se debe, principalmente, al hecho de que dichas metodologías no han sido definidas para abordar el desarrollo de elementos propios de la computación orientada a servicios, tales como los servicios o la composición de servicios. Así, por ejemplo, mientras que el diseño y la construcción de un servicio simple se pueden llevar a cabo de un modo sencillo y eficaz, la construcción de procesos de negocios reutilizables que componen varios servicios tanto externos como internos puede tornarse más complejo. Además, otra de las razones que justifica la necesidad de un nuevo enfoque, es la existencia de un nuevo paradigma de desarrollo. En el caso de las metodologías orientadas a objetos se propone la construcción del sistema en base a *objetos*, es decir “cosas”, del mundo real; mientras que en el caso de las metodologías orientadas a servicios se propone la construcción del sistema en base a *servicios*, es decir “tareas o actividades que realizamos” en el mundo real. Por otro lado, en el caso de las metodologías de desarrollo basadas en componentes, las diferencias residen fundamentalmente, en que los componentes y los servicios no son lo mismo, puesto que presentan varias diferencias en cuanto al tipo de comunicación, el tipo de acoplamiento, el tipo de invocación, el tipo de localización, etc. (Papazoglou *et al.*, 2006).

De la misma manera que el desarrollo orientado a objetos define un conjunto de técnicas o modelos específicos para la construcción de los sistemas en base a objetos, una metodología para el desarrollo orientado a servicios debe proporcionar ciertas técnicas o modelos que permitan la construcción de los sistemas en base a servicios, tales como:

- Técnicas de modelado de alto nivel que permitan *comprender el modelo de negocio de una organización*, proporcionando medios para la identificación de los servicios que se ofrecen (quien los ofrece y quien/es lo necesitan), así como de las responsabilidades de cada uno de los participantes del negocio en el ofrecimiento de dichos servicios.
- Técnicas que permitan el *modelado de servicios*, tanto simples (por ejemplo, una consulta a una base de datos o una validación de usuario) como compuestos, es decir, aquellos que se obtienen en base a la composición de dos o más servicios simples.
- Técnicas que permitan el *modelado de procesos para la composición de servicios*, es decir, que permitan representar el orden o flujo de trabajo necesario para llevar a cabo un servicio compuesto.

La utilización de este tipo de técnicas en el desarrollo de sistemas, facilita la construcción de los mismos utilizando las tecnologías disponibles actualmente para el paradigma SOC, tales como los *servicios Web*.

Los **servicios Web** juegan un papel preponderante en el paradigma SOC y proveen un marco para la aplicación basada en estándares de este paradigma (Curbera *et al.*, 2003). Los servicios Web proporcionan un marco de trabajo sistemático y extensible, basado en XML y construido en base a protocolos Web existentes, que permite la interacción de aplicaciones a través de la Web. Dichos servicios definen mecanismos estandarizados para describir, publicar/localizar e interactuar con aplicaciones en línea (Curbera *et al.*, 2002).

Pero más allá de la idea básica de “describir, publicar, interactuar” es necesario contar con mecanismos para la **composición de servicios Web**. Por este motivo, en los últimos años han surgido numerosas especificaciones en este área, tales como: XLANG (Microsoft, 2001), WSFL (*Web Services Flow Language*) (IBM, 2001) y BPEL4WS (*Business Process Execution Language for Web Services*) (Andrews *et al.*, 2003) que combina las características de los dos primeros y que se ha convertido en el lenguaje estándar más aceptado para la composición de servicios Web. Además, han surgido otras propuestas como WS-Coordination (*Web Services coordination*) (IBM, 2005a) y WS-Transaction (*Web Services transactions*) (IBM, 2005b) para el soporte de interacciones entre servicios Web más robustos.

De acuerdo con (Curbera, 2005), tradicionalmente se consideran dos modelos de composición, *orientada a procesos* y *distribuida o estructural*:

- La *composición orientada a procesos* (también conocida como *orquestración*) sigue un modelo de flujo de trabajo para la composición de servicios Web. En este caso, la composición de servicios Web está centralizada, en el sentido de que su ejecución está bajo el control de un motor de flujo de trabajo. Esta forma de composición es la más extendida. El lenguaje BPEL4WS es el ejemplo prototipo de este tipo de composición de servicios.
- La *composición distribuida o estructural* (también llamada *coreografía*) considera la integración de servicios desde una perspectiva distribuida. Se enfoca en el establecimiento y la documentación de las relaciones entre los servicios Web que participan en una interacción entre muchas partes. Este modelo de composición tiene su aplicación en transacciones B2B (*Business-to-Business*) múltiples, donde los roles de cada servicio Web participante necesitan estar documentados formalmente. Este tipo de composición es el utilizado por el proyecto de coreografía de servicios Web (Web Services Choreography) que se lleva a cabo en el W3C (W3C, 2002).

La orquestración es la opción más flexible para la implementación de procesos de negocio en SIW. En este caso, dado que se conoce exactamente quién es el responsable de la ejecución de todo el proceso de negocio, se pueden incorporar servicios Web incluso cuando éstos no estén advertidos de que son parte del proceso de negocio, y además, se pueden proveer escenarios alternativos en caso de error.

Ahora bien, como se ha dicho anteriormente, aunque el diseño e implementación de un servicio Web parece una tarea bastante simple, no se puede decir lo mismo del diseño y la implementación de los procesos de negocio. Los lenguajes para la implementación de procesos de negocio tienen bastantes limitaciones a la hora de ser utilizados en etapas tempranas del proceso de desarrollo de software (Verner, 2004). Esto se debe, principalmente, al hecho de que pasar del modelado de procesos de negocios de alto nivel a un lenguaje de composición, que implemente dichos procesos a través de servicios Web, no es una tarea sencilla. Su dificultad radica en que el modelado de negocio es realizado generalmente por los analistas o administradores de negocio, y, a menudo, desde un punto de vista económico. Este punto de vista de los analistas de negocio difiere, por lo general, de la perspectiva utilizada por los diseñadores o desarrolladores de software. Por esta razón, el paso desde los modelos utilizados por los analistas de negocio a los modelos utilizados por los diseñadores o

desarrolladores del SIW, requiere de técnicas o herramientas que faciliten la “alineación” entre ambos puntos de vista.

La propuesta **MDA** (*Model Driven Architecture*) (Miller y Mukerji, 2001), constituye una importante herramienta para la alineación entre los procesos de negocio de alto nivel y las tecnologías de la información (Harmon, 2004). Esto se debe a que MDA proporciona una estructura conceptual que se extiende desde los modelos utilizados por los analistas de negocio, hasta diversos modelos utilizados por los desarrolladores de software. Además MDA ofrece, a través de la transformación de modelos, la posibilidad de que los elementos especificados en un diagrama puedan transformarse, de manera automática, en elementos de otros diagramas más detallados que son derivados a partir de él.

El trabajo de investigación que se presenta en esta Tesis Doctoral define una *aproximación basada en MDA* para el desarrollo orientado a servicios de SIW, de modo que permite aprovechar los beneficios que dicha arquitectura proporciona. Así, en este trabajo, se propone un método orientado a servicios para el desarrollo de SIW que se extiende desde los modelos de negocio de alto nivel hasta los modelos específicos de plataformas, definiendo en concreto modelos específicos para plataformas basadas en servicios Web y proponiendo, además, transformaciones entre dichos modelos.

Esta propuesta se integra en **MIDAS**, un marco metodológico para el desarrollo de SIW basado en MDA (Marcos *et al.*, 2004; Cáceres *et al.*, 2003; Cáceres *et al.*, 2004). MIDAS propone una arquitectura de modelos basada en la arquitectura MDA del OMG (*Object Management Group*).

MIDAS contempla tres aspectos básicos para el modelado y desarrollo de SIW: hipertexto, contenido y comportamiento. Además, por estar basado en MDA, MIDAS propone: modelos independientes de computación (CIM, *Computational Independent Model*), modelos independientes de plataforma (PIM, *Platform Independent Model*) y modelos específicos de la plataforma (PSM, *Platform Specific Model*). Por último, MIDAS define guías para la construcción de los modelos que propone, como también para la transformación entre los mismos. La Figura 1.1 muestra la arquitectura de modelos de MIDAS.

Dentro de la arquitectura de modelos de MIDAS, la propuesta que se realiza en esta Tesis Doctoral se centra en el desarrollo del **aspecto del comportamiento** de los SIW, puesto que se ocupa del modelado de las funcionalidades del mismo; y, como se ha dicho previamente, propone un método orientado a servicios para el desarrollo de SIW que comprende:

- **Modelos independientes de computación** que permitan la comprensión y el modelado del negocio en el que se desenvolverá el SIW.
- **Modelos independientes de plataforma** tanto para el modelado de los servicios, como para el modelado de la composición de los mismos.
- **Modelos específicos de plataforma** que faciliten la generación del SIW utilizando las tecnologías disponibles actualmente para el paradigma SOC.
- **Transformaciones entre modelos**, ya sean éstos del mismo nivel (PIM-PIM, PSM-PSM), como también entre niveles diferentes (CIM-PIM, PIM-PSM).

En la Figura 1.1 aparece coloreado el aspecto y los niveles de abstracción que son objeto de estudio en este trabajo de investigación.

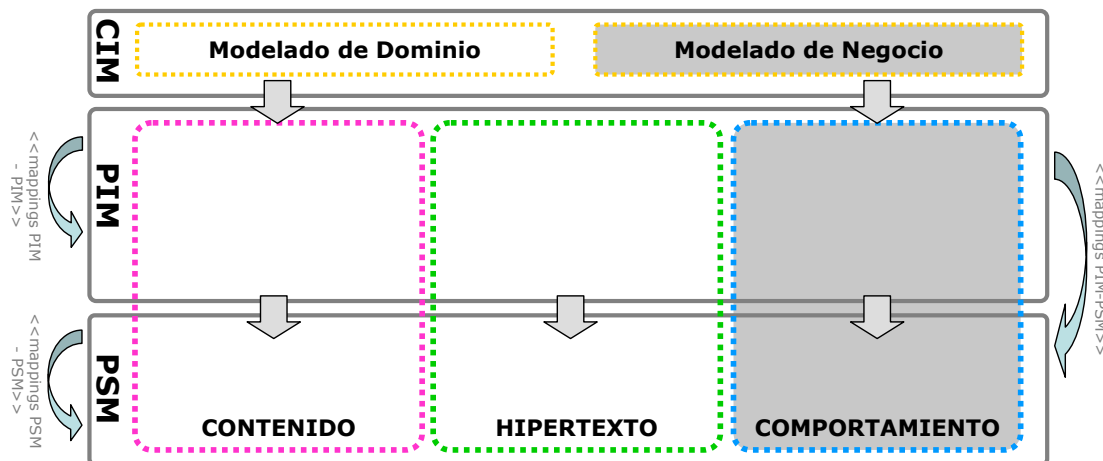


Figura 1.1. Arquitectura de MIDAS

1.2 Hipótesis y Objetivos

Al comienzo del trabajo de investigación que se ha llevado a cabo en esta Tesis Doctoral, y teniendo en cuenta lo descrito en la sección anterior, se planteó la siguiente **hipótesis** de trabajo:

"Es factible la especificación de un método dirigido por modelos que facilite el desarrollo orientado a servicios de SIW, permitiendo alinear los procesos de negocios de alto nivel con plataformas específicas, como las basadas en servicios Web".

El **objetivo principal** de este trabajo de investigación, derivado directamente de la hipótesis, es por tanto:

"La especificación de un método basado en MDA, que facilite el desarrollo orientado a servicios de SIW, partiendo del modelado del negocio de alto nivel y permitiendo obtener los modelos específicos para plataformas basadas en servicios Web".

Para la consecución de este objetivo se han planteado los siguientes objetivos parciales:

1. Análisis de las distintas metodologías existentes para el desarrollo de SIW, prestando especial atención a sus propuestas para el modelado del comportamiento del sistema, determinando sus aportaciones así como sus limitaciones, con el fin de abordar problemas no resueltos.
2. Análisis de las principales propuestas relacionadas con el desarrollo orientado a servicios, tales como propuestas para el desarrollo de servicios Web, para la composición de los mismos o para el desarrollo de procesos de negocios basados en servicios. Dichas propuestas se analizan con el fin de determinar sus aportaciones, he incorporarlas si fuera posible en el método que se va a definir.
3. Definición de un método para el modelado del comportamiento de SIW utilizando una aproximación orientado a servicios que incluya:
 - A. Modelos, tanto independientes de computación como independientes y específicos de plataforma, ya sea mediante la definición de los mismos o bien mediante la incorporación en el método de modelos ya existentes.
 - B. Transformaciones entre modelos, que permitan obtener modelos específicos para la tecnología de servicios Web a partir del modelado de negocio de alto nivel.
4. Integración del método propuesto para el desarrollo orientado a servicios del comportamiento de SIW en el marco de la arquitectura de MIDAS, teniendo en cuenta los demás aspectos contemplados en dicha arquitectura para el desarrollo de SIW.
5. Validación del método propuesto mediante su aplicación a diferentes casos de estudio.

1.3 Marco de Trabajo

El trabajo de investigación que se presenta en esta Tesis Doctoral se inició a principios del año 2003, y se realizó dentro del grupo de investigación Kybele de la Universidad Rey Juan Carlos (URJC). Este trabajo se integra en el marco de diferentes proyectos de investigación relacionados entre sí, como puede verse en la Figura 1.2. Además, parte de este trabajo de investigación se llevó a cabo en dos estancias realizadas en centros extranjeros, entre las que cabe destacar la estancia en la Universidad de Twente (Holanda) por un período de 12 semanas, y que ha permitido optar a la mención de Doctorado Europeo de esta Tesis Doctoral. Dichas estancias se muestran también señaladas en la Figura 1.2.

En la Figura 1.2 se muestra también un período previo a la realización de la Tesis Doctoral que se inicia en el año 2002. En enero de ese año, mientras la doctoranda finalizaba su carrera de Ingeniería en Informática en la Universidad Tecnológica Nacional (Argentina), se realizó una estancia de dos meses en la URJC (**España**), en el marco del *Programa de Cooperación Interuniversitaria - AL.E./2002* de la *Agencia Española de Cooperación Internacional (AECI)*. Durante ese tiempo la doctoranda trabajó bajo la tutoría de la directora de esta Tesis Doctoral, lo que facilitó la posterior integración de la doctoranda en el grupo Kybele y el inicio de este trabajo de investigación en el marco del proyecto DAWIS.

Además, durante ese período la doctoranda colaboró en el proyecto **HERA**, que se centraba en la generación de una herramienta para el desarrollo rápido de portales WAP (Wireless Application Protocol). Dicho proyecto se llevó a cabo junto con la empresa Intesys S.A. El trabajo realizado por la doctoranda en el marco de este proyecto consistió en un estudio de diversas herramientas disponibles para el desarrollo Web.

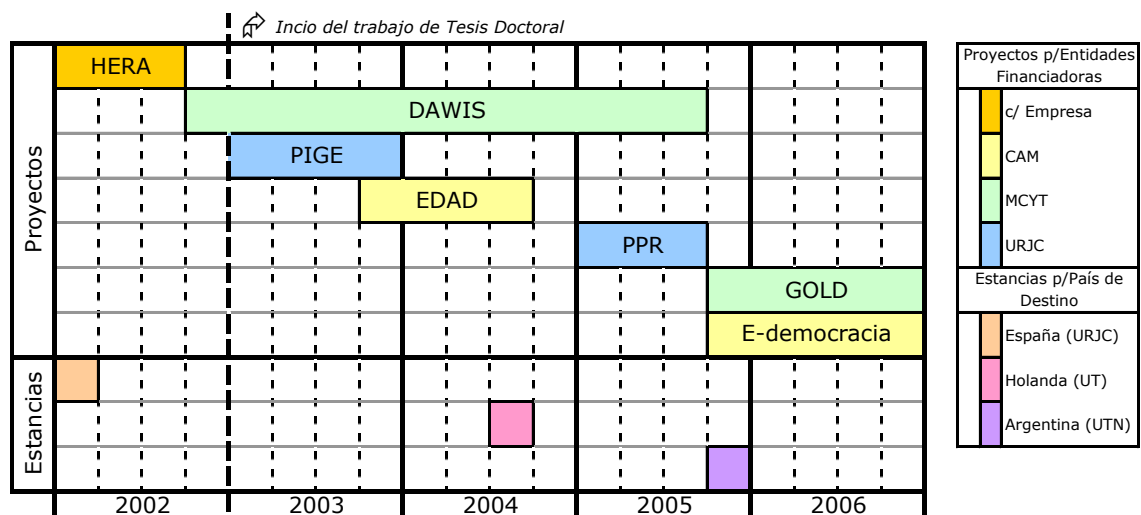


Figura 1.2. Marco de Trabajo de la Tesis Doctoral

En los siguientes apartados se describen los principales proyectos de investigación y estancias en centros extranjeros en los que se enmarca este trabajo de investigación.

1.3.1 Proyectos de Investigación

El trabajo que se presenta en esta Tesis Doctoral se inició en el marco del proyecto **DAWIS**. Este proyecto, comenzó en diciembre del año 2002 y finalizó en el año 2005, fue financiado por el Ministerio de Ciencia y Tecnología (MCYT) [TIC 2002-04050-C02-01]. En el han participado, además del grupo Kybele de la URJC donde residía la coordinación del proyecto, el grupo de Bases de Datos de la Universidad Politécnica de Madrid, ICM (Informática de la Comunidad de Madrid) y CRC Technologies. El objetivo del proyecto DAWIS era la definición de un método genérico basado en modelos que permitiera el desarrollo sistemático y semi-automático de portales Web. Dentro de este proyecto, el trabajo de investigación realizado por la doctoranda se centró en el aspecto del comportamiento del método. Este proyecto fue también financiado por la URJC [**PIGE** 2002/05].

El proyecto **EDAD**, que comenzó en el 2003 y finalizó en el 2004, tenía como objetivo ofrecer el soporte de automatización a las propuestas metodológicas de DAWIS. Este proyecto fue cofinanciado por la Comunidad de Madrid (CAM) y por la Unión Europea [07T/0056/2003 1] y en el participó la Universidad Carlos III de Madrid junto a los demás grupos y entidades integrantes de DAWIS. El objetivo de este proyecto era la definición de una propuesta para la integración de portales basadas en servicios Web; aquí el trabajo de doctoranda consistió en la definición de un modelo específico para la representación gráfica de servicios Web utilizando notación UML.

En el marco del segundo proyecto financiado por la URJC [**PPR**-2004-15], que se extendió de enero a septiembre de 2005, se trabajó en diversos casos de estudios en los que se han realizado pruebas de integración entre los distintos aspectos de modelado (comportamiento, hipertexto y contenido) propuestos por la arquitectura de modelos de MIDAS.

Finalmente, los proyectos GOLD y E-democracia, concedidos a finales del año 2005, aplicarán el método definido en esta Tesis Doctoral para la construcción de SIW adaptados a dos entornos específicos. El proyecto **GOLD**, que comenzó a finales del año 2005 y finalizará en 2008, está siendo financiado por el MCYT [TIN 2005-00010]. El objetivo de este proyecto es el desarrollo de una plataforma para

el desarrollo de SIW, aplicado al caso de un sistema de información para la gestión de imágenes médicas. El proyecto **E-democracia**, liderado por el grupo de estadística de la URJC, y en el que participa el grupo de investigación Kybele, comenzó también a finales de 2005 y se extenderá por un período de 4 años. Este proyecto está siendo financiado por la CAM [TIC 000230-0505] a través de su convocatoria de redes de excelencia, y su objetivo es la definición de conceptos y sistemas de apoyo a la democracia electrónica (e-democracia). La participación de la doctoranda en este proyecto ha comenzado con la aplicación del método propuesto en esta Tesis Doctoral para el desarrollo de aplicaciones Web en entornos de e-democracia.

En el marco de los proyectos de investigación previamente mencionados, se han realizado ya dos tesis doctorales, una por parte la Dra. D^a. Belén Vela Sánchez, centrada en el desarrollo del aspecto de contenido del SIW, y otra por parte de la Dra. D^a. Paloma Cáceres García de Marina en la que se ha definido el proceso de desarrollo de MIDAS y además, se ha propuesto el método para el desarrollo del aspecto de hipertexto del SIW. Este último método, en el que se ha trabajado de manera conjunta con la doctoranda, se ha integrado con el método para el desarrollo del aspecto del comportamiento que es objeto de esta Tesis Doctoral. Además, actualmente están en fase de realización otras tesis doctorales, relacionadas con la presente, a través de las cuales se pretende completar la arquitectura de MIDAS con otros aspectos tales como la arquitectura (Marcos *et al.*, 2006) y la semántica (Acuña y Marcos, 2006), así como también automatizar las transformaciones entre los modelos propuestos (Cáceres *et al.*, 2006a).

1.3.2 Estancias en Centros Extranjeros

Durante el desarrollo del trabajo de investigación de esta Tesis Doctoral, se han realizado dos estancias en centros extranjeros.

Durante los meses de agosto a noviembre de 2004, se realizó una estancia de investigación en el *Information Systems Research Group* de la *Universidad de Twente* (**Holanda**), liderado por el Profesor Roel J. Wieringa. En ese período se ha investigado sobre técnicas y metodologías de ingeniería de requisitos aplicables al desarrollo orientado a servicios de SIW. Tras evaluar algunas técnicas, el estudio se ha centrado en la metodología e³value, un modelo que facilita la comprensión de una idea de negocio mediante la construcción de un modelo de valores, llamado modelo de valor (*value model*) y que forma parte del método propuesto en esta Tesis Doctoral.

Posteriormente, durante los meses de octubre de 2004 a enero de 2005, se realizó otra estancia en *Departamento de Ingeniería en Sistemas de Información* de la Facultad Regional Resistencia en la *Universidad Tecnológica Nacional (Argentina)*. En ese período se ha trabajado con diferentes grupos de profesores y alumnos del área de Ingeniería de software de la universidad de destino, con el fin de realizar pruebas de aplicación del método que se presenta en esta Tesis Doctoral. En concreto, durante ese período se ha comenzado a trabajar en el marco de un proyecto para la automatización de historias clínicas en hospitales públicos.

1.4 Método de Investigación

El método de investigación utilizado en este trabajo de Tesis Doctoral es el resultante de la combinación de un método propuesto por (Marcos y Marcos, 1998), para la investigación en el campo de la Ingeniería del Software, y del método denominado Investigación en Acción (*Action Research*) (Avison *et al.*, 1999).

En (Marcos y Marcos, 1998) se propone un método de investigación (Figura 1.3), basado en el método hipotético-deductivo (Bunge, 1976), que consta de una serie de etapas que, por su generalidad, son aplicables a cualquier tipo de investigación. Dichas etapas comprenden: la *determinación del problema*, la *creación de la hipótesis*, la *definición del método de trabajo*, la *resolución y validación*, el *análisis de los resultados y elaboración de conclusiones*, y la *redacción de un informe final*.

Como puede apreciarse en la Figura 1.3, en el método de investigación se incluye una etapa de definición del propio método. Esta etapa es necesaria, puesto que cada investigación posee sus propias características, lo que implica que no exista un método genérico y universal que pueda aplicarse de forma directa y sin modificaciones a cualquier trabajo de investigación. Según sea la naturaleza del problema a resolver, se utilizará un método deductivo, experimental, etc.

Otra de las etapas del método de investigación es la de *resolución y validación*. En este caso, investigación en el campo de la Ingeniería del Software y cuyo objetivo es la elaboración de una aproximación basada en MDA para el desarrollo orientado a servicios de SIW, para la etapa de resolución y validación utilizamos una adaptación del método de investigación en acción, trabajando sobre diferentes casos de estudio.

El método de investigación en acción utilizado en la resolución y validación de esta Tesis Doctoral se describe en el apartado siguiente.

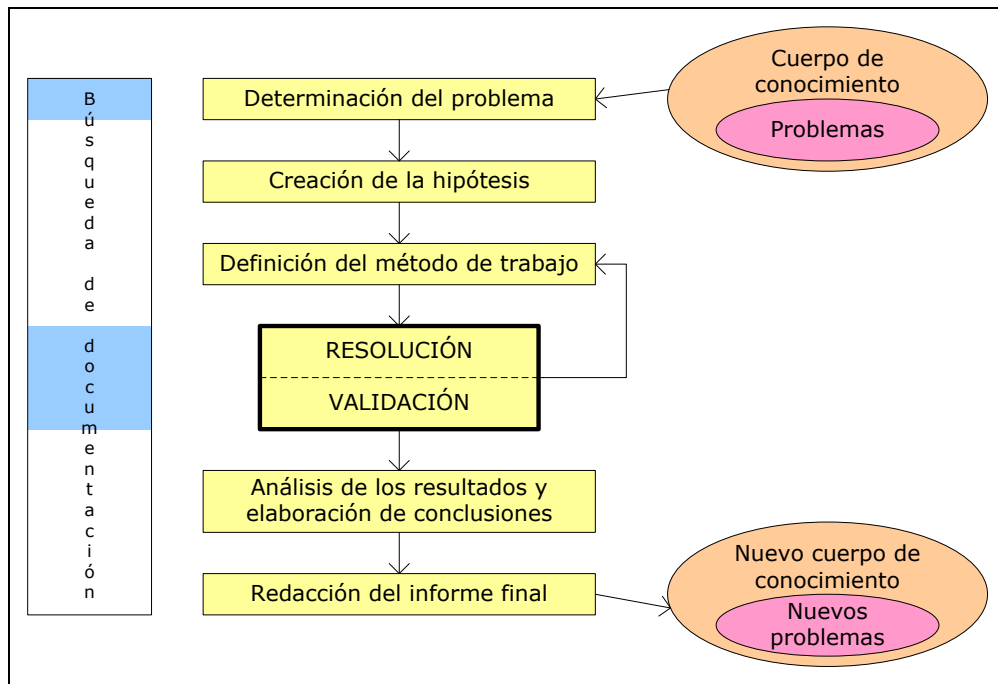


Figura 1.3. Método de Investigación

1.4.1 Método de Resolución y Validación

La *Investigación en Acción* (Avison *et al.*, 1999) es un método de investigación cualitativo utilizado para la validación de los trabajos de investigación mediante su aplicación en proyectos reales, reforzando la interacción entre los investigadores y los participantes de dichos proyectos reales. Este método, por su carácter de validación práctica, es especialmente apropiado para la investigación en Ingeniería y, específicamente, en Ingeniería del Software.

El método de investigación en acción es "la forma que tienen los grupos de personas para preparar las condiciones necesarias para aprender de sus propias experiencias y hacer estas experiencias accesibles a otros" (McTaggar, 1991). Más concretamente, puede definirse como "el proceso de recopilar de forma sistemática datos de la investigación acerca de un sistema actual en relación con algún objetivo, meta o necesidad de ese sistema; de alimentar de nuevo con esos datos al sistema; de emprender acciones por medio de variables alternativas seleccionadas dentro del sistema, basándose tanto en los datos como en las hipótesis; y de evaluar los resultados de las acciones, recopilando datos adicionales" (French y Bell, 1996).

Una de las características más relevantes de este método es que la investigación se realiza a la vez que se aplican sus resultados, de modo que esta aplicación permite validar y refinar los resultados de la investigación en un proceso

iterativo. El proceso de investigación definido en el método de investigación en acción no es por tanto un proceso lineal, sino que va avanzando mediante la realización de ciclos, llamados ciclos de investigación en acción, en cada uno de los cuales se ponen en marcha nuevas ideas, que son puestas en práctica y comprobadas hasta el siguiente ciclo (Wadsworth, 1998).

La Figura 1.4 representa las etapas definidas en (Susman y Evered, 1978) para cada ciclo de investigación en acción. Como puede verse en la figura, en este caso, los ciclos de investigación en acción se realizan durante la etapa de resolución y validación de la Figura 1.3.

A continuación, se describen las etapas de cada ciclo de investigación en acción:

- **Diagnóstico:** es la etapa inicial de cada iteración, involucra la identificación de los aspectos a mejorar y los problemas a resolver propios de cada iteración. Los problemas que se identifican y describen en esta etapa guardan relación con los problemas identificados durante la etapa de determinación del problema, ilustrada en la Figura 1.3. Sin embargo, es importante tener en cuenta que, según van sucediéndose las iteraciones sobre los diferentes casos de estudio, pueden surgir nuevos problemas y oportunidades de mejora de los resultados de las iteraciones anteriores.
- **Planificación de la Acción:** en esta etapa se consideran los diferentes cursos de acción a tomar para resolver los problemas detectados en la etapa anterior, a partir de los casos de estudios identificados preliminarmente durante la etapa de definición del método de trabajo (ver Figura 1.3). El resultado de esta actividad es la identificación de una serie de acciones a ejecutar sobre un determinado caso de estudio.
- **Ejecución de la Acción:** implica la implementación del curso de acción elegido en la etapa anterior sobre el caso de estudio seleccionado.
- **Evaluación:** después de completar la acción, se examinan los resultados. La evaluación incluye determinar si se han alcanzado los efectos esperados de la acción, y si esos efectos mitigaron los problemas identificados previamente. Si los resultados obtenidos fueran no satisfactorios, los problemas identificados se trasladan a las siguientes iteraciones del ciclo de investigación en acción.
- **Especificación del aprendizaje:** si bien esta actividad figura al final del ciclo, se trata de una actividad que se lleva a cabo durante todo el ciclo.

En ella, los conocimientos adquiridos durante el ciclo de investigación en acción, por más que hayan sido no satisfactorios, se comparten con las personas involucradas en el caso de estudio, para que dicho conocimiento pueda ser asimilado en las tareas que realizan y para que se puedan planificar nuevos ciclos de investigación en acción.

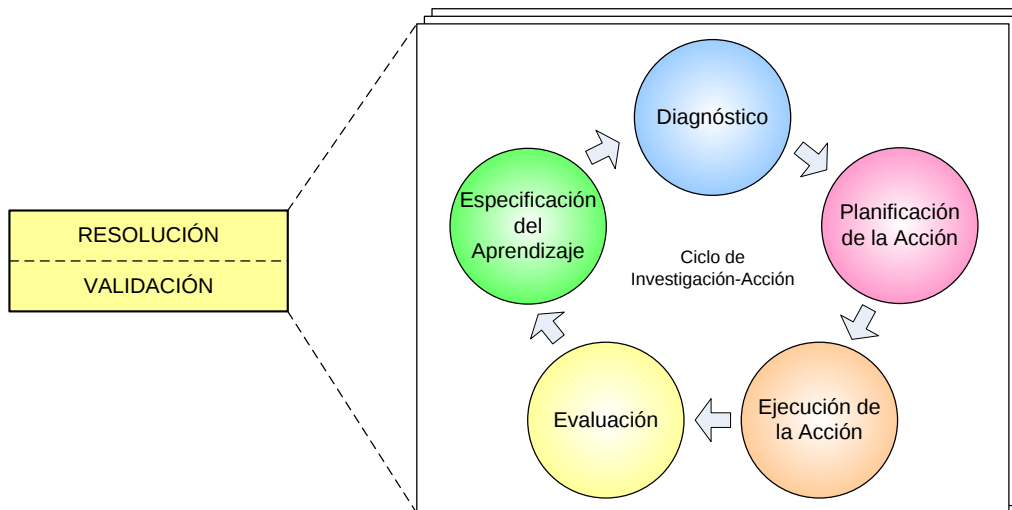


Figura 1.4. Etapa de Resolución y Validación

En el caso concreto de esta Tesis Doctoral, cada ciclo de investigación en acción se aplicó iterativamente sobre diferentes *casos de estudio*. Desde el punto de vista de las ciencias sociales, los casos de estudio se utilizan para investigar una situación o fenómeno que ocurre en un contexto real (Yin, 2003). Sin embargo, cuando se habla de investigación en sistemas software, como es el caso de esta Tesis Doctoral, los casos de estudios pueden abordar casos como el desarrollo o la implantación de un sistema de información en una organización particular (Cornford y Smithson, 1996).

Los casos de estudio realizados para la validación de este trabajo de investigación (ver Tabla 1.1) han consistido en el desarrollo de *un servicio Web*, sobre el que se ha aplicado una técnica concreta, y *cinco SIW*, sobre los que se han ido aplicando las diversas versiones del método propuesto de forma completa.

Los casos de estudio que aquí se presentan se clasifican en dos tipos: *casos de estudio de laboratorio* y *casos de estudio reales*. Dicha clasificación se basa en dos de los principales tipos de estudios empíricos cuantitativos utilizados en la validación en el campo de la Ingeniería de Software, los experimentos de laboratorios y los casos de estudios propiamente dichos (Juristo y Moreno, 2001). Los experimentos de laboratorio se desarrollan generalmente en condiciones

controladas, es decir manipulando y controlando las variables que sean necesarias y observando los resultados que se obtienen (Cornford y Smithson, 1996). Sin embargo, tales experimentos sugieren, a veces, que el contexto en el que desarrolla no es equivalente a lo que se pueden encontrar en una situación real (Cornford y Smithson, 1996); por ello, es importante también la validación en casos de estudio reales.

A continuación presentamos la clasificación de los casos de estudio desarrollados, los cuales se recogen en la Tabla 1.1:

- **Casos de estudio de Laboratorio:** llamamos así a los casos de estudio que han sido diseñados específicamente para investigar necesidades en el desarrollo orientado a servicios del comportamiento de SIW. La definición de este tipo de casos de estudio como “de laboratorio”, se fundamenta en que se han llevado a cabo en condiciones similares a las que se desarrolla un “experimento de laboratorio”; es decir, bajo condiciones controladas. Tales casos se desarrollaron con los siguientes objetivos: *aplicar prácticas de algunas de las metodologías de desarrollo presentes hasta el momento con el fin de detectar necesidades específicas para el desarrollo basado en servicios; y, aplicar las primeras versiones de las técnicas de modelado propuestas en esta Tesis Doctoral, a efectos de detectar posibles inconsistencias, fallos, o carencias en las mismas.*
- **Casos de estudio Reales:** estos casos de estudio han consistido en el desarrollo de SIW reales y se realizaron con los siguientes objetivos: *aplicar las distintas versiones del método para validar tanto las técnicas de modelado propuestas, como el método en sí mismo; y, detectar posibles mejoras o nuevas necesidades que surgen para dominios particulares.* Debido a que la generalización de los resultados obtenidos de un caso de estudio solo puede realizarse a partir de la ejecución de múltiples casos en los que se evidencien los mismos resultados (Cornford y Smithson, 1996), en este trabajo, se han realizado varios casos de estudios reales, todos ellos pertenecientes a diversos dominios de aplicación, en concreto: un sistema de información de tratamiento de imágenes médicas, un sistema de democracia electrónica y otros sistemas de información clásicos de gestión.

Tabla 1.1. Casos de estudios utilizados en la validación de la Tesis Doctoral

CASOS DE ESTUDIO	
de Laboratorio	Caso 0: Servicio Web de Información de Vuelos
	Caso 1: WebConference: Sistema para la Gestión del Envío de Trabajos a Conferencias
Reales	Caso 2: GesIMED: Sistema para la Gestión y Procesamiento de Imágenes Médicas
	Caso 3: PARBUD: Sistema para la Gestión de Presupuestos Participativos
	Caso 4: Atento: Sistema de Ventanilla Única para la Administración Pública
	Caso 5: ISLA: Sistema para la Gestión de una Autoescuela

Los casos de estudio realizados, incluyendo tanto los resultados obtenidos como las nuevas necesidades que fueron surgiendo a partir de cada uno de ellos, se describen detalladamente en el capítulo 4 de esta memoria. A continuación, y a modo de ejemplo, en la Figura 1.5 se presenta la aplicación del método de Investigación en Acción a uno de los casos de estudio reales desarrollados para la validación de esta Tesis Doctoral, el caso de GesIMED. En la Figura 1.5 se reflejan también los diferentes actores que, según (Wadsworth, 1998), intervienen en el proceso:

- El *investigador*, es aquel que impulsa, como sujeto, el proceso investigador. En este caso, la doctoranda.
- El *objeto investigado*, que es el problema que se desea resolver. En este caso, la problemática del desarrollo orientado a servicios de SIW, en particular, el desarrollo del aspecto del comportamiento de los mismos.
- El *grupo crítico de referencia*, es aquel para quien se investiga y que tiene un problema por resolver, por lo que participa en la investigación. En este caso podría definirse como el conjunto de desarrolladores que han participado en los casos de estudio realizados. Así, por ejemplo, en el caso de GesIMED, el cuerpo crítico de referencia son los desarrolladores, que precisan de metodologías orientado a servicios para el desarrollo del SIW que se desea construir.
- *Beneficiarios de la investigación*, son aquellos que pueden beneficiarse de del resultado de la investigación, aunque no haya participado directamente en el proceso. En este caso, cualquiera que pueda verse beneficiado por la aplicación del método propuesto. Los principales beneficiados de esta

investigación serían los desarrolladores del SIW, así como las organizaciones (clientes) y usuarios de los SIW desarrollados aplicando SOD-M. Así, para el caso del desarrollo del SIW GesIMED, los beneficiarios son los miembros del grupo GTEBIM (Grupo de Tecnología Electrónica, Bioingeniería e Imagen Médica), los investigadores en imágenes médicas, los propios desarrolladores del SIW GesIMED y otros desarrolladores de SIW.

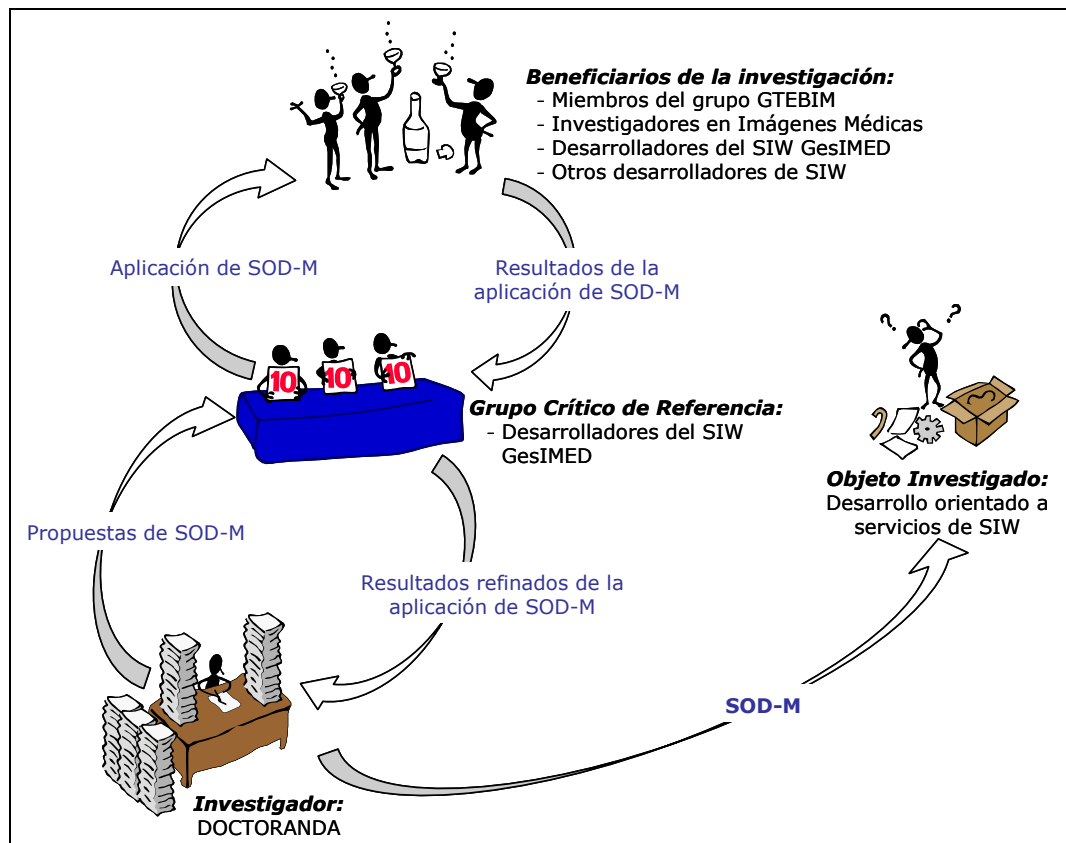


Figura 1.5. Aplicación del método de Investigación en Acción

1.5 Organización de la Memoria

La organización de los restantes capítulos de este trabajo de Tesis Doctoral es la siguiente:

- **Capítulo segundo:** en este capítulo se analiza el estado actual de las metodologías de desarrollo de SIW, centrándose en sus propuestas para el modelado del comportamiento del sistema, así como las principales propuestas relacionados con el desarrollo orientado a servicios. También en este capítulo se resume la propuesta del OMG, MDA.

- **Capítulo tercero:** presenta la propuesta de este trabajo de Tesis Doctoral, es decir, una aproximación metodológica basada en MDA para el desarrollo orientado a servicios del comportamiento del SIW, denominado SOD-M (*Service-Oriented Development Method*), así como su integración con la arquitectura de MIDAS que provee un marco para el desarrollo completo de SIW.
- **Capítulo cuarto:** en este capítulo se presenta la validación del método propuesto en esta Tesis Doctoral, mediante su aplicación a diferentes casos de estudio de laboratorio y reales.
- **Capítulo quinto:** presenta el análisis de los resultados, las principales aportaciones del trabajo, la contrastación de resultados en distintos foros tanto nacionales como internacionales, así como líneas de investigación que quedan abiertas para futuros trabajos.
- **Bibliografía:** en este capítulo se incluye toda la bibliografía consultada para la realización de este trabajo de investigación.
- **Lugares de Internet:** incluye direcciones Web de lugares consultados en Internet que han sido representativos para la realización de este trabajo.
- **Apéndices:** finalmente, se incluyen dos apéndices: el primero de ellos presenta la relación de siglas utilizadas a lo largo de este documento; y el segundo presenta, de forma detallada, el conjunto de casos de estudio utilizados en la validación de este trabajo de Tesis Doctoral.

ESTADO DEL ARTE

En este capítulo se presenta el estado de la investigación en cuestiones relacionadas con la propuesta que se realiza en esta Tesis Doctoral: una aproximación basada en MDA para el desarrollo orientado a servicios de SIW. Como se ha expuesto en el capítulo anterior, actualmente no existen metodologías que utilicen un enfoque orientado a servicios para el desarrollo de sistemas de información. Sin embargo, es posible identificar dos tipos de propuestas relacionadas con este trabajo: por un lado, aquellas que provienen del campo de la ingeniería Web y que, como SOD-M, definen metodologías para el desarrollo de SIW; y por otro, aquellas propuestas que definen métodos o técnicas de modelado para el desarrollo de ciertos aspectos relacionados con los sistemas orientados a servicios. Por ello, y dado que en este trabajo se propone definir una aproximación basada en MDA para el desarrollo orientado a servicios de SIW, en este capítulo se presenta: una visión de MDA; una visión de las metodologías de desarrollo de SIW; y una visión de propuestas relacionadas con el desarrollo orientado a servicios. Así, en primer lugar, en la sección 2.1 se presenta una introducción a MDA, describiendo sus principales características. A continuación, en la sección 2.2 se presentan las propuestas metodológicas más relevantes del campo de la Ingeniería Web, analizando en qué manera soportan el desarrollo orientado a servicios de SIW, y poniendo especial énfasis en sus propuestas para el modelado del comportamiento de tales sistemas. Al final de la sección, se realiza una comparación entre dichas propuestas determinando sus aportaciones y limitaciones, y destacando las principales contribuciones del método propuesto en esta Tesis Doctoral. Por último, en la sección 2.3 se analizan diversas propuestas relacionadas con el desarrollo orientado a servicios. Estas propuestas se estudian con el propósito de determinar sus principales aportaciones y limitaciones, y analizar si son susceptibles de ser incorporadas en el método que se pretende definir. Finalmente en la sección 2.4 se describen las principales conclusiones del capítulo.

2.1 Model Driven Architecture – MDA

El Object Management Group (OMG) es un consorcio internacional dedicado al cuidado y el establecimiento de diversos estándares para un amplio rango de tecnologías. Uno de los principales objetivos del OMG desde sus inicios en el 1989, ha sido ayudar a los usuarios de ordenadores a solventar los problemas de integración e interoperatividad entre sus sistemas, proporcionando para ello

especificaciones abiertas e independientes de fabricantes. Así, persiguiendo este objetivo, a fines del año 2000, la OMG introduce MDA, una arquitectura para el desarrollo de software dirigido por modelos (Miller y Mukerji, 2001). La característica fundamental de MDA es la *definición de modelos formales como elementos de primera clase* para el diseño e implementación del sistema y la *definición de transformaciones de un modelo a otro*, de forma que estas transformaciones puedan ser automatizadas. MDA propone la especificación del sistema separando la especificación de la lógica del mismo de los detalles de su implementación en una plataforma concreta. Así, MDA y los estándares que soporta, permite que la lógica de los sistemas, expresada en modelos, sea realizada en diversas plataformas específicas a través de la definición de transformaciones (*mappings*). De este modo, MDA proporciona a las organizaciones una manera de salvar los problemas de integración de sus sistemas actuales con las nuevas plataformas que surgen para su implementación, a la vez que le permite preservar la lógica de sus aplicaciones y por tanto sus inversiones en plataformas existentes.

2.1.1 Los Modelos en MDA

En función del nivel de abstracción, MDA define tres niveles conceptuales de modelado:

En un nivel, se modelan los requisitos del sistema mediante Modelos Independientes de Computación (CIM, *Computer Independent Model*) que sirven de puente entre los expertos del negocio y los desarrolladores que afrontan la realización del sistema. Los modelos de este nivel no representan detalles del sistema en sí, sino más bien del dominio de aplicación del mismo.

Otro nivel comprende los Modelos Independientes de la Plataforma (PIM, *Platform Independent Model*), que se utilizan para la representación la funcionalidad y estructura del sistema abstrayéndose de los detalles tecnológicos de la plataforma en la que se implementará. Una buena forma de definir los PIM es pensar en el desarrollo del sistema para una máquina virtual de tecnología neutral, que proporciona servicios que se implementarán luego de forma diferente en cada plataforma concreta. Además, los modelos de nivel PIM pueden ser refinados tantas veces como se quiera hasta obtener una descripción del sistema con el nivel de claridad y abstracción deseado.

Finalmente, en otro nivel, se encuentran los Modelos Específicos de la Plataforma (PSM, *Platform Specific Model*), que combinan las especificaciones contenidas en un PIM, con los detalles de la plataforma elegida. A partir de los

distintos PSM se pueden generar automáticamente distintas implementaciones del mismo sistema.

Adicionalmente, podemos considerar el mismo código que implementa el sistema como otro modelo, de más bajo nivel de abstracción, que utiliza una notación distinta de la del resto de modelos.

Una de las mayores aportaciones de MDA reside en la naturaleza de todos estos modelos: son modelos formales, y por lo tanto, entendibles por el ordenador. Frente a la concepción tradicional de estos modelos, contemplados sólo como parte de la documentación del sistema, en el enfoque de MDA los modelos son objetos de primera categoría, y pasan a ser auténticos productos software, y de hecho, el sistema finalmente implementado se corresponde directamente con dichos modelos.

2.1.2 Transformaciones de modelos en MDA

La transformación de modelos en MDA se define como el proceso de convertir un modelo (o modelos) del sistema en otro modelo (o modelos) del mismo sistema. MDA propone la definición de reglas de transformación (*mapping rules*) solo entre modelos de nivel PIM y PSM; respecto de las transformaciones desde el nivel CIM, MDA propone que se mantenga una relación de "trazabilidad"¹ entre los requisitos del sistema expresados en los modelos CIM y los elementos o artefactos representados en los modelos PIM y PSM que permiten llevar a cabo tales requisitos.

Los tipos de transformaciones contemplados en MDA pueden agruparse en dos grandes conjuntos (France y Bieman, 2001):

- *Transformaciones verticales*: son aquellas en las que los niveles de abstracción del modelo origen y destino son diferentes. Refinar un modelo o implementarlo en un lenguaje de programación concreto son ejemplos de transformaciones verticales. En el ámbito de MDA se contemplan varias transformaciones verticales. Así por ejemplo, cuando un PIM está suficientemente refinado, se transforma en un modelo dependiente de la infraestructura final de ejecución, de modo que un PIM se transforma en uno o varios PSM; igualmente, cuando partiendo de una implementación en una plataforma concreta, se desea abstraerse de los detalles concretos

¹ Este término es también expresado como *dependencia de traza*, y se refiere a la dependencia que indica una relación de proceso o histórica entre dos elementos que representan el mismo concepto, sin reglas para derivar el uno del otro (OMG, 2003a).

de dicha plataforma, se está aplicando una transformación vertical para pasar del PSM al PIM.

- *Transformaciones horizontales:* son aquellas en las que el modelo origen y destino corresponden al mismo nivel de abstracción. La principal aplicación de este tipo de transformaciones es la evolución de modelos; dicha evolución puede ser: perfectiva, por ejemplo, para mejorar un diseño; correctiva, para corregir errores en el diseño; o bien, adaptativa, para introducir en un diseño nuevos requisitos o restricciones. La aplicación de transformaciones horizontales en MDA permite además obtener modelos de diferentes vistas del sistema (por ejemplo, de contenido, de hipertexto o de comportamiento) pasando de un PIM a otro; o bien obtener también modelos específicos para distintas plataformas pasando de un PSM a otro.

En la Figura 2.1 se muestra la descripción del metamodelo de MDA. En ella se indica la relación de PIM y PSM con el metamodelo de UML y los tipos de transformaciones posibles entre estos.

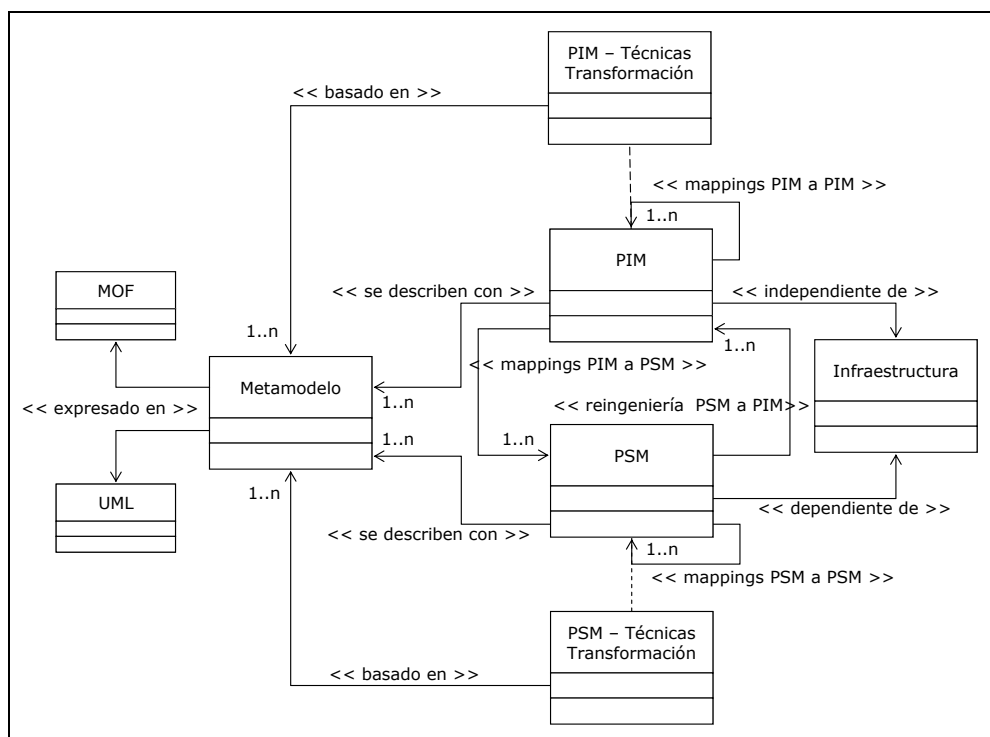


Figura 2.1. Metamodelo de MDA

Durante los últimos años, se ha producido una gran actividad investigadora en el campo de las transformaciones. Estas investigaciones han resultado en numerosas aproximaciones: basadas en teoría de grafos, lenguajes formales, lenguajes específicos de dominio (*DSL, Domain Specific Language*), etc. para definir

transformaciones entre modelos, surgiendo así propuestas concretas tales como ATL (*Atlas Transformation Language*) (Jouault y Kurtev, 2005) o AGG (*Attributed Graph Grammar System*) (Taentzer, 2000) que proporcionan la funcionalidad necesaria para automatizar transformaciones entre modelos expresadas con grafos.

QVT (*Query View Transformation*) (OMG, 2005), es un estándar promovido por el OMG para la especificación de transformaciones entre modelos. Cabe destacar que en actualidad no existe una implementación de referencia que abarque la propuesta completa de QVT, aunque si han surgido propuestas que lo soportan en parte, tales como la herramienta MOMENT (Queralt *et al.*, 2006) y SmartQVT (ModelWare Project, 2006).

En este trabajo de Tesis Doctoral se ha considerado, como una primera aproximación para la definición de las transformaciones, la posibilidad de formalizar las reglas de transformación entre los modelos propuestos por medio de gramáticas de grafos, dado que resultan más intuitivas y proporcionan soporte directo para implementarlas. En concreto, la propuesta que se realiza en esta Tesis Doctoral incluye los siguientes pasos: en primer lugar, describir las reglas mediante lenguaje natural, y, posteriormente, formalizar dichas reglas de transformación por medio de diagramas que utilizan gramáticas de grafos, de que modo se puedan implementar utilizando algunas de las propuestas ya existentes para automatizar las transformaciones entre modelos, tales como AGG System (Taentzer, 2000).

2.2 Metodologías para el Desarrollo de SIW

El rápido crecimiento de la Web a mediados de la década de los 90, ha motivado el interés por el desarrollo de sitios o aplicaciones Web por parte de muchas organizaciones que veían en ellas, una manera de extender sus capacidades de negocios y de proveer mayores facilidades de acceso a sus clientes o usuarios. A pesar de que muchas organizaciones han conseguido desarrollar exitosas aplicaciones de alto rendimiento, otras han fracasado debido a la utilización de pobres o incorrectos diseños y procesos de desarrollo. Atendiendo a este tipo de problemas surge, a finales de los 90, una nueva área de trabajo denominada **Ingeniería Web** (*Web Engineering*). Las aportaciones de esta área surgen con el objetivo de facilitar la comprensión, el desarrollo y la implementación de aplicaciones o sistemas Web, así como para mejorar la mantenibilidad y calidad de las mismas (Ginige y Murugesan, 2001a).

Las primeras propuestas metodológicas que existen en el campo de la Ingeniería Web han surgido, en general, como adaptaciones de metodologías

clásicas que se especializaban en un tipo concreto de desarrollo. Así algunos trabajos proceden del campo de la hipermedia y multimedia, tales como: HDM (Garzotto *et al.*, 1993), RMM/ERMM (Isakowitz *et al.*, 1995; Isakowitz *et al.*, 1998), OOHDM (Schwabe y Rossi, 1995; Schwabe y Rossi, 1998), UWE (Hennicker y Koch, 2000; Koch *et al.*, 2000), W2000 (Baresi *et al.*, 2000) y Ariadne (Díaz *et al.*, 2001); mientras que otros proceden principalmente de aproximaciones para el desarrollo de bases de datos, como la propuesta de Fraternali (Fraternali, 1999), Araneus (Atzeni *et al.*, 1998; Mecca *et al.*, 1999) y WebML (Ceri *et al.*, 2000). También surgieron otros trabajos como el método WSDM (De Troyer y Leune, 1998) para el diseño de sitios Web centrado en el usuario y otras que son extensiones de metodologías de desarrollo de propósito general como (Conallen, 2000; Gómez *et al.*, 2001).

Sin embargo, las nuevas propuestas tecnológicas para la Web tales como XML, servicios Web, automatización de procesos de negocios, Web semántica, etc., han impulsado en los últimos años, una evolución de las técnicas y metodologías para el desarrollo de SIW. Así, han surgido nuevas extensiones: tanto para la representación de procesos de negocios en SIW, tales como las extensiones a WebML (Brambilla *et al.*, 2003a; Brambilla *et al.*, 2006), a UWE y OO-H (Koch *et al.*, 2004) y a OOHDM (Schmid y Rossi, 2004); como también para la representación de SIW que interactúan con servicios Web, como la extensión a WebML (Brambilla *et al.*, 2003b; Manolescu *et al.*, 2005). Además, han surgido otras propuestas novedosas, tales como el modelo WIED (Tongrunrojana y Lowe, 2004) que actúa como puente entre el modelado de negocios de alto nivel y los modelos de diseño de WebML.

El análisis de las metodologías de desarrollo de SIW que se presenta en esta Tesis Doctoral, se centra en el estudio de los modelos que éstas proponen para la especificación de la funcionalidad o comportamiento del sistema, así como en otras características generales de las metodologías, tales como paradigma de desarrollo utilizado, su orientación, etc. A continuación se describen los criterios de evaluación a tener en cuenta, indicando qué se estudiará en cada uno de ellos:

- 1. Modelos propuestos:** cada una de las metodologías de desarrollo de SIW que se analizan en esta sección define un conjunto de modelos necesarios para el diseño de los SIW. En este punto el estudio se centrará en aquellos modelos propuestos por cada una de las metodologías para la especificación del comportamiento de los SIW, así como para el modelado de negocio de alto nivel; en particular, se verá si definen técnicas

específicas para el desarrollo orientado a servicios, tales como técnicas para el modelado de servicios y técnicas para el modelado de la composición de los mismos.

2. Paradigma de desarrollo: el paradigma de desarrollo se entiende aquí como el enfoque u orientación que utiliza una metodología concreta para la construcción de todo el SIW. Como se ha dicho previamente, se puede hablar de dos enfoques utilizados tradicionalmente en el desarrollo de sistemas (Booch *et al.*, 1999): el *enfoque estructurado o algorítmico*, en el que el bloque principal de construcción de todo del software es el procedimiento o función; y el *enfoque orientado a objetos* (OO), en el que el bloque de construcción de todo el software es el objeto o clase. En este trabajo de Tesis Doctoral se presenta un nuevo enfoque para el desarrollo de sistemas que llamamos *enfoque orientado a servicios* (OS), y que definimos como aquel en el que el bloque principal para la construcción del software es el servicio. Este nuevo enfoque surge para permitir a los desarrolladores de software aprovechar al máximo los beneficios del nuevo paradigma de computación SOC (Papazoglou y Georgakopoulos, 2003), que se basa en la utilización de servicios como elementos básicos para el desarrollo de aplicaciones. En este punto se determinará qué enfoque de desarrollo es utilizado por cada una de las metodologías para la construcción de todo el SIW.

3. Enfoque para la construcción del hipertexto: en este punto se pretende analizar cómo, las diversas metodologías, proponen construir el hipertexto de los SIW; de esta manera el objetivo es resaltar las diferencias en los procesos de desarrollo del SIW de cada una de las metodologías que se analicen. En este trabajo se propone un criterio para distinguir un enfoque de otro, aunque evidentemente existen diversas propuestas o aproximaciones para ello. El criterio propuesto consiste en distinguir diferentes enfoques teniendo en cuenta el modelado de qué elemento/s de los sistemas de información (*objetos, procesos, servicios o tareas*) es/son determinante/s para la construcción del hipertexto del SIW. Dicho de otra manera, se pretende distinguir a partir de qué elemento/s se obtiene la estructura básica de la navegación de la aplicación Web (páginas Web y enlaces) en cada propuesta. Así, por ejemplo, se dice que una metodología utiliza un enfoque *orientado a objetos* cuando este se centra, principalmente, en los objetos del dominio para derivar el hipertexto del SIW; es decir, cuando tanto las páginas Web como los

enlaces esenciales entre ellas se obtienen a partir de un modelo en el que se representan los objetos del dominio, aunque eso no implica que algunas páginas Web o enlaces auxiliares se obtengan de otro modelo, como por ejemplo un modelo de proceso. De manera similar, se dice que una metodología utiliza un enfoque *orientado a servicios*, o un enfoque *orientado a procesos*, o un enfoque *orientado a tareas*, cuando el hipertexto del SIW (páginas Web como los enlaces esenciales entre ellas) se obtiene a partir de un modelo en el que se representan los servicios, o los procesos, o las tareas, respectivamente.

- 4. Notación de modelado:** en este punto se analizará si las propuestas utilizan UML como lenguaje de modelado, por ser UML es el lenguaje estándar para el modelado de sistemas de información. Además, se analizará si dichas propuestas definen perfiles UML para el desarrollo específico de SIW. Un Perfil UML se define como un paquete que contiene elementos de modelado que han sido personalizados para un propósito o dominio específico (OMG, 2003c).
- 5. Aproximación basada en MDA:** para cada una de las metodologías de desarrollo de SIW, se estudiará si estas definen propuestas que se adaptan a las características de MDA, analizado además en qué grado; es decir, si definen modelos para los distintos niveles de abstracción propuestos por MDA y si se especifican transformaciones entre ellos.

En el apartado 2.2.10 de esta sección, se muestra una comparación en base a estos criterios, de las metodologías de desarrollo de SIW analizadas, así como una tabla resumen (Tabla 2.1) en la que se destacan las principales contribuciones del método propuesto en esta Tesis Doctoral.

2.2.1 OOHDM: Object Oriented Hypermedia Design Method

Esta aproximación (Schwabe y Rossi, 1995; Schwabe *et al.*, 2001) se centra en el proceso de diseño y desarrollo de aplicaciones hipermedia (Web). OOHDM propone un proceso que consta de cinco etapas, siguiendo un modelo de proceso con prototipado, iterativo e incremental: *captura de requisitos*, *diseño conceptual*, *diseño navegacional*, *diseño de la interfaz abstracta* e *implementación*.

- La etapa de *captura de requisitos* propone la especificación de escenarios de uso, casos de uso y diagramas de interacción de usuarios (UID, *User Interaction Diagram*). El UID utiliza una notación propietaria y representa la interacción entre el usuario y la aplicación.

- El *diseño conceptual* hace referencia a la construcción del modelo conceptual del dominio de la aplicación, utilizando para ello notación UML. OOHDM define en esta etapa el comportamiento de la aplicación, especificando el comportamiento de cada uno de los objetos del dominio.
- La etapa de *diseño navegacional* se realiza a través de dos modelos: el modelo de clases navegacionales y el modelo de contextos navegacionales. El primero está basado en el modelo de datos conceptual y representa los elementos navegables y los atributos a los que se van a tener acceso, teniendo en cuenta los perfiles de los usuarios. El segundo contiene las estructuras navegacionales y se representa mediante nodos y enlaces. El contexto navegacional organiza el espacio navegacional en un conjunto que puede ser navegado en un orden particular. Ambos modelos utilizan notación UML extendida.
- En la etapa de *diseño de la interfaz abstracta* se especifica cómo aparecen en la interfaz los elementos identificados en la etapa anterior y cuál es el comportamiento de la interfaz ante determinados eventos externos o generados por el usuario. Este diseño se realiza mediante la utilización de ADVs (Abstract Data Views).
- La *implementación* incluye la transformación de los objetos de la interfaz en objetos de implementación.

En (Schmid y Rossi, 2004) se propone una extensión a OOHDM para el modelado de procesos de negocios. Esta propuesta define nuevos estereotipos para la representación de procesos de negocios, que se añaden al modelo conceptual y navegacional; a la vez que agrega un nuevo diagrama de actividad en UML, para la representación del flujo de control de los procesos.

Como su nombre lo indica OOHDM utiliza los principios básicos del paradigma orientado a objetos para el desarrollo de aplicaciones. En cuanto al enfoque para la construcción del hipertexto, se puede decir que OOHDM utiliza un enfoque orientado a objetos para el desarrollo del hipertexto del SIW, aunque para la construcción de los distintos contextos navegacionales se tienen en cuenta también los perfiles de los usuarios de la aplicación Web.

OOHDM utiliza UML como notación para la mayoría de sus modelos, sin embargo, hasta el momento, no se ha definido un perfil que reúna todos los

elementos de modelado propuestos por la metodología, como tampoco sus metamodelos.

Respecto a su adaptación al desarrollo dirigido por modelos, en (Schmid y Donnerhak, 2005) se propone una aproximación, denominada OOHDMDA, para la generación de aplicaciones Web basadas en *Servlets* a partir de la propuesta inicial OOHDM. En concreto, la propuesta define transformaciones entre modelos PIM que describen el esquema conceptual y esquema navegacional y modelos PSM que permiten la generación de código Java. Cabe aclarar que la propuesta no describe las transformaciones entre los modelos, sino, transformaciones entre ficheros XMI. Así, cada fichero XMI que describe un PIM es transformado en un fichero XMI que describe un PSM.

2.2.2 UWE: UML Based Web Engineering

Esta propuesta (Koch *et al.*, 2000; Hennicker y Koch, 2000) proporciona soporte metodológico para la construcción de aplicaciones Web basada en UML.

UWE propone un proceso, iterativo e incremental, que consta de cuatro etapas principales: *análisis de requisitos*, *diseño conceptual*, *diseño navegacional* y *diseño de la presentación*.

- En la etapa de *análisis de requisitos* se propone la representación de un diagrama de casos de uso. En el se pueden representar, además de los casos de uso estándar de UML, casos de uso navegacionales. Éstos representan las actividades de navegación del usuario del sistema.
- En la etapa de *diseño conceptual*, se realiza el modelado conceptual del dominio del SIW, que se representa a través de un diagrama de clases.
- Para el *diseño navegacional* se considera tanto el modelado conceptual como el análisis de requisitos. Esta etapa consta de dos actividades de modelado: en primer lugar, se realiza el modelado espacial de navegación, en el que se representan las clases navegacionales, que se obtienen a partir de las clases del modelo conceptual, y cuyas instancias se visitan durante la navegación; en segundo lugar, se realiza el modelado estructural de navegación, que representa la estructura de la navegación agregando elementos de navegación (*menu*, *indexs*, *guide tours* y *queries*) al modelo de espacial de navegación, dichos elementos de navegación se obtienen a partir de las relaciones entre clases del modelo conceptual. Estos dos modelos, espacial de navegación y estructural de navegación, se

corresponden respectivamente con los modelos de fragmentos y de navegación propuestos en RMM (Isakowitz *et al.*, 1995).

- La última etapa es la del *diseño de la presentación*. En esta etapa se construye un modelo de presentación de la aplicación basada en la información recogida en etapas previas.

En (Koch *et al.*, 2004) se propone, como en la anterior propuesta, una extensión a UWE para el modelado de procesos de negocios, agregando nuevos estereotipos y nuevos modelos para este fin. En concreto, se propone la realización de dos actividades: el *modelado conceptual del proceso*, a través de un diagrama de clases y el *modelado del flujo de proceso*, mediante un diagrama de actividad. Además, se añaden nuevos estereotipos para la representación de procesos en el modelo navegacional y en el modelo de presentación.

La metodología UWE es orientada a objetos. En cuanto al enfoque para la construcción del hipertexto, UWE utiliza también un enfoque orientado a objetos para la construcción del hipertexto, dado que UWE propone derivar tanto las páginas Web como los principales enlaces entre ellas a partir del modelo de clases en primer lugar, aunque también utiliza, en segundo lugar, el modelo de flujo de proceso. Dicho modelo se utiliza para obtener nuevas clases navegacionales y enlaces entre ellas, que se agregan al modelo que se obtiene a partir del modelo de clases.

UWE utiliza UML como notación para la representación de sus modelos y además ha definido un perfil para el desarrollo de aplicaciones Web.

Con respecto a su relación con MDA, en (Meliá *et al.*, 2005) se presenta una aproximación para la transformación de modelos de diseño a modelos de implementación denominada WebSA. En ella, se propone un proceso de desarrollo y un conjunto de modelos arquitectónicos para especificar los diferentes puntos de vista que hay que considerar en las aplicaciones Web. Con ello, permite la integración de los diferentes puntos de vista considerados, a través de las transformaciones de modelos, usando la notación QVT-P, (QVT Partners, 2003). Los modelos propuestos por la metodología de UWE descritos anteriormente, se corresponden con el nivel de abstracción PIM de la arquitectura de MDA. Para el nivel PSM, se propone utilizar modelos específicos de las plataformas J2EE, .NET u otros.

2.2.3 WebML: Web Modeling Language

WebML (Ceri *et al.*, 2000) es un lenguaje modelado de alto nivel para la especificación de aplicaciones Web. En esta aproximación, se propone la especificación de la aplicación Web en base a cuatro perspectivas: *modelo estructural*, *modelo del hipertexto*, *modelo de presentación* y *modelo de personalización*. WebML define también un proceso iterativo, con las siguientes etapas: *recolección de requisitos*, *diseño de datos*, *diseño del hipertexto*, *diseño de presentación*, *diseño de usuarios y de grupos* y *diseño de personalización*.

- En la etapa de *recolección de requisitos*, se especifican los requisitos de la aplicación tales como, usuarios a los que está dirigida, ejemplos de contenido, guías de estilo, personalización requerida, restricciones debido a datos heredados, etc.
- El *diseño de datos* se realiza mediante un modelado estructural identificando entidades y relaciones. WebML permite la utilización en esta etapa de modelos E-R (*Entity-Relationship*) o modelos de clases UML.
- En la etapa de *diseño del hipertexto* se realiza el modelado de la estructura del hipertexto, mediante la construcción de dos modelos: el modelo de composición, en el que se identifican qué páginas componen el hipertexto y con qué contenido; y el modelo de navegación que expresa cómo se enlazan las páginas para formar el hipertexto. Para estos y los demás modelos propuestos, WebML define una notación propia.
- La etapa de *diseño de la presentación*, implica la realización del modelo de presentación en el que se representa la apariencia gráfica de las páginas.
- En la etapa de *diseño de usuarios y de grupos* se realiza un modelado de usuarios, donde se identifican los diferentes perfiles de usuarios y grupos y sus necesidades de personalización.
- Finalmente, el *diseño de personalización* permite la definición de perfiles para el acceso de los grupos de usuarios identificados.

A pesar de que WebML se creó inicialmente para el diseño de aplicaciones Web intensivas en datos, ésta es, sin duda, una de las metodologías que más esfuerzos de adaptación ha realizado en la necesidad de dar soporte al desarrollo

de aplicaciones orientadas a servicios. WebML ha sido extendido para dar soporte al desarrollo de aplicaciones que integran, tanto servicios Web (Brambilla *et al.*, 2003b; Manolescu *et al.*, 2005), definiendo nuevas primitivas para la representación de éstos en el modelo de hipertexto, como también procesos de negocios (Brambilla, 2003; Brambilla *et al.*, 2003a; Brambilla *et al.*, 2006), añadiendo para ello una nueva etapa de análisis de los procesos y extendiendo el modelo estructural y de hipertexto para la captura de procesos.

En el caso de WebML, los autores no especifican explícitamente el paradigma de desarrollo utilizado. Sin embargo, por algunos de sus modelos, como el modelado orientado a objetos en la etapa del modelado conceptual, y por las características de los modelos propuestos para el diseño del hipertexto, que se realiza a través de la identificación de objetos (páginas) que componen la aplicación Web, se puede decir que WebML utiliza los principios del paradigma orientado a objetos.

Teniendo en cuenta el criterio establecido en esta Tesis Doctoral, WebML utiliza un enfoque orientado a objetos para la construcción del hipertexto de las aplicaciones Web. Como en el caso de la metodología anterior, WebML propone la construcción del hipertexto de la aplicación, principalmente, en base al modelado estructural de la misma, aunque también se basa al modelado de procesos que realiza. Además, en este caso se definen también algunos tipos de enlaces que no derivan directamente de las relaciones del modelo estructural ni del proceso, sino que son obtenidas de una manera libre.

WebML define un lenguaje propio y específico de dominio para el diseño de aplicaciones Web, y solo propone la utilización de un modelo de clases UML en la etapa del diseño conceptual. Recientemente se ha definido un perfil UML para WebML (Moreno *et al.*, 2006), aunque este sólo incluye extensiones para la representación de las primitivas básicas de WebML; así, por ejemplo, no se ocupa de las primitivas definidas por WebML para la representación de servicios Web y procesos de negocios.

WebML está soportado por la herramienta de desarrollo WebRatio (WebRatio). Dicha herramienta permite el desarrollo dirigido por modelos de aplicaciones Web, utilizando para ello el lenguaje WebML, y la generación de código compatible con la arquitectura J2EE a partir de tales modelos. Aunque dicha herramienta utiliza principios de MDA, puesto que se basa en el desarrollo dirigido por modelos y la generación de código automática, WebML no se define como una aproximación basada en MDA. WebML no distingue entre modelos independientes o dependientes

de plataforma, ni tampoco se centra en la definición de transformaciones entre los modelos que propone. Sin embargo, teniendo en cuenta las características de los modelos propuestos por WebML se puede decir que éstos coinciden con el nivel independiente de plataforma de MDA.

2.2.4 OOWS: Object Oriented Web-Solutions

OOWS (Pastor *et al.*, 2001) es una metodología que se centra en el desarrollo de sistemas de información hipermedia y aplicaciones de comercio electrónico en ambientes Web. OOWS esta basada en los modelos conceptuales de OO-Method (Gómez *et al.*, 2000) y propone la utilización del diseño navegacional, junto con el modelado conceptual, como punto de partida para la generación de código de aplicaciones Web.

OOWS propone un proceso de desarrollo de dos grandes pasos: especificación del sistema y desarrollo de la solución. Dentro de la especificación del sistema se contempla una etapa de *recolección de requisitos* y la realización de los siguientes modelos: *modelo de dominio*, *modelo dinámico*, *modelo funcional* y *modelo de navegación*. Como parte del desarrollo de la solución, se contempla el desarrollo de la aplicación en tres capas: datos, lógica de negocio y presentación.

- La etapa de *recolección de requisitos* propone la especificación del comportamiento del sistema a través de diagramas de casos de uso.
- En el *modelo de objetos* se define la estructura estática (clases, con sus atributos y operaciones, y relaciones entre ellas) del dominio de la aplicación. Este modelo se representa por medio de un diagrama de clases UML.
- El *modelo dinámico* se especifica a través de un diagrama de transición de estado y describe la "vida" de los objetos, representando el comportamiento de cada clase en el sistema.
- El *modelo funcional* captura la semántica asociada a los cambios de estado de los objetos, motivado por la ocurrencia de servicios.
- El *modelo navegacional* se compone de un mapa navegacional que se representa como un grafo dirigido. Los nodos del grafo son contextos navegacionales y los enlaces representan la navegación de un contexto a otro. Cada contexto navegacional representa un punto de vista de los usuarios sobre los objetos del modelo de objetos.

Como su nombre indica, OOWS utiliza los principios del paradigma orientado a objetos para el desarrollo de aplicaciones Web. En cuanto al proceso de desarrollo del hipertexto, se puede decir que OOWS utiliza un enfoque similar al OOHDM, orientado también a objetos. Así, de manera similar a OOHDM, OOWS propone construir el hipertexto de la aplicación Web en base a los contextos navegacionales, los que representan clases del dominio de la aplicación y relaciones entre ellas.

OOWS utiliza notación UML para la representación de sus modelos, aunque también propone la realización de diagramas que utilizan otra notación como el diagrama de *ConcurTaskTrees* (Paterno *et al.*, 1997), que sirve para realizar taxonomías de tareas (Valderas *et al.*, 2005). Hasta donde se conoce, los autores de OOWS no han definido un perfil UML que reúna los elementos de modelado propuestos por la metodología.

Con respecto a su relación con MDA, OOWS plantea una estrategia basada en transformaciones de grafos para la definición y automatización de transformaciones de modelos (Valderas *et al.*, 2005). En concreto, se propone automatizar la transformación necesaria para la obtención del modelo navegacional de nivel PIM a partir del diagrama de *ConcurTaskTrees*, en el que se representa una taxonomía de tareas que realiza el usuario, y una descripción de tales tareas, en la que se modela la posible interacción del sistema con los usuarios. Para automatizar las transformaciones entre grafos, OOWS propone utilizar la herramienta *AGG Tool* (Taentzer, 2000). Sin embargo, dado que no todos los modelos definidos por OOWS pueden ser representados como grafos, sus autores define guías para transformar sus modelos en grafos, y viceversa, basándose para ello en transformaciones entre documentos XML.

2.2.5 OO-H: Object-Oriented Hypermedia Method

Esta propuesta define un método orientado a objetos para el desarrollo de interfaces Web (Gómez *et al.*, 2001). OO-H se basa también en los modelos conceptuales definidos en OO-Method (Gómez *et al.*, 2000), y propone un conjunto de modelos de diseño y constructores propios los cuales han sido especialmente establecidos para agilizar el proceso de desarrollo de interfaces Web.

OO-H propone un proceso de desarrollo iterativo e incremental que consta de cuatro etapas fundamentales: *análisis de requisitos*; *ingeniería*, en la que se contemplan todas las actividades relacionadas con el *análisis y diseño del dominio*, *análisis y diseño de la navegación* y *diseño de la presentación*; *construcción y adaptación*; y, *evaluación del cliente*.

Como resultado de algunas de estas actividades se generan los siguientes modelos:

- *Modelo de requisitos funcionales*, que se representa a través de un diagrama de casos de uso;
- *Modelo de dominio*, que se representa a través de un diagrama de clases.
- *Modelo de análisis de navegación*, que se representa a través de una diagrama llamado diagrama de navegación semántica.
- *Modelo de diseño de navegación*, representado a través de un diagrama llamado diagrama de acceso navegacional.
- *Modelo de diseño de presentación*, que se representa a través de dos diagramas llamados diagrama de presentación y diagrama de diseño visual.

En (Koch *et al.*, 2004) se ha propuesto una extensión a OO-H para el modelado de procesos de negocio, que comparte los modelos de análisis con la metodología de UWE. Así, OO-H propone realizar las siguientes actividades: *modelado conceptual del proceso* y el *modelado del flujo de proceso*, para el análisis de los procesos de negocio; *refinamiento del modelo conceptual del proceso*; *refinamiento del modelo del flujo de proceso*, identificando subactividades en el proceso y agregando nuevos estereotipos; *construcción del modelo de navegación*, a través del diagrama de acceso navegacional; y, finalmente, *diseño de la vista de presentación*.

OO-H define un enfoque orientado a objetos para el desarrollo de aplicaciones Web. Con respecto al enfoque utilizado para la construcción del hipertexto, se puede decir que OO-H utiliza un enfoque orientado a procesos, dado que define guías para derivar el hipertexto de la aplicación (páginas Web y enlaces entre ellas) a partir del modelo flujo de proceso.

Esta metodología utiliza UML de manera parcial, puesto que también propone utilizar algunos diagramas propios, como el diagrama de acceso navegacional.

El proceso de desarrollo de OO-H está soportado por una herramienta para la generación de interfaces de aplicaciones Web llamada VisualWADE (VisualWADE). OO-H se define como una aproximación dirigida por modelos y, aunque los autores no especifican los niveles de abstracción propuestos, se puede decir que sus

modelos se corresponden con el nivel de modelado independiente de plataforma de MDA.

2.2.6 WSDM: Web Site Design Method

La propuesta de (De Troyer y Leune, 1998) llamada *Web Site Design Method* (WSDM), define un proceso de diseño para aplicaciones Web basado en los requisitos de diferentes grupos de usuarios del sistema.

El proceso de diseño de WSDM incluye cuatro etapas básicas: *modelado de usuario*, *diseño conceptual*, *diseño de la implementación*, y finalmente, *implementación*.

- En la etapa de *modelado de usuario* se analizan y clasifican los distintos tipos de usuarios de la aplicación.
- En etapa de *diseño conceptual* se realiza el modelado de objetos y modelado navegacional teniendo en cuenta los diferentes tipos de usuarios identificados y sus necesidades de navegación
- El *diseño de la implementación* plantea el diseño de la apariencia final de la aplicación en base a los diseños navegacionales identificados en la etapa anterior.
- Por último, durante la etapa de *implementación* se desarrolla la aplicación Web en el entorno de implementación seleccionado.

WSDM ha definido también una extensión para dar soporte al modelado de procesos complejos (De Troyer y Casteleyn, 2003). En ella se amplía la etapa de diseño conceptual incluyendo un modelado de las tareas que necesita llevar a cabo cada tipo de usuario. Dicha actividad se realiza utilizando una notación específica para la descripción de actividades de usuario, llamada *ConcurTaskTrees* (Paterno *et al.*, 1997), que las define utilizando una estructura de árbol.

WSDM propone un enfoque centrado en el usuario para el desarrollo de aplicaciones Web y, aunque utiliza los principios del paradigma orientado a objetos, todo su proceso de desarrollo está centrado en los requisitos de los distintos tipos de usuarios de la aplicación. En concreto, el modelo principal para la obtención del hipertexto del SIW es el modelo de tareas, en el que se representan las tareas que realiza un usuario, por lo que se dice que utiliza un enfoque orientado a tareas.

WSDM sólo utiliza UML para el modelado de objetos, y define una notación propia para el resto de sus modelos.

Aunque los modelos propuestos por WSDM se pueden corresponder con el nivel de modelado independiente de plataforma de MDA, la propuesta de WSDM no define una aproximación basada en MDA para el desarrollo de aplicaciones Web. Por ello, sus autores no especifican distintos niveles de modelado del sistema acordes a los niveles de abstracción propuestos por MDA, ni tampoco definen reglas para la transformación entre ellos.

2.2.7 Autoweb: Una metodología para el desarrollo de aplicaciones Web

Autoweb (Fraternali y Paolini, 1998; Fraternali y Paolini, 2000) es una metodología para el desarrollo de aplicaciones Web, que integra los modelos de diseño de aplicaciones de hipermedia con la generación automática de código. Esta metodología define una notación de modelado, llamada HDM-lite, que hereda conceptos de la metodología HDM (Garzotto *et al.*, 1993) y de los modelos E-R.

HDM-lite define tres esquemas para el diseño de aplicaciones Web: un *esquema de la estructura*, un *esquema de navegación* y un *esquema de presentación*.

- El *esquema de la estructura*, describe las propiedades de los objetos básicos que componen la aplicación.
- El *esquema de navegación* especifica las acciones disponibles para moverse de un objeto a otro y los caminos de acceso para alcanzar los distintos objetos de la aplicación.
- El *esquema de presentación* define la manera en que los objetos de la aplicación se presentarán a los usuarios.

Esta metodología utiliza el paradigma orientado a objetos para el desarrollo de aplicaciones Web y su enfoque para la construcción del hipertexto es también orientado a objetos, puesto que, como se centra en el desarrollo de aplicaciones Web intensivas en datos, el hipertexto se genera en base a las clases del dominio de la aplicación.

Autoweb no utiliza UML puesto que como se ha dicho anteriormente esta metodología define una notación propia como lenguaje de modelado llamada HDM-lite.

Autoweb define técnicas de transformación de los esquemas conceptuales a modelos relacionales, y de éstos a páginas físicas que constituirán la aplicación Web. Para ello, los autores proveen una herramienta, llamada Autoweb System,

que soporta la definición de aplicaciones utilizando HDM-lite y la implementación automática de las transformaciones mencionadas.

2.2.8 W2000: Un marco para el diseño de aplicaciones Web

Esta propuesta (Baresi *et al.*, 2000; Baresi *et al.*, 2001) define un marco para el diseño de aplicaciones Web que integra UML, como notación estándar para el modelado en ingeniería del software, con los principios de HDM (Garzotto *et al.*, 1993), como notación para el modelado de sistemas hipermedia; definiendo estereotipos para la representación de elementos de hipermedia, propios de HDM, a través de modelos UML.

Su proceso de desarrollo se divide en las etapas de *análisis de requisitos*, *diseño de la evolución de estados*, *diseño de hipermedia*, *diseño funcional* y *diseño de la visibilidad*.

- En la primera etapa, *análisis de requisitos*, consta de dos actividades: análisis de requisitos funcionales y análisis de requisitos navegacionales, que consisten en determinar las principales necesidades de información y de navegación por parte de los diferentes usuarios de la aplicación. Ambas actividades se especifican a través de modelos de casos de uso UML.
- La etapa de *diseño de la evolución de estados* representa la evolución que sufre el contenido de la aplicación según ocurre la navegación. Esta etapa es requerida solo en aplicaciones que poseen un comportamiento complejo y se realiza a través de un diagrama de estados UML.
- La etapa de *diseño de hipermedia* incluye una actividad de diseño de la información y otra de diseño de navegación. La primera diseña y organiza el contenido de la aplicación, y la segunda define cómo los usuarios pueden acceder y navegar en el sistema. En ambos casos, representa el diseño mediante diagramas de clases UML.
- La etapa de *diseño funcional* integra el diseño hipermedia con el diseño de evolución de estados, especificando las principales operaciones que realizan los usuarios en la aplicación. Para su representación utiliza un diagrama de secuencia UML.
- Finalmente, la etapa de *diseño de la visibilidad*, consiste en la especificación de qué información y estructuras de navegación deben ser visibles por cada uno de los usuarios.

W2000 utiliza el paradigma orientado a objetos para el desarrollo de aplicaciones Web. En cuanto al enfoque para el desarrollo del hipertexto, se puede decir que esta metodología utiliza un enfoque orientado a objetos, puesto que el diseño de la estructura navegacional de la aplicación se realiza en base a las necesidades de información (objetos del dominio y sus atributos) de los usuarios de la aplicación.

W2000 utiliza UML para la representación de sus modelos, sin embargo, hasta donde se conoce esta metodología no ha definido un perfil que reúna los elementos de modelado propuestos.

Los autores de W2000 no han discutido su adaptación a MDA, sin embargo es evidente que los modelos propuestos se corresponden con el nivel de modelado independiente de plataforma de dicha arquitectura.

2.2.9 Otras metodologías de Desarrollo de SIW

Además de las metodologías más relevantes para el desarrollo del SIW, las cuales se han presentado hasta el momento, existen otras propuestas que abordan el diseño de aplicaciones Web, y que por tanto se han ocupado también de definir modelos y procesos para el desarrollo de las mismas.

En primer lugar, se destaca la propuesta de (**Díaz et al., 2005**) llamada ***Ariadna Development Method*** (ADM), que se centra en el desarrollo de sistemas hipermedias y Web. Esta propuesta plantea un proceso interactivo y centrado en el usuario para el modelado, de manera completa e independiente de plataforma, de sistemas hipermedias y Web. El proceso de diseño propuesto por ADM consta de tres fases, cada una de las cuales incluye un conjunto de actividades que tienen como objetivo la creación de uno o más productos: *diseño conceptual*, que tiene como objetivo la descripción de las características estáticas y dinámicas del sistema desde una perspectiva abstracta; *diseño detallado*, en la que se especifican las unidades de implementación de manera concreta; y *evaluación*, en la que se realiza la verificación de la validez y utilidad de las soluciones propuestas involucrando en ello al usuario o destinatarios del producto.

Por otro lado, se destaca uno de los trabajos más recientes para el modelado y desarrollo de aplicaciones Web en el marco de MDA (**Muller et al., 2005**). En este trabajo se definen modelos independientes de plataforma para el desarrollo de aplicaciones Web en el contexto de MDA. En concreto, se propone la representación de las aplicaciones Web en base a tres vistas: negocio, hipertexto y presentación. Como *modelo de negocio* se propone un modelo de clases en el que se representan

los conceptos u objetos del negocio. Para el *modelado del hipertexto*, se define un modelo de composición de páginas Web a partir de varios fragmentos de información y para la especificación entre páginas. Y finalmente, el *modelo de presentación* contiene detalles de la apariencia gráfica de las aplicaciones. La principal aportación de esta propuesta es que, debido al alto nivel de abstracción de los elementos de modelado que se proponen, el modelo de composición y navegación de páginas Web presentado, puede ser aplicado en varias de las metodologías de desarrollo Web vistas anteriormente.

2.2.10 Comparación de Metodologías para el Desarrollo de SIW

Para finalizar el estudio de las propuestas metodológicas para el desarrollo de SIW, se ha realizado un resumen de las principales características analizadas en cada una de ellas que se muestra en la Tabla 2.1. Dicha tabla incluye además una columna correspondiente al método que se presenta en esta Tesis Doctoral, SOD-M, con el fin de mostrar sus principales diferencias con respecto al estado de la cuestión actual.

A continuación se realiza una breve descripción sobre cada una de las características analizadas en las metodologías, destacando, en cada caso, las principales aportaciones de este trabajo de Tesis Doctoral.

Con respecto a los modelos propuestos por cada una de las metodologías, se pueden destacar dos aportaciones del método propuesto en esta Tesis Doctoral. Por un lado, la **definición de un modelo de negocio de alto nivel e independiente de computación**, que en el caso de SOD-M, permite la representación de los actores que participan en el negocio y de la manera en que éstos colaboran, facilitando además la comprensión de los servicios que deben ser ofrecidos a los usuarios del SIW, así como las responsabilidades de cada uno de los participantes del negocio en el ofrecimiento de los mismos. Cabe remarcar aquí que, en general, todas las metodologías analizadas definen actividades para la captura de requisitos, utilizando para ello modelos de casos uso (OOHDM, UWE, OO-H, W2000) o diagramas de tareas (OOWS, WSDM) en los que se representan las interacciones del usuario con el sistema. Sin embargo, hay que destacar que, desde el punto de vista de MDA, los modelos independientes de computación no se utilizan para representar detalles del sistema en sí, sino del dominio de la aplicación, de modo que sirva como puente entre los expertos del negocio y los desarrolladores del sistema, característica que posee el modelo de negocio propuesto por SOD-M y no poseen ninguna de las otras metodologías analizadas. Por otro lado, se destaca como aportación de esta Tesis Doctoral, la **definición de modelos para la**

representación de servicios, así como para el modelado de la composición de los mismos. La metodología WebML, que como se ha dicho es una de las que más esfuerzos de adaptación al paradigma SOC ha realizado, también se ha ocupado de la representación de servicios Web y de la composición de éstos. Sin embargo, en este caso, no se definen modelos específicos, sino que sólo se proponen nuevos elementos de modelado para la representación de servicios Web y de su composición, que se agregan a los modelos ya utilizados en la etapa del diseño del hipertexto. Respecto de los demás modelos propuestos por las principales metodologías de desarrollo de SIW, vemos que la mayoría de ellas (OOHDM, UWE, WebML, OOWS, OO-H y W2000) proponen la utilización de los modelos típicos para el modelado del comportamiento desde un enfoque orientado a objetos, tales como modelos de casos de usos, modelos de objetos, y también modelos de procesos, tales como diagramas de actividad.

En cuanto al paradigma de desarrollo, hay que remarcar que todas las metodologías de desarrollo de SIW analizadas utilizan los principios del paradigma orientado a objetos para la construcción de todo el SIW. Sin embargo, como puede verse en la Tabla 2.1, **SOD-M propone un nuevo enfoque orientado a servicios para el desarrollo de los SIW**, es decir utilizando servicios como elementos básicos para la construcción del mismo. Esta característica, que diferencia a SOD-M de las metodologías de desarrollo Web tradicionales, permite a los desarrolladores de software aprovechar al máximo los beneficios del nuevo paradigma de computación SOC, a la vez que facilita la implementación de las aplicaciones Web utilizando las tecnologías Web actuales.

Respecto del enfoque utilizado para la construcción del hipertexto, se puede también destacar una aportación del método propuesto en esta Tesis Doctoral. Como se ha dicho anteriormente, en este punto se pretendía analizar cómo, las diversas metodologías, proponen construir el hipertexto de los SIW, y para ello se proponía un criterio para distinguir un enfoque de otro, es decir, tener en cuenta el modelado de qué elementos del sistema es determinante para la construcción del hipertexto del SIW. Como puede verse en la Tabla 2.1, el enfoque más utilizado para la construcción del hipertexto de los SIW, es el orientado a objetos. Esto muestra que, a diferencia de SOD-M, la mayoría de las metodologías de desarrollo de SIW (OOHDM, UWE, WebML, OOWS, Autoweb y W2000) proponen obtener la estructura básica de la navegación de la aplicación Web (páginas Web y enlace) en base a modelos estructurales (modelos de clases, o de objetos) del dominio de la aplicación. Este enfoque para el desarrollo del hipertexto proviene de los inicios de la Web, en la que las aplicaciones se concebían básicamente como un medio para

publicar información; así, las primeras aproximaciones para el desarrollo de SIW, proponían obtener la estructura navegacional de la aplicación teniendo en cuenta qué datos debían ser mostrados en la Web. Sin embargo, es evidente que, en la actualidad, los usuarios de la Web no sólo esperan información del SIW de una organización, sino más bien la posibilidad de acceder a través de Internet a los diversos servicios que ésta le puede proporcionar. Debido a este cambio, algunas metodologías tales como (UWE, WebML y OO-H, entre otras) han extendido sus procesos de desarrollo, para contemplar también, en el desarrollo del hipertexto, la representación de procesos de negocio. El método propuesto en esta Tesis Doctoral, define también guías para contemplar los procesos de negocios en el hipertexto de los SIW pero además, a diferencia de las metodologías actuales, **define un enfoque para el desarrollo del hipertexto basado en los servicios del SIW**, en el, todo el proceso de modelado y desarrollo de la aplicación se centra en los servicios que serán ofrecidos a los usuarios de la Web. Este enfoque proporciona como ventaja adicional, la posibilidad de construir SIW más intuitivos y fáciles de navegar por parte de los usuarios, dado que se propone construir un modelo de navegación en el que se identifiquen explícitamente tanto los servicios como las rutas para llevar a cabo cada uno de ellos (De Castro *et al.*, 2006b).

En cuanto a la utilización de UML, se puede decir que es una característica compartida por la mayoría de las metodologías de desarrollo de SIW que han sido analizadas, aunque muchas de ellas la integran con notaciones propias, como es el caso de OOHDM, OOWS, OO-H y WSDM. La metodología AutoWeb propone utilizar una notación propia (HDM-lite) para la realización de sus modelos. Por otro lado, la metodología WebML también define una notación propia para la mayoría de sus modelos. Esta característica, es quizás la principal desventaja de esta metodología, puesto que su adopción como técnica de modelado requiere el conocimiento de todos los elementos de modelado propuestos por WebML y el significado de cada uno de éstos. El método que se presenta en este trabajo de investigación, **propone la utilización de UML como notación única para el modelado del SIW (a nivel PIM y PSM), y define además, un perfil UML para permitir el desarrollo orientado a servicios de SIW**. La metodología UWE es una de las que más ha trabajado en la línea de UML y ha definido también un perfil UML para el desarrollo de aplicaciones Web tradicionales.

Muchas de las principales metodologías de desarrollo de SIW han trabajado en la línea de MDA, definiendo modelos para los distintos niveles de abstracción de dicha arquitectura y, en algunos casos, tales como OOHDM, UWE, OOWS, OO-H y Autoweb, definiendo reglas para la transformación entre los mismos. Como se ha

explicado en el capítulo 1, SOD-M es parte de un marco metodológico para el desarrollo de SIW basado en MDA, y por tanto ***define modelos a nivel CIM, PIM y PSM de la arquitectura propuesta por MDA***, a la vez que ***propone transformaciones entre ellos***.

Para finalizar esta sección, es posible resaltar los siguientes puntos como principales aportaciones de SOD-M, respecto de las actuales metodologías de desarrollo de SIW:

- La definición de un ***nuevo enfoque orientado a servicios para el desarrollo de todo el SIW***.
- La ***utilización de UML*** como lenguaje de modelado y la definición de un ***perfil UML específico para el desarrollo orientado a servicios de SIW***.
- La ***integración del método propuesto en el marco de una arquitectura MDA***, proporcionando modelos de nivel CIM, PIM y PSM, así como transformaciones entre ellos.

Algunos de los modelos propuestos en SOD-M, como el modelo de negocio a nivel CIM, son modelos que, una vez demostrada su eficacia como técnica de modelado a través de uso en diversos casos de estudio, han sido incluidos como parte de esta propuesta. Tales modelos, y otras técnicas propuestas relacionadas con el desarrollo orientado a servicios de SIW se analizan en la siguiente sección.

Tabla 2.1. Resumen de las principales propuestas para el Desarrollo de SIW

			Propuestas para el Desarrollo de SIW								
			OOHDM	UWE	WEBML	OOWS	OO-H	WSDM	Autoweb	W2000	SOD-M
1. Modelos propuestos	Modelo de Negocio		x	x	x	x	x	x	x	x	✓
	Modelo de Caso de Uso		✓	✓	x	✓	✓	x	x	✓	✓
	Modelo de Procesos		✓	✓	✓	✓	✓	✓	x	✓	✓
	Servicios	Modelo de Servicios	x	x	✓ (en el Hipertexto)	x	x	x	x	x	✓
		Modelo de Composición de Servicios	x	x	✓ (en el Hipertexto)	x	x	x	x	x	✓
2. Paradigma de desarrollo			OO	OO	OO	OO	OO	OO	OO	OO	OS
3. Enfoque para la construcción del hipertexto			Orientado a <i>Objetos</i>	Orientado a <i>Objetos</i>	Orientado a <i>Objetos</i>	Orientado a <i>Objetos</i>	Orientado a <i>Procesos</i>	Orientado a <i>Tareas</i>	Orientado a <i>Objetos</i>	Orientado a <i>Objetos</i>	Orientado a <i>Servicios</i>
4. Notación de modelado	Utiliza UML		✓ (Parcialmente)	✓	x	✓ (Parcialmente)	✓ (Parcialmente)	✓ (Parcialmente)	x	✓	✓
	Define Perfil UML		x	✓	✓ (Parcialmente)	x	x	x	x	x	✓
5. Aproximación basada en MDA	Niveles de Abstracción Cubiertos		PIM, PSM	PIM, PSM	PIM	PIM	PIM	PIM	PIM	PIM	CIM, PIM, PSM
	Transformaciones entre Modelos		✓ (Parcialmente)	✓	x	✓ (Parcialmente)	✓ (Parcialmente)	x	✓	x	✓

2.3 Propuestas para el Desarrollo Orientado a Servicios

El paradigma de computación orientada a servicios surge en respuesta en un cambio fundamental de la manera en que las empresas dirigen su negocio. Tal como afirma (Curbera *et al.*, 2003), hoy en día la mayoría de las grandes empresas están sustituyendo sus sistemas completamente centralizados por redes del negocio en las que cada participante provee a los demás de servicios especializados.

Este nuevo entorno, ha dado lugar a la especificación de nuevas técnicas y métodos para el desarrollo de aplicaciones en la Web, y como ya hemos visto, también ha motivado cambios y adaptaciones en las metodologías de desarrollo de SIW tradicionales.

Analizando la literatura relacionada con el desarrollo de sistemas orientados a servicios, se ha detectado que, en la actualidad, no existen propuestas metodológicas que definan un enfoque orientado a servicios para abarcar todo el proceso de desarrollo de software, sino más bien, una gran cantidad de trabajos, tales como técnicas de modelado o métodos, que se ocupan de ciertos aspectos del desarrollo de los mismos. Así, por ejemplo, existen diversas propuestas para el desarrollo de servicios Web, para el desarrollo de la composición de servicios Web, o propuestas para el desarrollo de procesos de negocios que utilizan servicios Web.

En esta sección se presentará un análisis de tales propuestas, las cuales se describirán resaltando sus principales características, así como sus aportaciones y limitaciones a la hora de ser utilizadas en un proceso de desarrollo de SIW. A continuación se describen los criterios que servirán para la evaluación y comparación de las mismas:

- 1. Elementos que modela:** en este caso, y teniendo en cuenta las actuales propuestas, se propone una clasificación basada en dos elementos fundamentales, ambos muy relacionados con el desarrollo de aplicaciones orientadas a servicios: el negocio y los servicios Web. Con respecto a las técnicas que se utilizan para el modelado de negocios, en este trabajo se distinguirá entre: propuestas que se ocupan del *modelado de negocio* en general, es decir, aquellas que se ocupan del modelado de los miembros que participan en el negocio, sus relaciones y los servicios que ofrecen a los demás, más que en los procesos del mismo; y, propuestas que se ocupan del *modelado de procesos de negocio*, en las que generalmente se utilizan técnicas de representación de flujos de trabajos, tales como

diagramas de actividad. Con respecto a las técnicas que se utilizan para el modelado de servicios Web, se distinguirán entre: propuestas que se ocupen del *modelado de servicios Web* de manera individual, y aquellas que se ocupen del *modelado de la composición de servicios Web*.

2. Correspondencia con niveles de abstracción de MDA: para cada una de las técnicas o métodos que se analicen, se indicará el nivel de abstracción de la arquitectura de MDA con el que se corresponde. Es decir, se analizará si la técnica permite obtener *modelos independientes de computación* (CIM), *modelos independientes de plataforma* (PIM), o *modelos específicos de plataforma* (PSM).

3. Notación utilizada: por último, se indicará la notación que se utiliza en cada una de las propuestas analizadas.

En el apartado 2.3.8 de esta sección se muestra una comparación de las propuestas que se analizan en base a éstos criterios, así como una tabla resumen (Tabla 2.2). Dicha tabla incluye además una columna para SOD-M, el método propuesto en esta Tesis Doctoral, para facilitar su comparación con las propuestas estudiadas.

2.3.1 Método e³-value

Este método (Gordijn y Akkermans, 2003) se centra en el modelado de negocio y está específicamente orientado al diseño de redes de negocios. El método resulta especialmente útil para la comprensión de una idea de negocio, lo que se consigue mediante la construcción de un modelo de alto nivel llamado *modelo de valor* (*value model*).

E³value se centra en la identificación de qué objetos de valor (bienes, dinero o servicios) son creados y proporcionados por cada uno de los miembros que participan en un negocio, y qué miembros del negocio están interesados en dichos objetos de valor. Para ello, define un conjunto de conceptos, procedentes de las ciencias de la administración de empresas y del marketing, y propone una notación propia para la representación de los mismos.

El método define, además, guías para la representación en el *modelo de valor* de caminos de dependencias. Los caminos de dependencia muestran los intercambios de objetos de valor que tiene lugar ante un determinado estímulo o necesidad de un consumidor final perteneciente al negocio. Esto permite, además, el cálculo del rendimiento económico de cada uno de los miembros del negocio, mejorando también la comprensión de la idea del mismo.

Por su alto nivel de abstracción en el modelado de negocio, este modelo se corresponde con el nivel de abstracción CIM de la arquitectura de MDA.

2.3.2 Método de modelado WIED

Esta propuesta (Tongrungrrojana y Lowe, 2004) define un modelo llamado WIED (*Web Information Exchanged Diagram*), que actúa como nexo entre el modelado de negocio de alto nivel, para lo que se propone la utilización del método *e³-value*, y el diseño de bajo nivel de SIW, utilizando para ello la metodología WebML.

El modelo WIED es también un modelo de negocio, que se ocupa de la representación de la información que se intercambia entre los diferentes participantes del negocio. El método define guías para la transformación del modelo de negocio al modelo WIED, y de este a un modelo de hipertexto de WebML.

Aunque el modelo WIED es también un modelo de negocio, este modelo se corresponde con el nivel PIM de la arquitectura de MDA, puesto que en el ya se representan características y funcionalidades propias del sistema que se desea construir.

Los autores del modelo WIED han definido una notación propia para la representación del mismo, aunque posteriormente han propuesto también una extensión UML para su representación.

La principal desventaja de este método es la utilización de WebML como técnica de modelado de bajo nivel, lo que limita las posibilidades de utilización del modelo WIED en otras metodologías de desarrollo de SIW distintas a WebML, principalmente teniendo en cuenta que WebML no utiliza una notación estándar, sino la suya propia.

2.3.3 Business Process Modeling Notation - BPMN

La propuesta BPMN (BPMI, 2004) es una iniciativa del OMG (*Object Management Group*) y el BPMI (*Business Process Management Initiative*), y define una notación para el modelado de procesos de negocios.

BPMN propone un diagrama, llamado *Business Process Diagram* (BPD), que facilita el modelado de flujos de procesos a través de elementos que resultan familiares a los analistas de negocios. Un BPD permite la representación de: eventos que ocurren cuando comienza un proceso, de los procesos que se ejecutan y de los resultados de procesos. BPMN incluye cuatro categorías básicas de elementos gráficos: *Flow Objects*, que son los elementos básicos del modelo;

Connecting Objects, utilizados para conectar los elementos básicos del modelo; *Swimlanes*, que son particiones para la separación de las actividades; y finalmente, *Artifacts*, que representan objetos de datos, comentarios o agrupaciones, y que se agregan a los diagramas para una apropiada representación del contexto del procesos de negocio.

BPMN podría ser utilizado en los niveles CIM, PIM o PSM de la arquitectura de MDA, según sea el grado de abstracción con el que se quiera representar un proceso de negocio.

Una limitación de BPMN es la definición de una notación gráfica propia, lo que reduce sus posibilidades de utilización y adopción como método de modelado. Sin embargo, debido a su reciente adopción por parte del OMG, esta propuesta va camino de convertirse en una notación estándar para el modelado de procesos de negocios.

Por otro lado, una de las ventajas de BPMN es la definición de guías para la generación de código ejecutable BPEL (Andrews *et al.*, 2003) a partir de un BPD, aunque ya se han demostrado que existen varios problemas en esta generación. La razón de estos problemas es que no todos los procesos que se representan en un BPD tienen una correspondencia directa en código BPEL, fundamentalmente debido a ciertas limitaciones de BPEL, como pueden ser los problemas en el manejo de ciclos (Ouyang *et al.*, 2006), o bien, los problemas en el manejo de procesos de negocios que requieren la intervención de un usuario (López *et al.*, 2006b).

2.3.4 Perfil UML para Business Process Execution Language - UML4BPEL

Esta propuesta (Mantell, 2005) define un perfil UML para el modelado de procesos de negocios y además, describe las transformaciones necesarias para obtener código ejecutable BPEL (Andrews *et al.*, 2003).

La propuesta define estereotipos para la representación de un *proceso BPEL*, que se representa a través de una clase y cuyos atributos corresponden al estado del proceso. El comportamiento del proceso, se representa utilizando un diagrama de actividad UML. En este modelo las actividades son también estereotipadas para la representación actividades de *recieve*, *invoke*, *reply* y otras que son propias del estándar BPEL.

Este modelo se corresponde con el nivel PSM de la arquitectura de MDA. Sin embargo, puesto que es un perfil UML para BPEL, el modelo conserva las características propias de BPEL, y por tanto, permite modelar procesos que se

componen exclusivamente de servicios Web. Por ello, de cara a poder integrar esta propuesta en el marco del desarrollo de sistemas, es necesario contar con algún tipo de técnicas o métodos que permita primero identificar, tanto los servicios Web como los procesos de composición de los mismos, a un nivel de abstracción más alto.

2.3.5 Perfil UML para Web Services Description Languages - UML4WSDL

En esta propuesta (Parikh y Pradhan, 2003) se define un perfil UML para la representación de servicios Web basada en el estándar WSDL (W3C, 2004a). En ella se especifican estereotipos para el modelado de servicios Web a nivel PIM y PSM de la arquitectura de MDA.

A nivel PIM se propone la representación de los siguientes elementos del metamodelo de WSDL: *Definitions*, *PortTypes*, *Operations*, *Messages*, *Parts* y *PartType*; y a nivel PSM los elementos: *Service*, *Ports*, *Bindings*. De este modo, utilizando los estereotipos propuestos es posible modelar un servicio Web que podría ser fácilmente transformado a código WSDL.

El principal problema de esta propuesta es la representación de servicios Web a nivel PIM de la arquitectura MDA, proponiendo la representación en este nivel de conceptos tales como *PortTypes*, *Operations*, *Messages*. Los modelos de nivel PIM se utilizan para el modelado de la funcionalidad del sistema desde una punto de vista independiente de plataforma, y, cuando se habla de servicios Web, ya se habla de una plataforma concreta. Por ello, en este caso, sería más apropiado realizar el modelado de los servicios, como funcionalidades, a nivel PIM, y luego agregar el nivel de detalle necesario para la representación de tales servicios como servicios Web en el nivel PSM.

2.3.6 Método para el desarrollo dirigido por modelos de servicios Web - MDWSD (Model-Driven Web Service Development)

Este método se define como parte de un marco para desarrollo dirigido por modelos de servicios Web (Baïna *et al.*, 2004) y comprende tanto en el desarrollo de servicios Web como en la composición de los mismos.

Esta propuesta define el concepto de *especificación externa* de un servicio Web, que se refiere a la descripción de un servicio Web incluyendo no sólo la interfaz, sino también el protocolo de negocio soportado por el mismo, es decir, la especificación de qué secuencias del mensaje serán soportadas por el servicio Web

y el orden en que se invocarán las operaciones del mismo. En concreto, el método define cómo a partir de una *especificación externa* de un servicio Web, es posible generar automáticamente descripciones completas y ejecutables de un servicio que no solo implementa las operaciones propias del mismo sino que se ocupa también del mantenimiento del estado de la conversación. Para el modelado de la conversación los autores proponen un modelo llamado *Service Conversation Model*, en el que se representa la secuencia de mensajes que serán intercambiados entre un cliente y el servicio Web cuando ocurre una invocación al servicio. Para la representación de este modelo se utiliza una notación propia.

Esta propuesta se corresponde con el nivel PSM de la arquitectura de MDA, puesto que se centra en el desarrollo e implementación de servicios Web.

El principal problema de esta propuesta, a la hora de ser utilizada en el proceso de desarrollo de SIW, es la alta especificidad de la misma, así como la definición de una notación propia y no estándar.

2.3.7 Service Components: Un marco para el desarrollo de la composición de servicios Web

Esta propuesta (Yang y Papazoglou, 2004) define un marco para el desarrollo de la composición de servicios, que permite gestionar el ciclo de vida completo de la composición de servicios Web.

Esta propuesta no define un modelo concreto, sino un conjunto de tareas de que deben ser llevadas a cabo para el desarrollo de servicios compuestos. Tales tareas se agrupan en cinco fases: *fase de planificación*, en la que se determinan que operaciones de servicios necesitan ser localizadas y compuestas para satisfacer un necesidad de servicio requerida por un usuario; *fase de definición*, en la que se definen de manera abstracta los servicios compuestos; *fase de programación de la ejecución*, en la que se determina como y cuando los servicios serán ejecutados y se los prepara para la ejecución; *fase de construcción*, en la que se construyen servicios compuestos, concretos y no ambiguos, que están listos para la ejecución; y finalmente, *fase de ejecución*, en la que se implementan servicios compuestos que se enlazan con servicios básicos, según el plan realizado en la etapa de programación de la ejecución.

La principal aportación de esta propuesta es que, a diferencia de las analizadas previamente, define un marco para el desarrollo de la composición de servicios Web en el que se describen claramente las tareas de diseño e implementación necesarias para llevar a cabo la composición. Sin embargo, ésta se

orienta exclusivamente al desarrollo de servicios compuestos, y no al desarrollo de aplicaciones Web en general, por lo que no se tienen en cuenta aspectos tales como el modelado de procesos de negocios, o de aquellas funcionalidades que son necesarias para una aplicación Web y que no pueden ser ofrecidas a través de servicios Web.

2.3.8 Resumen de las principales propuestas relacionadas con el desarrollo orientado a servicios

En la Tabla 2.2 se presenta un resumen de las principales técnicas y métodos estudiados en esta sección. En ella se han destacado, para cada una de las propuestas, el aspecto de modelado que contempla, nivel de abstracción de la arquitectura de MDA con el que se corresponde, y la notación utilizada. Además, dicha tabla incluye una columna correspondiente a SOD-M, a fin de facilitar su comparación con las demás propuestas analizadas.

Como se ha comentado anteriormente, en el análisis de las propuestas relacionadas con el desarrollo de sistemas orientado a servicios, se ha detectado que no existen propuestas metodológicas que contemplen el desarrollo completo de los sistemas, sino más bien, diversas propuestas para el modelado de un aspecto concreto de los mismos. Por ello, como puede verse en la Tabla 2.2, existe una diferencia fundamental entre el método propuesto en este trabajo de Tesis Doctoral y las demás propuestas analizadas en esta sección, y es que ***SOD-M es un método de desarrollo de SIW completo, y que por tanto contempla el modelado de todos los elementos relacionados con el desarrollo de sistemas orientados a servicios.*** Con respecto al modelado de negocio, SOD-M propone la utilización del método e³value para llevar a cabo esta actividad, dado que define guías para la identificación y representación de las principales conceptos de un negocio, a la vez que facilita la identificación de los requisitos del sistema que será implementado, tales como servicios que deben ser ofrecidos a los usuarios o consumidores del mismo. Para el modelado de los demás elementos, procesos de negocio, servicios Web y composición de servicios Web, SOD-M propone la utilización de modelos UML, definiendo los estereotipos necesarios para cada caso. De las demás propuestas analizadas, cabe destacar MDWSD y Service Components, dado que son las únicas que proponen aproximaciones metodológicas para el desarrollo de servicios Web, y en el caso de Service Components, para la composición entre los mismos. Sin embargo, hay que remarcar como diferencias respecto a SOD-M, que dichas propuestas están orientadas al diseño de bajo nivel y

a la implementación de servicios Web, dejando de lado actividades para la identificación y modelado de servicios de alto nivel.

Respecto de la relación de las técnicas o métodos analizados con los niveles de abstracción de MDA, cabe destacar que, debido a que estas propuestas no contemplan el desarrollo completo de sistemas sino, más bien, un aspecto concreto de los mismos, tales propuestas se corresponden en general con un sólo nivel de la arquitectura de MDA. Así, puede verse que el modelado de servicios Web y la composición de los mismos se realizan a un nivel de abstracción más bajo como el nivel PSM; mientras que el modelado de negocio y procesos de negocio se realiza a un nivel de abstracción más alto como los niveles CIM o PIM. Sin embargo, como puede verse en la Tabla 2.2, en el caso de BPMN y UML4BPEL, que definen guías para la generación automática de código BPEL, los modelos propuestos se corresponden también con el nivel PSM. En el caso de SOD-M, que es un método de desarrollo completo y basado, además, en MDA, **se proponen modelos que se corresponden con todos los niveles de la arquitectura MDA**; abarcando de este modo, tanto el modelado de negocio de alto nivel, como el modelado del sistema en los niveles independientes y dependientes de plataforma.

Para finalizar, como se ha explicado previamente, SOD-M **propone la utilización de la notación UML para el modelado del SIW, y adopta la notación definida por e³value, para el caso concreto del modelado de negocios de alto nivel.**

Tabla 2.2. Resumen de propuestas relacionadas con el Desarrollo Orientado a Servicios

		Propuestas para el Desarrollo Orientado a Servicios							
		e ³ -value	WIED	BPMN	UML 4BPEL	UML 4WSDL	MDWSD	Service Components	SOD-M
1. Elementos que modela	Modelado de Negocio	✓	✓						✓*
	Modelado de Procesos de Negocio			✓	✓				✓
	Modelado de Servicios Web					✓	✓	✓	✓
	Modelado de Composición de Servicios Web							✓	✓
2. Correspondencia con niveles de MDA		CIM	PIM	CIM, PIM, PSM	PSM	PIM, PSM	PSM	-	CIM, PIM, PSM
3. Notación utilizada		Propia	Propia y UML	Propia	UML	UML	Propia	- (Define Tareas)	e ³ value y UML

* SOD-M propone la utilización del e³value como método para el modelado de negocio a nivel CIM.

2.4 Conclusiones del capítulo

En este capítulo se ha presentado una revisión del estado de la investigación relacionada con la problemática del desarrollo orientado a servicios de SIW.

En primer lugar, y dado que esta Tesis Doctoral propone una aproximación basada en MDA para el desarrollo de SIW, se ha presentado un resumen de la arquitectura de modelos propuesta por el OMG y se han descrito sus principales características.

Se han revisado también las principales metodologías para el desarrollo de SIW, poniendo especial énfasis en los modelos que éstas proponen para el desarrollo del aspecto del comportamiento de los SIW, y analizando, además, otros aspectos generales de dichas metodologías como su paradigma de desarrollo, la utilización de UML o su adaptación a la propuesta de MDA. Se ha realizado, además, un estudio comparativo de tales propuestas metodológicas donde se han destacado las principales aportaciones del método que se propone en esta Tesis Doctoral.

Por otro lado, se han analizado otras propuestas que, aunque no se han ocupado del desarrollo completo de SIW, sí han realizado importantes aportaciones en el modelado y desarrollo de ciertos aspectos. En este caso se han estudiado las principales técnicas y métodos propuestos para el modelado de negocio, modelado de procesos de negocios, modelado de servicios Web y de composición de los mismos.

En general, tras la revisión del estado actual de la investigación relacionada con la problemática del desarrollo orientado a servicios de SIW, se puede obtener una conclusión principal, y es que en la actualidad *no existen propuestas que definan un enfoque orientado a servicios para el desarrollo completo de sistemas*. En cuanto a las metodologías de desarrollo de SIW existentes, se puede observar que, aunque muchas de ellas han hecho esfuerzos para dar soporte a las nuevas propuestas tecnológicas para la Web tales como servicios Web o automatización de procesos de negocio, en general, dichas metodologías continúan manteniendo sus procesos y enfoques tradicionales para el desarrollo de SIW, realizando, en algunos casos, modificaciones mínimas que generalmente sólo se reflejan en el hipertexto de la aplicación Web.

Como conclusión final del capítulo, se puede destacar una aportación principal de método propuesto en esta Tesis Doctoral respecto de las actuales propuestas relacionadas con el desarrollo orientado a servicios de SIW, y es que **SOD-M es la**

única propuesta que define un enfoque orientado a servicios para el desarrollo completo de SIW en un marco MDA, permitiendo alinear los procesos de negocio de alto nivel con las tecnologías disponibles actualmente para el paradigma SOC.

La Tabla 2.3 presenta, a modo de resumen, las características generales del método que se presenta en esta Tesis Doctoral. En ella se integran tanto los criterios utilizados en la evaluación de las propuestas para el desarrollo de SIW, como los criterios utilizados en la evaluación de las técnicas y métodos relacionados con el desarrollo orientado a servicios. La Tabla 2.3 incluye, además, uno de los métodos presentados en la sección anterior (*e³value*), dado que, por sus ventajas en el modelado de negocio, ha sido incluido como parte de SOD-M.

Tabla 2.3. Características Generales de SOD-M

		SOD-M		
		1. Elementos que modela	2. Correspondencia con niveles de MDA	3. Notación utilizada
1. Modelos propuestos en SOD-M	Modelo de Valor	<i>Modelado de negocio (e³value)</i>	CIM	Notación <i>e³value</i>
	Modelo de Proceso de Negocio	Modelado de procesos de negocios	CIM	UML (Diagrama de Actividad)
	Modelo de Casos de Uso	Modelado de servicios	PIM	UML (Diagrama de Casos de Uso)
	Modelo de Casos de Uso Extendido	Modelado de casos de uso	PIM	UML (Diagrama de Casos de Uso) + Perfil UML SOD-M
	Modelo de Proceso de Servicio	Modelado de proceso de servicios	PIM	UML (Diagrama de Actividad)
	Modelo de Composición de Servicio	Modelado de composición de servicios	PIM	UML (Diagrama de Actividad)
	Modelo de Interfaz de Servicio Web	Modelado de interfaz de servicio Web	PSM	UML (Diagrama de Clases) + Perfil UML SOD-M
	Modelo de Composición de Servicio Extendido	Modelado de composición de servicios Web	PSM	UML (Diagrama de Actividad) + Perfil UML SOD-M
2. Paradigma de Desarrollo		Orientado a Servicios		
3. Enfoque para la construcción del hipertexto		Orientado a Servicios		
4. Notación de modelado	Utiliza UML	SI		
	Define Perfil UML	SI		
5. Aproxim. basada en MDA	Niveles de Abstracción Cubiertos	CIM, PIM, PSM		
	Transformaciones entre Modelos	SI		

***SOD-M: UNA APROXIMACIÓN MDA
PARA EL DESARROLLO ORIENTADOS A
SERVICIOS DE SIW***

En este capítulo se presenta la propuesta de este trabajo de Tesis Doctoral, una aproximación metodológica basada en MDA para el desarrollo orientado a servicios de SIW, denominada SOD-M (*Service-Oriented Development Method*). SOD-M se centra en el desarrollo del aspecto del comportamiento de los SIW y define guías para la integración con los demás aspectos de la arquitectura de MIDAS, que provee un marco basado en MDA para el desarrollo completo de SIW.

El presente capítulo se estructura de la siguiente manera: en la primera sección se presentan las características principales del método propuesto, así como los objetivos del mismo; en la sección 3.2 se presentan los fundamentos de SOD-M, describiendo principalmente los nuevos conceptos de modelado en los que se basa; posteriormente, en la sección 3.3, se presentan los modelos propuestos por SOD-M en correspondencia con los diversos niveles de abstracción de la arquitectura de MDA; dichos modelos se describen de manera detallada en la sección 3.4; en la sección 3.5 se presenta el perfil UML que reúne todos los elementos para el modelado del comportamiento de un SIW utilizando una aproximación orientada a servicios; en la sección 3.6 se presenta el proceso para el modelado del comportamiento, en el que se describe el conjunto de actividades que forman parte del proceso así como las transformaciones entre los modelos propuestos; y, finalmente, en la sección 3.7, se describe las principales guías de diseño para la integración de SOD-M en el marco de la arquitectura de MIDAS.

3.1 Características Principales y Objetivos de SOD-M

El interés por el desarrollo de sitios o aplicaciones Web por parte de muchas organizaciones ha ido creciendo desde la aparición de la Web hasta nuestros días, hasta el punto de convertirse, a veces, en una necesidad para el progreso de las organizaciones en un entorno competitivo. Por ello, a lo largo de las últimas décadas, han surgido diversas propuestas metodológicas para abordar la construcción de sistemas o aplicaciones Web de un modo sistemático y riguroso, como, por ejemplo, las metodologías que se han analizado en la sección 2.2 del capítulo anterior.

Pero en los últimos años, la vertiginosa evolución de la Web y la aparición de nuevas facilidades y mejores tecnologías, ha planteado la necesidad de una

evolución por parte de las propuestas metodológicas para el desarrollo de aplicaciones para este medio. La computación orientada a servicios (SOC), definida como un nuevo paradigma que utiliza *servicios* como elementos básicos para la construcción de aplicaciones, es hoy en día, una de las mayores líneas de investigación en el área de desarrollo Web (Bichler y Lin, 2006). La importancia de este nuevo paradigma radica, entre otras cosas, en su capacidad para dar soporte a la manera en que las grandes empresas están conduciendo actualmente sus negocios, basados en redes de negocio en las que cada uno de los participantes provee a los demás servicios especializados (Curbera *et al.*, 2003). No obstante, y a pesar del impacto que los *servicios* han producido en desarrollo de software y en la manera en que los sistemas pueden ser construidos (Jacobs, 2005), en la actualidad, no existen metodologías que utilicen un *enfoque orientado a servicios* para el desarrollo de aplicaciones Web.

La propuesta que se presenta en esta Tesis Doctoral, llamada SOD-M (*Service-Oriented Development Method*), es un método para el desarrollo orientado a servicios de SIW, que reúne las siguientes **características principales**:

- Utiliza un enfoque orientado a servicios para el modelado y desarrollo completo de SIW: por lo que define guías para el modelado y desarrollo de aplicaciones Web en base a servicios, es decir, funcionalidades simples o complejas que son ofrecidas por una entidad y que permite satisfacer las necesidades de otra entidad que las solicita.
- Es una aproximación basada en MDA: por lo que propone un conjunto de modelos, que se extiende desde el nivel más alto de abstracción de la arquitectura MDA hasta los niveles PIM Y PSM de la misma. El método define, además, transformaciones entre los modelos, las cuales permiten pasar de manera sistemática de un nivel de abstracción a otro, facilitando así la alineación de los procesos de negocio de alto nivel con las tecnologías disponibles actualmente para el paradigma SOC.
- Utiliza UML como lenguaje de modelado del sistema y define un perfil UML específico para el desarrollo orientado a servicios de SIW: dicho perfil, llamado Perfil UML SOD-M, incluye todos los elementos de modelado necesarios a nivel PIM y PSM para el desarrollo de sistemas desde una perspectiva orientada a servicios. Cabe remarcar que, a nivel CIM, SOD-M propone la utilización del método de modelado de negocio de *e³value*, adoptando su notación, y la realización del modelado de procesos de negocio a través de un diagrama de actividad UML.

Como se ha dicho anteriormente (en el capítulo 1), SOD-M se integra en la arquitectura de modelos de MIDAS, una arquitectura basada en MDA para el desarrollo de SIW. Dentro de la arquitectura de modelos de MIDAS, SOD-M se centra en el desarrollo del aspecto del comportamiento de los SIW, puesto que se ocupa del modelado de las funcionalidades del mismo, y define guías para construir tales modelos a partir del modelado de negocio de alto nivel. Sin embargo, MIDAS contempla otros aspectos para el desarrollo de SIW (contenido e hipertexto), por lo que SOD-M define, además, guías para la integración de los modelos del aspecto de comportamiento con los demás modelos propuestos por MIDAS. Dicha integración se realiza mediante la definición de transformaciones horizontales entre los modelos del comportamiento y los modelos del aspecto del hipertexto. Esta integración se describe detalladamente en la sección 3.7 del presente capítulo.

Como **principales objetivos** de SOD-M se pueden resumir en los siguientes puntos:

- Ofrecer a los analistas de software un proceso de modelado centrado en la identificación de los servicios que serán ofrecidos a los usuarios o consumidores del SIW, partiendo para ello del modelado de negocio de la organización, e identificando los miembros del negocio y la manera en que éstos colaboran en la realización de cada uno de los servicios ofrecidos.
- Proveer a los diseñadores y desarrolladores de software un conjunto de modelos y transformaciones entre ellos, de manera que partiendo del modelado del negocio de alto nivel, les permita obtener modelos específicos para las tecnologías basadas en servicios Web, facilitando de esta manera la alineación entre los procesos de negocios de la organización y las actuales tecnologías de la información.

Antes de entrar de manera detallada en los modelos y el proceso propuesto por SOD-M, en la siguiente sección se presentan los fundamentos en los que se apoya el método propuesto, describiendo los nuevos conceptos de modelado en los que se basa y las relaciones entre tales conceptos.

3.2 Fundamentos de SOD-M

SOD-M propone un nuevo enfoque orientado a servicios para el desarrollo de SIW. Para ello, el método se apoya sobre un conjunto de conceptos que son necesarios para el modelado de un SIW desde una aproximación orientada a servicios; algunos de estos conceptos son ya conocidos mientras que otros son nuevos y se proponen como parte de este trabajo de investigación. La Figura 3.1

muestra, a través de un modelo de clases UML, los distintos conceptos en los que se apoya SOD-M así como las relaciones entre ellos.

Tal como puede verse en la Figura 3.1, SOD-M especifica dos perspectivas o puntos de vistas, desde los cuales es posible definir los conceptos propuestos:

- *Perspectiva del Negocio:* que se centra en las características y particularidades del negocio en el que se desenvolverá el sistema de información que se desea construir.
- *Perspectiva del Sistema:* que se centra en las funcionalidades y procesos necesarios para el desarrollo del comportamiento del sistema de información que se va a construir.

Aquellos conceptos de la Figura 3.1 que se corresponden con la perspectiva del negocio, describen elementos propios del negocio; los que se corresponden con la perspectiva del sistema, se utilizan para describir las funcionalidades y el procesamiento del sistema; y, aquellos que se corresponden con ambas perspectivas, son conceptos que pueden ser analizados desde ambos puntos de vista y que, por tanto, como se verá más adelante, serán los que permitan establecer una alineación o correspondencia entre los modelos de negocio de alto nivel y los modelos del SIW que se desee construir.

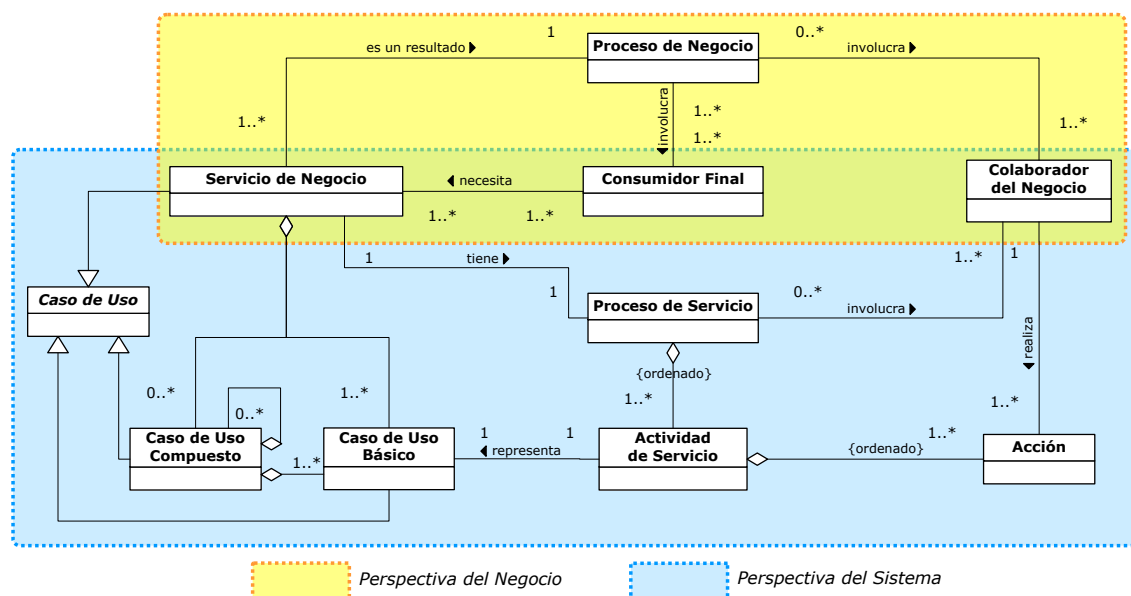


Figura 3.1. Conceptos de SOD-M

Con el fin de mejorar la comprensión en la definición de cada uno de los conceptos de SOD-M, éstos se ilustrarán con un ejemplo, para lo que se utilizará el siguiente escenario de negocio:

La agencia de viajes "Sin Rumbo" proporciona a sus clientes servicios de venta de billetes de avión y reserva de hoteles. Para ello, la agencia trabaja en colaboración con las distintas compañías aéreas, a las que le vende sus pasajes a cambio de una comisión, y con distintos hoteles con los que trabaja de igual manera. Por razones de seguridad, la agencia gestiona el cobro de los pasajes a través de un banco que le proporciona ese servicio. Además, la agencia colabora también con una agencia de correo que se encarga de la distribución de los billetes y los talones de reserva de hoteles.

A continuación se definen cada uno de los conceptos de SOD-M mostrados en la Figura 3.1:

Proceso de Negocio: un proceso de negocio se define como "un conjunto de tareas lógicamente relacionadas que se llevan a cabo para conseguir un resultado de negocio definido" (Pressman, 2005). Un proceso de negocio involucra gente, equipamientos, recursos materiales y procedimientos, los cuales se combinan para producir un resultado especificado. Los procesos de negocio a menudo sobrepasan los límites de una organización, por lo que es necesario que diferentes entidades o grupos organizacionales colaboren en la realización de las tareas que forman parte del proceso (Pressman, 2005).

Relaciones: cada proceso de negocio *involucra* a uno o varios consumidores finales o clientes, que son quienes reciben el resultado del negocio, y a uno o varios miembros o colaboradores del negocio que son quienes colaboran para llevar a cabo un proceso de negocio.

Ejemplo: en el escenario descrito anteriormente, ejemplos de procesos de negocio pueden ser: la venta de alguno de los productos de la agencia, el diseño de un nuevo paquete vacacional o el proceso de aceptación de nuevas compañías aéreas u hoteles.

Consumidor Final: un consumidor final se define aquí como "una entidad que necesita y consume los servicios de un negocio". Un consumidor final puede verse también como un cliente del negocio, y puede ser una persona, una organización, o bien un sistema hardware que consume servicios del negocio. Desde la perspectiva del negocio, un consumidor final es aquel que paga (ya sea con dinero, u otro objeto de valor) por obtener un servicio que el mismo consume, es decir, que no revende. Desde la perspectiva del sistema, un consumidor final puede verse como un usuario del sistema.

Relaciones: un consumidor final es aquel que *necesita* los servicios del negocio, y un servicio de negocio se ofrece porque hay un consumidor final que *tiene esa necesidad* y, en efecto, está dispuesto a pagar por el.

Ejemplo: en el escenario descrito anteriormente, un consumidor final es el “cliente”, es decir, quien va a la agencia porque necesita comprar un billete de avión o reservar un hotel.

Servicio de Negocio: un servicio de negocio se define como “un servicio que forma parte de un negocio y que se ofrece para satisfacer una necesidad de un consumidor final”. Desde la perspectiva del negocio, un servicio de negocio describe un resultado de un proceso de negocio. Desde la perspectiva de un sistema de información, un servicio de negocio describe una funcionalidad que es ofrecida por el sistema, para satisfacer una necesidad o deseo específico de un consumidor final o usuario del mismo.

Relaciones: un servicio de negocio se ofrece para *satisfacer la necesidad* de un consumidor final y es *un resultado* de un proceso de negocio de una organización. Además, todo servicio de negocio *tiene* asociado un proceso que es necesario llevar a cabo en el sistema para poder ofrecer el servicio.

Ejemplo: en el ejemplo de la agencia, los servicios de “venta de billetes de avión” y “reserva de hoteles” son ejemplos de servicios de negocio, y en ese caso, satisfacen las necesidades por parte del cliente de comprar un billete de avión y reservar un hotel, respectivamente.

Colaborador del Negocio: un colaborador del negocio se define como “una entidad que colabora en los procesos de negocio de una organización”. Desde la perspectiva del negocio, un colaborador del negocio es una entidad u organización que forma parte del negocio, es decir, que realiza ciertas tareas que son necesarias para el correcto funcionamiento del mismo. Desde la perspectiva del sistema, un colaborador del negocio es una entidad que realiza algunas de las acciones que son necesarias para llevar a cabo los servicios de negocio, y que puede ser interno o externo al sistema que se está modelando.

Relaciones: un proceso de servicio *involucra* a diferentes colaboradores del negocio, los cuales *realizan* una o varias de las diferentes acciones necesarias para la realización del servicio.

Ejemplo: en el caso de la agencia, los colaboradores del negocio son: las compañías de vuelo, los hoteles, la agencia que gestiona el cobro de los servicios, la agencia de correo y la misma agencia de viajes. Observe que desde un punto de vista del sistema, y centrándonos, por ejemplo, en el caso del servicio de negocio “venta de billetes de avión”, se pueden identificar los siguientes colaboradores del negocio: “la agencia de viajes” en sí, que realiza las acciones propias del sistema, tales como el registro de clientes; “las compañías de vuelo” que ofrecen los vuelos para la venta; “la agencia que gestiona el cobro” y que se encarga del cobro de los pasajes; y “la agencia de correo” que se encarga del envío de los billetes. En este caso, los tres últimos colaboradores del negocio son externos.

Caso de Uso Básico: Un caso de uso se define en UML como “un conjunto de acciones que realiza el sistema y que tiene un resultado observable y de interés para un actor particular del mismo” (OMG, 2003c). En SOD-M, un caso de uso básico es un caso de uso que representa un comportamiento elemental del sistema, y describe un conjunto de acciones que realiza el sistema para llevar a cabo parte de un servicio de negocio.

Relaciones: un caso de uso básico es un tipo de caso de uso y representa una parte de la funcionalidad de un servicio de negocio. Un servicio de negocio es una agregación de casos de uso (básicos o compuestos) que describen un conjunto de acciones que son requeridas por el sistema para llevar a cabo dicho servicio.

Ejemplo: en el caso de la agencia, se pueden identificar, por ejemplo, los siguientes casos de uso básicos: “buscar vuelos”, “registrar clientes”, “pagar vuelo”, “solicitar envío de billetes”, etc.; todos ellos son funcionalidades que realiza el sistema para llevar a cabo el servicio de negocio “venta de billetes de avión”.

Caso de Uso Compuesto: un caso de uso compuesto es también un caso de uso. En SOD-M, un caso de uso compuesto es un caso de uso que se puede descomponer en distintos casos de uso, los cuales pueden ser a su vez básicos o compuestos. Un caso de uso compuesto describe también, como el caso de uso básico, un conjunto de acciones que realiza el sistema para llevar a cabo parte de un servicio de negocio.

Relaciones: un caso de uso compuesto es un tipo de caso de uso y representa una parte de la funcionalidad de un servicio de negocio. Un caso de uso

compuesto *se puede descomponer* en uno o varios casos de uso básicos, y en ninguno o varios casos de uso compuestos.

Ejemplo: Un ejemplo de caso de uso compuesto puede ser un caso de uso llamado “reservar vuelo”, que incluya los casos de uso básicos “buscar vuelo” y “registrar cliente”.

Proceso de servicio: un proceso de servicio se define como “el conjunto de actividades lógicamente relacionadas que es necesario llevar a cabo en el sistema para la realización de un servicio de negocio”. Así, un proceso de servicio representa el flujo de trabajo necesario para realizar un servicio de negocio.

Relaciones: todo servicio de negocio *tiene asociado* un proceso de servicio. Un proceso de servicio *es un conjunto* de actividades, que se realizan con un cierto orden; tales actividades son necesarias para la realización de un servicio de negocio y se denominan actividades de servicio.

Ejemplo: en el caso de la agencia, un ejemplo de proceso de servicio, es el proceso asociado al servicio de negocio “venta de billetes de avión”, en el se identificará, por ejemplo, que el usuario debe realizar en primer lugar la “búsqueda de vuelos”, después realizar el “registro del cliente”, luego de eso el “pago del vuelo”, y finalmente introducir los datos para el “envío de billetes”.

Actividad de Servicio: una actividad se define en UML como “la especificación de un comportamiento que produce finalmente la ejecución de un conjunto de acciones individuales, cada una de las cuales puede producir un cambio en el estado del sistema o la comunicación de mensajes” (OMG, 2003c). En SOD-M, se llama actividades de servicio a aquellas actividades que forman parte del flujo de trabajo necesario para llevar a cabo un servicio de negocio.

Relaciones: un proceso de servicio *es una agregación* de actividades de servicios que se realizan con un cierto orden. Una actividad de servicio *representa*, en un proceso de servicio, el comportamiento que se define en un caso de uso básico.

Ejemplo: en el caso del servicio de negocio “venta de billetes de avión”, las actividades del servicio son: “búsqueda de vuelos”, “registro del cliente”, “pago del vuelo” y “envío de billetes”. El comportamiento expresado en dichas

actividades coincide con el comportamiento expresado en los casos de uso básico que se han identificado para el mismo servicio de negocio.

Acción: de acuerdo con la definición de UML, una acción es “una unidad de comportamiento fundamental que forma parte de una actividad, y que representa alguna transformación o procesamiento en el sistema que se está modelando” (OMG, 2003c). En SOD-M, una acción representa una unidad de comportamiento fundamental que forma parte de una actividad de servicio.

Relaciones: una actividad de servicio *se compone* de una o varias acciones que se realizan en un cierto orden y, por tanto, una acción *forma parte* de una actividad de servicio. Cada acción *es realizada* por un colaborador del negocio.

Ejemplo: en el caso que se analiza, para la actividad de servicio “pago del vuelo” podrían identificarse, por ejemplo, las acciones de “verificación de la tarjeta de crédito” y “registro del pago”, que se realizarían de una manera secuencial.

Todos los conceptos que se han definido en esta sección, tienen una representación en algunos de los diferentes modelos propuestos por SOD-M. En la siguiente sección se analiza la relación de SOD-M con el marco de MDA, y se describe cómo los distintos conceptos propuestos por SOD-M se distribuyen en los distintos niveles de modelado propuestos por MDA.

3.3 SOD-M en el Marco de MDA

La propuesta MDA del OMG define un marco de trabajo para el desarrollo de software basada en una idea principal: “separar la especificación de las funcionalidades del sistema, de la especificación de la implementación de tales funcionalidades sobre una plataforma tecnológica concreta” (OMG, 2001).

Para ello, MDA utiliza el concepto de “punto de vista” (*viewpoint*) sobre un sistema, utilizado como una técnica de abstracción para centrarse en un asunto o aspecto particular del sistema y propone, en concreto, tres puntos de vistas:

- *Independiente de Computación (Computation Independent Viewpoint):* centrado en el entorno del sistema y los requisitos del mismo, dónde los detalles de la estructura y el procesamiento del sistema son todavía desconocidos o indeterminados.

- *Independiente de Plataforma (Platform Independent Viewpoint)*: centrado en la funcionalidad del sistema, pero ocultando los detalles de una plataforma en particular, mostrando así aquella parte del sistema que no cambia de una plataforma a otra.
- *Específico de Plataforma (Platform Specific Viewpoint)*: que combina el punto de vista independiente de plataforma con los aspectos propios de la plataforma en la que se implementará el sistema.

MDA propone así, la representación del sistema desde los tres puntos de vistas definidos, utilizando para ello: modelos independientes de computación (CIM), modelos independientes de plataforma (PIM) y modelos específicos de plataforma (PSM).

Para llevar a cabo un desarrollo orientado a servicios de SIW, SOD-M se basa en las dos perspectivas definidas en la sección 3.2: *Perspectiva del Negocio* y *Perspectiva del Sistema*, y define modelos para la representación de un sistema desde ambas perspectivas. Así, aquellos conceptos que se corresponden con la perspectiva del negocio, es decir, que describen elementos propios del negocio, se representarán en los modelos de negocio propuestos por SOD-M; aquellos que se corresponden con la perspectiva del sistema, se representarán en los modelos propuestos para el modelado del sistema; y, aquellos que se corresponden con ambas perspectivas, tendrán una representación tanto en los modelos de negocio, como en los modelos del sistema propuestos por SOD-M.

La Figura 3.2 muestra la correspondencia entre las perspectivas propuestas por SOD-M para el desarrollo orientado a servicios de SIW, y los puntos de vistas propuestos por MDA. Así, tal como puede verse en la Figura 3.2, la perspectiva de negocio se corresponde con los modelos de nivel CIM de la arquitectura de MDA y la perspectiva del sistema se corresponde con los modelos de nivel PIM y PSM de dicha arquitectura.

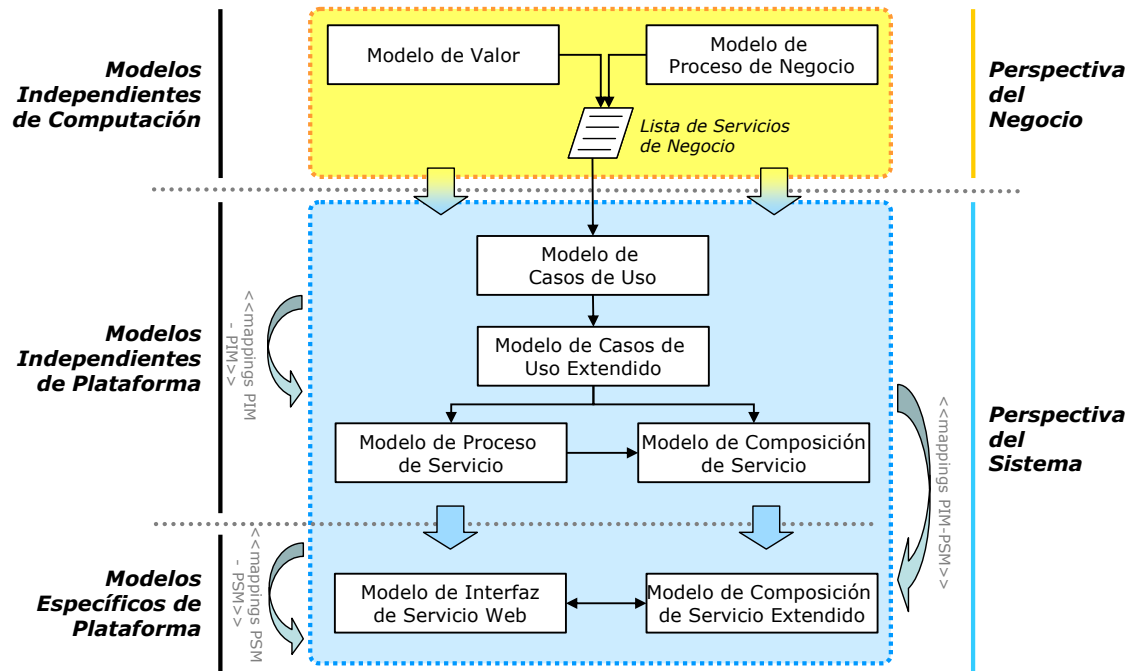


Figura 3.2. Modelos de SOD-M en el Marco de MDA

SOD-M propone, en concreto, llevar a cabo las siguientes actividades de modelado:

- **Modelado del Negocio:** se centra en el análisis de las características del negocio en el que se desenvolverá el sistema que se va a construir. Para el modelado de negocio a nivel CIM, SOD-M propone la realización de dos modelos: *modelo de valor* (Gordijn y Akkermans, 2003) y *modelo de proceso de negocio*. Además, se propone la especificación de una lista de servicios de negocio, que se utiliza como punto de partida para el modelado del sistema. Los modelos de valor y de proceso de negocio definen guías para la identificación y representación de los conceptos, presentados en la Figura 3.1, que coinciden con la perspectiva del negocio: proceso de negocio, consumidor final, servicio de negocio y colaborador del negocio.
- **Modelado del sistema:** se centra en el análisis de las funcionalidades y procesos necesarios para el desarrollo del comportamiento del sistema que se va a construir y, como se ha dicho antes, se corresponde con los modelos de nivel PIM y PSM de la arquitectura de MDA. Para el modelado del comportamiento del sistema a nivel PIM, SOD-M propone la realización de los siguientes modelos: *modelo de casos de uso*, *modelo de casos de uso extendido*, *modelo de proceso de servicio* y *modelo de composición de servicio*. Para el modelado del comportamiento del sistema a nivel PSM,

SOD-M propone la realización de dos modelos: *modelo de interfaz de servicio Web* y *modelo de composición de servicio extendido*. Tanto los modelos de nivel PIM como los modelos de nivel PSM definen guías para la identificación y representación de los conceptos, presentados en la Figura 3.1, que coinciden con la perspectiva del sistema: consumidor final, colaborador del negocio, servicio de negocio, caso de uso básico y compuesto, procesos de servicio, actividad de servicio y acción.

Todos los modelos propuestos por SOD-M, tanto para el modelado del negocio a nivel CIM como para el modelado del sistema a nivel PIM y PSM, se describen en detalle en la siguiente sección.

3.4 Modelos de SOD-M

En esta sección se presentan en detalle todos los modelos propuestos por el método, organizados por niveles de abstracción de la arquitectura de MDA. Así, en el apartado 3.4.1 se presentan los modelos de nivel CIM, en el apartado 3.4.2 se presentan los modelos de nivel PIM, y los modelos de nivel PSM en el apartado 3.4.3. Cada uno de los modelos propuestos se describirá indicando qué representa y la notación que utiliza, así como los elementos de modelado que propone, los cuales se presentarán a través del metamodelo de cada modelo.

El proceso y las actividades necesarias para la construcción de los modelos que aquí se presentan, se describirán detalladamente en la sección 3.6 de este capítulo. Además, este proceso se ilustrará con un ejemplo en la sección 4.3, donde se describe el proceso de desarrollo completo para uno de los SIW que se ha realizado como caso de estudio de esta Tesis Doctoral.

3.4.1 Modelos Independientes de Computación

Los modelos CIM se utilizan para describir la situación o el entorno en el que se utilizará el sistema, abstrayéndose totalmente de cómo el sistema será implementado. Tales modelos son especialmente útiles, no sólo para comprender el problema, sino también para establecer un vocabulario compartido que podrá ser utilizado en el resto de los modelos del sistema. Uno de los roles más importantes de los modelos CIM en el desarrollo de sistemas, es la posibilidad de establecer un puente que sirva de enlace entre los analistas o expertos en el negocio y sus requisitos, y los diseñadores o desarrolladores encargados del diseño y la implementación de los sistemas de información contruidos para satisfacer tales requisitos (OMG, 2001).

Como se ha dicho anteriormente, el método que se presenta en esta Tesis Doctoral se centra en el aspecto del comportamiento de los SIW. Por ello, SOD-M propone la realización de dos modelos a nivel CIM, cuyo objetivo es describir el negocio en el que se desenvuelve el sistema e identificar los servicios que deben ser ofrecidos a los usuarios o consumidores del mismo. Tales modelos son el *modelo de valor* y el *modelo de proceso de negocio*, los cuales se describen en los siguientes apartados según la estructura antes mencionada.

3.4.1.1 Modelo de Valor

Este modelo permite la representación gráfica de una idea de negocio. Por sus características, el modelo de valor es especialmente útil para la comprensión del negocio, proporcionando herramientas para la identificación de los actores del negocio y las relaciones entre ellos, así como para la identificación de las necesidades requeridas por los consumidores finales del negocio, y la manera en que dichas necesidades se satisfacen entre los actores que colaboran en el mismo.

El modelo de valor se obtiene como resultado de la aplicación del método de modelado de negocio *e³value* (Gordijn y Akkermans, 2003). El método *e³value* define un conjunto de conceptos, procedentes de las ciencias de la administración de empresas y del marketing, y propone una notación propia y específica para la representación de los mismos en un modelo de valor.

Elementos de Modelado

El modelo de valor permite la identificación de aquellos objetos de valor (bienes, dinero, o servicios) que son creados y proporcionados por cada uno de los miembros que participan en un negocio (*colaboradores del negocio*), así como de los miembros del negocio que están interesados en dichos objetos de valor. En el modelo de valor se muestra, además, el intercambio de objetos de valor existente entre cada uno de los miembros del negocio.

La Figura 3.3 presenta el metamodelo del modelo de valor propuesto por el método *e³value*, donde se muestran los principales conceptos de un modelo de valor y las relaciones entre ellos. Tales conceptos se describen detalladamente en la Tabla 3.1.

El método *e³value* define también, guías para identificar caminos de dependencias (*dependency paths*). Los caminos de dependencias muestran los intercambios de objetos de valor que tiene lugar ante un determinado estímulo o necesidad de un *consumidor final* perteneciente al negocio. Tales estímulos o

necesidades de un consumidor final se utilizan para la identificación de los *servicios de negocio* que el sistema deberá proporcionar a los usuarios.

Un camino de dependencia tiene una dirección, y consta de nodos de dependencia y conexiones. Los elementos utilizados para la representación de caminos de dependencias se describen también en la Tabla 3.1.

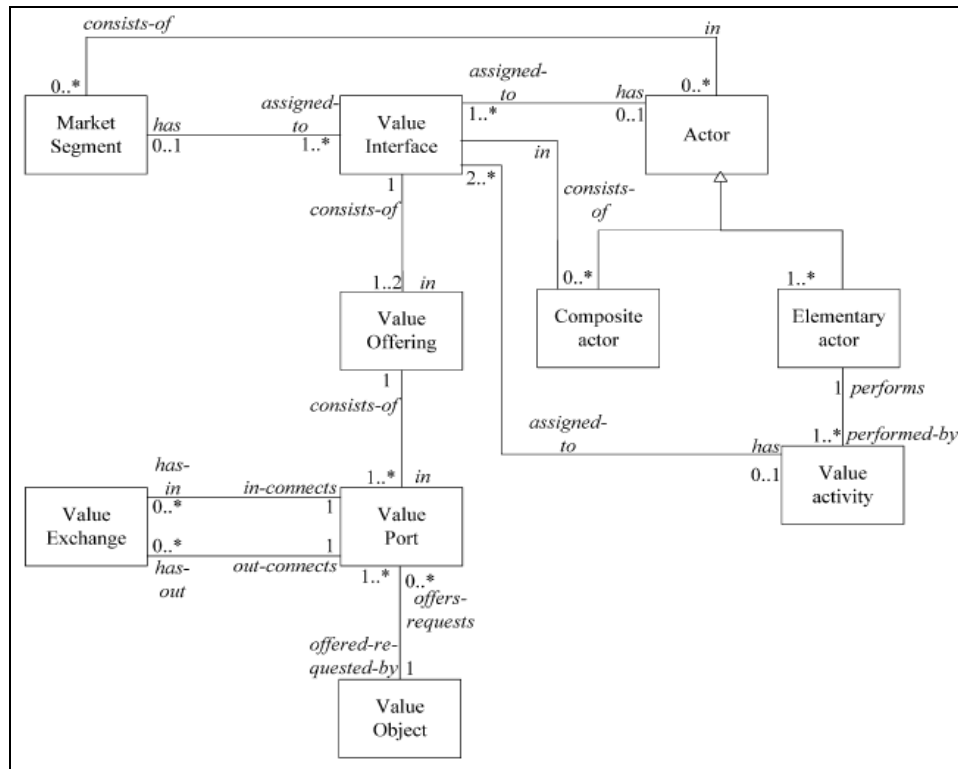
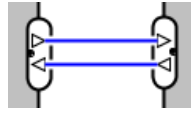
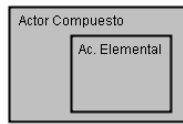
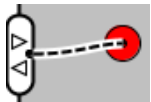


Figura 3.3. Metamodelo del modelo de valor de e³value (Gordijn y Akkermans, 2003)

Tabla 3.1. Conceptos, semántica y notación de los elementos del Modelo de Valor.

Concepto	Semántica	Notación
Actor	Es una entidad que es percibida por el entorno como económicamente independiente. Un actor es un miembro del negocio y forma parte de este para generar beneficios o incrementar su utilidad. En un modelo de negocio sostenible cada actor debe ser capaz de generar beneficios. Un consumidor final del negocio es también un actor.	
Objeto de Valor (Value Object)	Es un objeto que tiene valor para, al menos, uno de los actores del negocio. Los actores intercambian objetos de valor que pueden ser servicios, productos, dinero, etc.	
Puerto de Valor (Value Port)	Representa un puerto que utiliza un actor para mostrar a su entorno que quiere proveer o solicitar determinados objetos de valor. Así, un puerto de valor se utiliza para interconectar actores que están dispuestos a intercambiar objetos de valor. Un puerto de valor tiene una dirección que indica si el objeto se recibe o se ofrece.	

Concepto	Semántica	Notación
Ofrecimiento de Valor (<i>Value Offering</i>)	Representa lo que un actor ofrece a, o requiere de, su entorno. Este concepto está estrechamente relacionado con el concepto de interfaz de valor. Un ofrecimiento de valor es un conjunto de puertos de valor de igual dirección (saliente o entrante). El intercambio de objetos de valor entre dos actores vía puertos de valor, es siempre atómico; es decir, todos los puertos intercambian un objeto, o ninguno lo hace.	Sin Representación
Interfaz de Valor (<i>Value Interface</i>)	Una interfaz de valor consta de puertos de valor a través de los cuales se está ofreciendo o requiriendo objetos de valor. Un actor puede tener una o más interfaces de valor. A través de las interfaces de valor se muestra el mecanismo de reciprocidad económica; es decir, en un negocio con actores actuando de manera racional, ningún actor ofrecerá nada sin recibir algo a cambio. A través de una interfaz de valor se representa que un actor está dispuesto a ofrecer algo de valor para su entorno, pero recibiendo algo a cambio que es de valor para el mismo.	
Intercambio de Valor (<i>Value Exchange</i>)	El intercambio de valor se utiliza para conectar dos puertos de valor entre sí y representa que dos actores, con puertos de valor conectados, están dispuestos a intercambiar objetos de valor entre ellos.	
Segmento de Mercado (<i>Market Segment</i>)	Representa un conjunto de actores que atribuyen el mismo valor económico a los objetos que intercambian con el resto de actores.	
Actor Compuesto y Elemental (<i>Composite and Elementary actors</i>)	Ambos son una especialización de un actor. Un actor compuesto agrupa interfaces de valor de otros actores y tiene sus propias interfaces de valor con el entorno. De esta manera, es posible modelar una reunión o asociación (<i>partnership</i>) de actores. Un actor elemental no tiene interfaces de valor y es el nivel más bajo de descomposición que se puede alcanzar desde la perspectiva de un actor.	
Actividad de Valor (<i>Value Activity</i>)	Es una actividad que realiza un actor y que se espera que sea rentable para el mismo. Una actividad de valor se representa gráficamente dentro del actor que realiza la actividad.	
Nodo de Dependencia (<i>Dependency Node</i>)	Un nodo de dependencia puede ser: a) un estímulo de inicio; b) un nodo final; c) una unión o bifurcación de tipo OR; o, d) una unión o bifurcación de tipo AND. Un estímulo de inicio representa una necesidad de un consumidor final perteneciente al negocio, lo que provoca un intercambio de objetos de valor. Un nodo final representa un límite del modelo.	a)  b)  c)  d) 
Conexión de Dependencia (<i>Dependency Connection</i>)	Una conexión de dependencia, que se representa por una línea punteada, conecta nodos de dependencia e interfaces de valor.	

3.4.1.2 Modelo de Proceso de Negocio

El modelo de proceso de negocio, se utiliza para entender y describir los procesos de negocio relacionados con el entorno en el que se utilizará el sistema que se desea construir. Como se ha definido previamente (en la sección 3.2), los procesos de negocio describen un conjunto de tareas que es necesario llevar a cabo para conseguir un resultado de negocio definido. Sin embargo, es necesario tener en cuenta que este modelo es un modelo de negocio de alto nivel, por lo que los procesos identificados aquí no tendrán, necesariamente, una implementación asociada en el sistema que se va a construir, sino que simplemente se llevan a cabo para comprender las características del entorno, e identificar los servicios de negocio que el SIW deberá ofrecer a los consumidores finales o usuarios del mismo.

SOD-M propone la representación del modelo de proceso de negocio a través de un diagrama de actividad UML (OMG, 2003c).

Elementos de Modelado

Este modelo permite la identificación de los *servicios de negocio*, que serán ofrecidos a los consumidores finales o usuarios del SIW que se desea construir. Cada actividad o conjunto de actividades de este modelo, puede dar lugar a la identificación de uno o varios *servicios de negocios*.

Los elementos de este modelo coinciden con los elementos de modelado utilizados tradicionalmente en un diagrama de actividad UML, tales como actividad, flujo de control, nodos de control, etc. (OMG, 2003c); por ello, en este caso se omite el metamodelo del mismo.

3.4.2 Modelos Independientes de Plataforma

Los modelos PIM se utilizan para el modelado de la funcionalidad y estructura del sistema abstrayéndose de los detalles tecnológicos de la plataforma en la que se implementará. De esta manera, es posible que tales modelos puedan ser igualmente adecuados para utilizarlos en una u otra plataforma de implementación de similares características.

Tal como se ha descrito en la sección 3.3 del presente capítulo, SOD-M propone a nivel PIM la realización de diversos modelos para el comportamiento de sistema. Dado que SOD-M define un método para el desarrollo orientado a servicios de SIW, tales modelos se centran en la identificación de los servicios que serán ofrecidos por el sistema, así como en la identificación de las funcionalidades y procesos necesarios para llevar a cabo tales servicios. Los modelos propuestos por

SOD-M a nivel PIM, y que se describen a continuación, son los siguientes: *modelo de casos de uso*, *modelo de casos de uso extendido*, *modelo de proceso de servicio* y *modelo de composición de servicio*.

3.4.2.1 Modelo de Casos de Uso

El modelo de casos de uso es un modelo de comportamiento que muestra un conjunto de casos de usos y sus relaciones con actores (OMG, 2003c). SOD-M propone la realización de un modelo de casos de uso para la representación de los servicios de negocio que se implementarán en el sistema y sus relaciones con los consumidores finales o usuarios del SIW que los requieren.

SOD-M propone representar este modelo a través de la técnica de diagrama de casos de uso de UML (OMG, 2003c).

Elementos de Modelado

En este modelo se identifican dos de los conceptos que se han presentado anteriormente en la sección 3.2: *consumidor final*, que se representa como un actor; y *servicio de negocio*, que se representa como un caso de uso. Cabe remarcar que, en este modelo, sólo se representan los servicios de negocio que serán ofrecidos a través del sistema que se está modelando, puesto que es posible que en el modelado de negocio se hayan identificado otros servicios de negocio pero que no formarán parte del SIW que se va a desarrollar.

La Figura 3.4 presenta el metamodelo de casos de uso, donde se muestran también los nuevos conceptos propuestos por SOD-M (sombreados), así como su relación con los elementos básicos del modelo de casos de uso. Tales elementos, junto con su semántica, notación, y restricciones se describen en la Tabla 3.2.

Como puede verse en la Figura 3.4 los elementos *UseCase* y *Actor* del metamodelo son clases abstractas. Esto es así, porque en este modelo sólo se permite la representación de *servicios de negocio*, un tipo especial de caso de uso, y de *consumidores finales*, un tipo especial de actor. Por esta razón, y para no complicar la representación del modelo, en este caso no se han definido estereotipos particulares para los nuevos conceptos propuestos, sino que solo se propone la representación de estos conceptos a través de un modelo de casos de uso tradicional de UML.

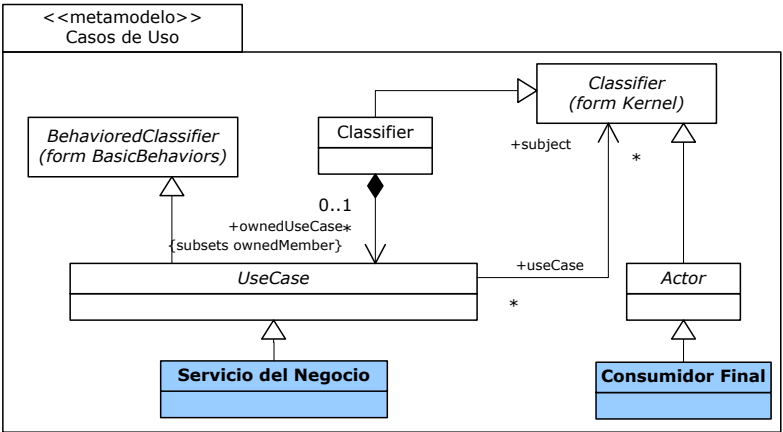


Figura 3.4. Metamodelo de Casos de Uso

Tabla 3.2. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Casos de Uso

Concepto	Semántica	Notación		Restricciones
		Meta-clase de UML	Estereotipo	
Servicio de Negocio	Representa una funcionalidad que es ofrecida por el sistema para satisfacer una necesidad o deseo específico de un consumidor final o usuario del mismo.	Caso de Uso	-	Solo puede estar relacionado con un consumidor final.
Consumidor Final	Representa a un usuario del sistema que requiere un servicio de negocio del sistema.	Actor	-	Solo puede estar relacionado con un servicio de negocio.

3.4.2.2 Modelo de Casos de Uso Extendido

El modelo de caso de uso extendido es también un modelo de comportamiento del sistema, y se utiliza para el modelado de las funcionalidades requeridas por el sistema para llevar a cabo los servicios de negocio.

Para la representación de este modelo se utiliza también la técnica de diagrama de casos de uso de UML (OMG, 2003c), pero en este caso se definen nuevos elementos de modelado que se representan en el modelo y que extienden los elementos para el modelado de casos de uso tradicional de UML.

Elementos de Modelado

En este modelo se representan los siguientes conceptos definidos en la sección 3.2: *consumidor final* que, al igual que en el modelo anterior, se representa como un actor; *caso de uso básico*, que se representan como un caso de uso; y *caso de uso compuesto*, que se representa también como un caso de uso, pero estereotipado para indicar que es un caso de uso que se puede descomponer.

Además, en este modelo se propone la utilización de relaciones de *include* y *extend* entre los distintos casos de uso identificados. La semántica de dichas relaciones en este modelo es la misma que en el modelo de casos de uso tradicional de UML: “Una relación *include* especifica la existencia de un flujo de eventos en el cual el caso de uso base incluye el comportamiento del otro” y “una relación *extend* especifica que el comportamiento de un caso de uso base puede ser opcionalmente extendido por el comportamiento de otro caso de uso” (OMG, 2003c).

La Figura 3.5 presenta el metamodelo de casos de uso extendido; en ella se destacan también los nuevos conceptos propuestos por SOD-M, así como sus relaciones con los elementos básicos del modelo de casos de uso. Tales elementos se describen en la Tabla 3.3 de la misma manera que en el modelo anterior, indicando su semántica, notación, y restricciones.

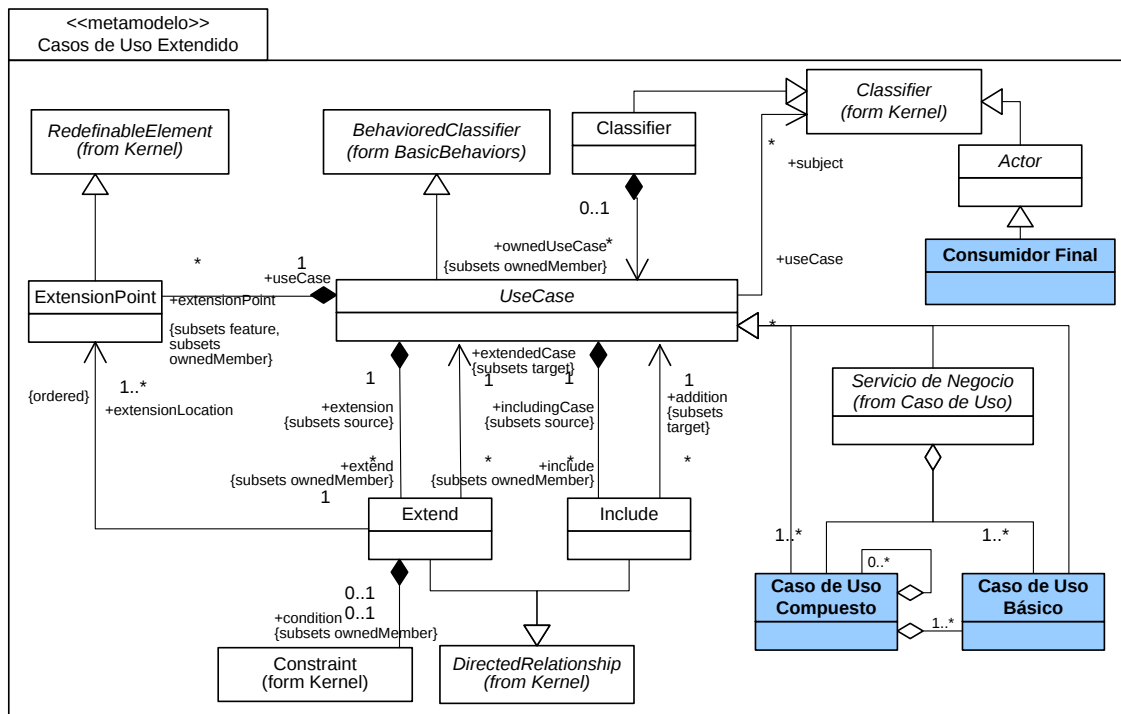


Figura 3.5. Metamodelo de Casos de Uso Extendido

Tabla 3.3. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Casos de Uso Extendido

Concepto	Semántica	Notación		Restricciones
		Meta-clase de UML	Estereotipo	
Caso de Uso Básico	Representa un conjunto de acciones que realiza el sistema para llevar a cabo parte de un servicio de negocio.	Caso de Uso	-	Puede estar relacionado con un consumidor final, o con uno o varios casos de uso.

Concepto	Semántica	Notación		Restricciones
		Meta-clase de UML	Estereotipo	
Caso de Uso Compuesto	Representa un conjunto de acciones que realiza el sistema para llevar a cabo parte de un servicio de negocio. Se puede descomponer en varios casos de uso, los cuales pueden ser a su vez básicos o compuestos.	Caso de Uso	<<Comp>>	Puede estar relacionado con un consumidor final, o con uno o varios casos de uso.

3.4.2.3 Modelo de Proceso de Servicio

Este modelo se utiliza para la representación de un proceso de servicio, por lo que muestra el conjunto de actividades lógicamente relacionadas que es necesario realizar en sistema para llevar a cabo un servicio de negocio. Las actividades de este modelo representan un comportamiento que forma parte del flujo de trabajo necesario para la realización de un servicio de negocio.

SOD-M propone representar este modelo a través la técnica de diagrama de actividad de UML (OMG, 2003c).

Elementos de Modelado

En este modelo se identifican dos de los conceptos que se han presentado anteriormente en la sección 3.2: el concepto de *proceso de servicio*, y el concepto de *actividad de servicio*, que se representa como una actividad. Hay que destacar que, en este caso, el concepto de *proceso de servicio* no es un elemento que se represente en el modelo, sino que el modelo en sí mismo representa ese concepto.

En la Figura 3.6 se presenta el metamodelo de proceso de servicio; en ella se destaca el elemento *actividad de servicio*, que es un tipo especial de actividad. Este nuevo elemento de modelado, junto con su semántica, notación, y restricciones se describe en la Tabla 3.4.

En este caso, como en el modelo de casos de uso presentado anteriormente, tampoco se define un estereotipo propio para la representación de las *actividades de servicio*, puesto que sólo se permite representar este tipo especial de actividad en el modelo.

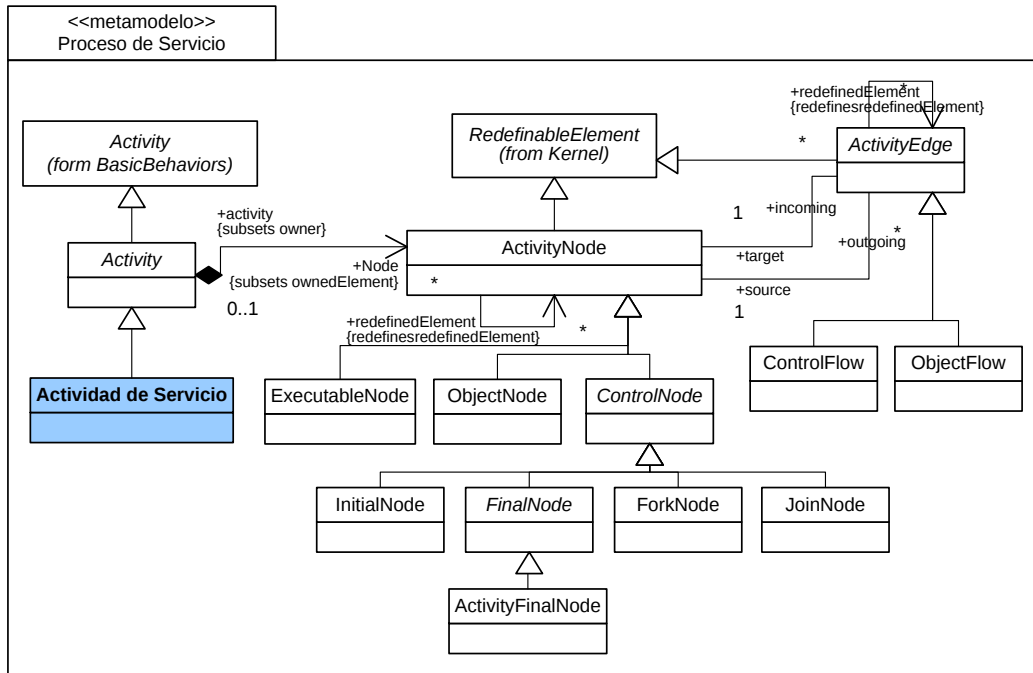


Figura 3.6. Metamodelo de Proceso de Servicio

Tabla 3.4. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Proceso de Servicio

Concepto	Semántica	Notación		Restricciones
		Meta-clase de UML	Estereotipo	
Actividad de Servicio	Representa un comportamiento que forma parte del flujo de trabajo necesario para llevar a cabo un servicio de negocio.	Actividad	-	-

3.4.2.4 Modelo de Composición de Servicio

Este modelo extiende la representación del proceso de servicio que se realiza en el modelo de proceso de servicio descrito anteriormente. El modelo de composición de servicio representa el flujo de trabajo necesario para llevar a cabo un servicio pero de una manera más detalla, es decir, identificando quiénes son diferentes colaboradores del negocio que participan en el proceso de realización del servicio, y qué acciones lleva a cabo cada uno de ellos. Así, el modelo permite la representación del proceso de composición de las distintas acciones que son necesarias para llevar a cabo un servicio del negocio. Como se ha definido en la sección 3.2, cada acción identificada en este modelo describe una unidad de comportamiento fundamental que forma parte de una actividad de servicio, y que representa alguna transformación o procesamiento en el sistema que se está modelando.

SOD-M propone representar este modelo a través la técnica de diagrama de actividad de UML (OMG, 2003c).

Elementos de Modelado

En este modelo se representa el flujo de trabajo necesario para llevar a cabo un servicio de negocio, y en el se identifican tres de los conceptos que se han presentado anteriormente en la sección 3.2: el *proceso de servicio*, que es el flujo de trabajo en sí mismo; las *acciones*, que se representan en el modelo como un nodo de actividad (*ActivityNode* en la Figura 3.7); y los *colaboradores del negocio*, que se representa como una partición en el diagrama de actividad (*ActivityPartition* en Figura 3.7). Las *acciones* que se identifican en este modelo se distribuyen entre los distintos colaboradores del negocio (particiones) que se representan en el mismo. Como ya se ha dicho, un *colaborador del negocio* puede ser interno o externo al sistema que se esta modelando. Cuando el colaborador del negocio es externo al sistema, la partición del diagrama de actividad que representa a dicho colaborador se estereotipa con la palabra <<external>>.

En la Figura 3.7 se presenta el metamodelo de composición de servicio, en la que se señalan los nuevos conceptos de modelado propuestos, así como sus relaciones con los elementos del modelo de actividad. La Tabla 3.5 describe tales conceptos, indicando su semántica, notación y restricciones.

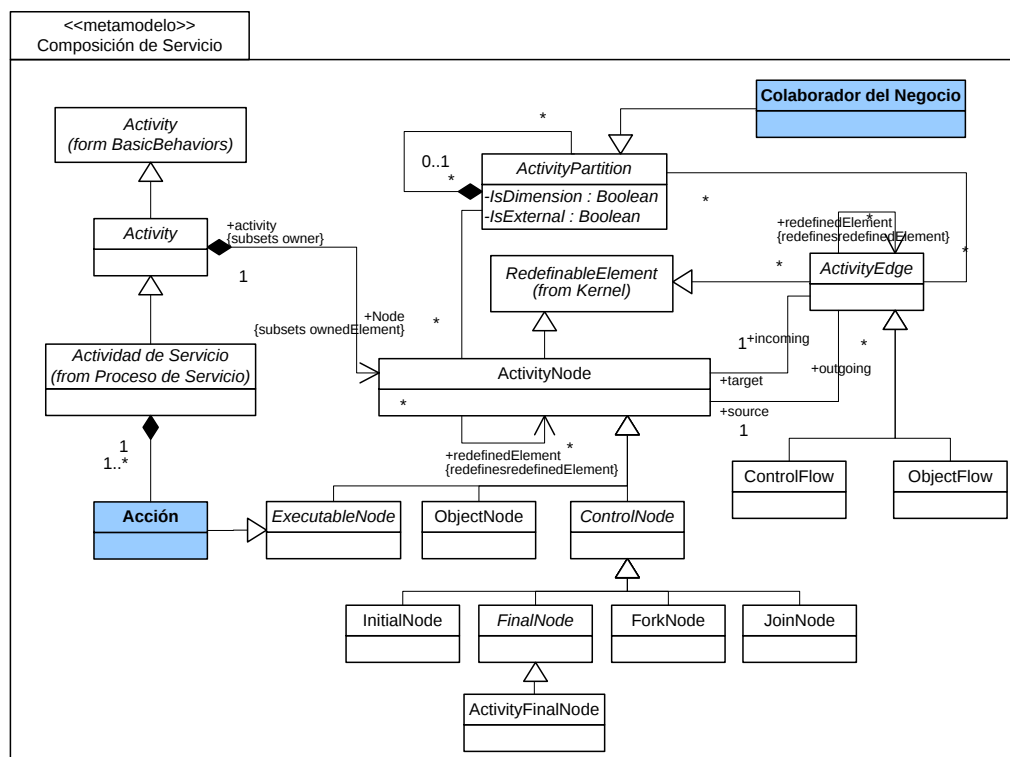


Figura 3.7. Metamodelo de Composición de Servicio

Tabla 3.5. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Composición de Servicio

Concepto	Semántica	Notación		Restricciones
		Meta-clase de UML	Estereotipo	
Acción	Representa una unidad de comportamiento fundamental que forma parte de una actividad de servicio, y que representa alguna transformación o procesamiento en el sistema que se está modelando.	Nodo de Actividad	-	Debe estar asociado a un colaborador del negocio.
Colaborador del Negocio	Representa una entidad que realiza algunas de las acciones que son necesarias para llevar a cabo los servicios de negocio, y que puede ser interno o externo al sistema que se está modelando.	Partición (Activity Partition)	<<external>> (en el caso de que represente un colaborador del negocio externo al sistema)	Debe estar asociado por lo menos a una acción.

3.4.3 Modelos Específicos de Plataforma

Los modelos PSM se utilizan para combinar las especificaciones contenidas en los modelos PIM, con los detalles de la plataforma elegida para la implementación del sistema.

Tales modelos pueden proporcionar más o menos detalles dependiendo de su propósito. Así, si un PSM va a ser utilizado para la implementación del sistema, este debe proporcionar toda la información necesaria tanto para construir el sistema y sus componentes, como para ponerlo en ejecución. Sin embargo, un PSM puede actuar también como un modelo que será utilizado para nuevos refinamientos a modelos PSM (OMG, 2001), dando así lugar a nuevos niveles de modelos PSM. Así, por ejemplo, en (Muller *et al.*, 2005) se habla de dos subniveles dentro del nivel PSM: *Platform Dependent PSM (PD-PSM)*, a nivel PSM superior, en el que se realiza el modelado del sistema teniendo en cuenta detalles específicos de una plataforma concreta; y *Technology Dependent PSM (TD-PSM)*, a nivel PSM inferior, en el que se realiza el modelado del sistema teniendo en cuenta detalles de una tecnología concreta con la que se implementa la plataforma.

SOD-M propone realizar dos modelos de nivel PSM, en los que se combinan algunos de los conceptos descritos en la sección 3.2 con detalles específicos de las plataformas basadas en servicios Web: el *modelo de interfaz de servicios Web* y el *modelo de composición de servicio extendido*.

Dichos modelos, que se describen en los siguientes apartados de esta sección, se corresponden con el nivel denominado PD-PSM en (Muller *et al.*, 2005), por lo que no incluyen detalles sobre una tecnología concreta para la implementación de plataformas basadas en servicios Web. Sin embargo, tales modelos podrán ser posteriormente refinados, dando lugar a nuevos modelos de nivel PSM dependientes de una tecnología basada en servicios Web concreta, como la plataforma .NET (Microsoft .NET) o Globus Toolkit 4 (Foster, 2005) basada en OGSA (*Open Grid Services Architecture*) (OSGA, 2003). Tanto este último nivel de detalle dentro de los modelos de nivel PSM como la generación de código a partir de ellos quedan fuera del alcance de esta Tesis Doctoral, y se incluyen como líneas de investigación con las que se continuará trabajando en el futuro.

3.4.3.1 Modelo de Interfaz de Servicio Web

Este modelo se utiliza para describir la interfaz de los servicios Web que serán utilizados en la ejecución de cada uno de los servicios que ofrece el sistema.

El modelo de interfaz de servicio Web, está basado en el estándar WSDL (*Web Service Description Language*) (W3C, 2004a). WSDL es el lenguaje propuesto por el W3C para la descripción de servicios Web y permite describir la interfaz del servicio en un formato XML. El modelo de interfaz de servicio Web propuesto permite obtener una representación gráfica de la interfaz de un servicio Web, a partir de la cual es posible la generación automática del correspondiente código WSDL.

SOD-M propone representar este modelo a través la técnica de diagrama de clases UML (OMG, 2003c), definiendo para ello un conjunto de nuevos elementos de modelado que se representan en el modelo y que extienden los elementos del modelo de clases UML.

Elementos de Modelado

Los elementos de modelado que se incluyen en este modelo se corresponden con los componentes definidos por el estándar WSDL para la representación de la interfaz de un servicio Web. Tales componentes, así como las relaciones entre ellos se muestran en la Figura 3.8 por medio de un diagrama de clases UML.

Como puede verse en la Figura 3.8, un documento WSDL contiene un componente raíz *Definition* que posee un nombre (*name*) y un atributo *TargetNamespace*, que declara el espacio de nombres al que pertenecerán todos los nombres de componentes definidos en el documento. Un componente *Definition* puede contener un componente *Types* y cero o más componentes *Message*,

Interface, *Binding*, *Service* e *Import*. Todos los componentes de WSDL pueden tener asociado un componente *Documentation*.

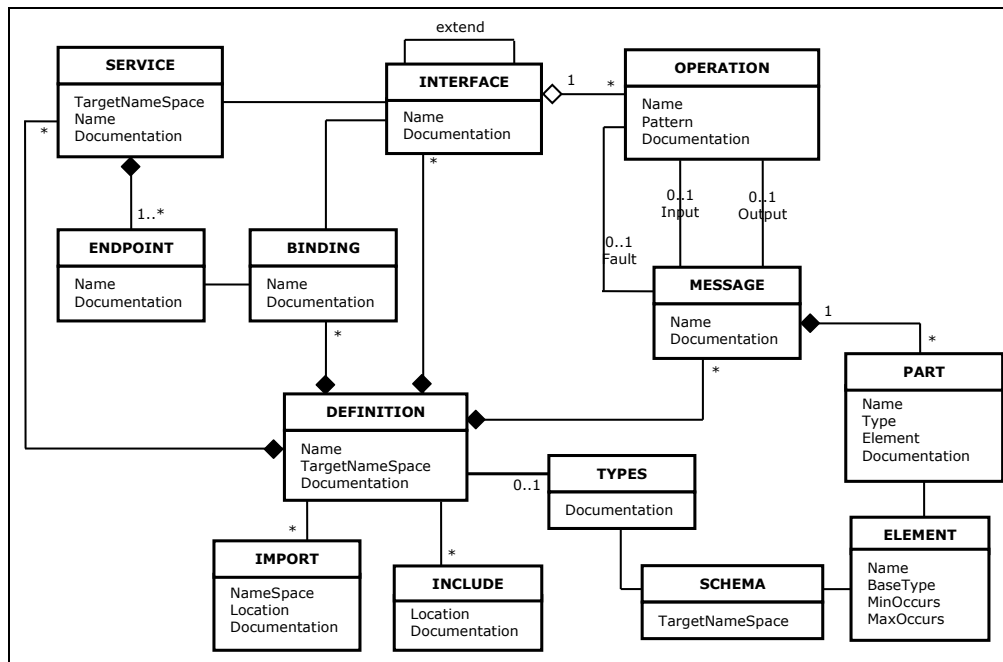


Figura 3.8. Metamodelo del estándar WSDL Versión 2.0 Core Language

Un componente *Types* se utiliza para la definición de los tipos de datos que serán necesarios para el envío de mensajes. Para esta definición, WSDL se basa en el estándar *XML Schema* (W3C, 2004b). Por ello contiene un componente *Schema* y asociado a este, componentes *Element* que describen cada uno de los tipos de datos, de acuerdo al estándar mencionado.

WSDL permite incluir documentos *XML Schemas* definido previamente y para ello utiliza un componente *Include*, mediante el cual se indica la localización del documento. De la misma manera, un componente *Import* se utiliza para reutilizar definiciones hechas en otros documentos WSDL, indicando el nombre y localización del documento que se desea importar.

El componente *Interface* se utiliza para describir el conjunto de mensajes que serán intercambiados entre el proveedor del servicio y el cliente. Los mensajes se agrupan en operaciones, *Operation*. Cada operación contiene un nombre y puede tener uno, dos o tres mensajes asociados, es decir, un *input message*, un *output message* o ambos, y opcionalmente un *fault message*. Un mensaje, *Message* tiene una definición similar a la de una función, contiene un nombre y parámetros, *Part*. El tipo asociado a un *Part* puede ser un tipo base XSD (int, float, string, etc.) o un tipo definido en la sección de tipos. En este último caso, el tipo de dato puede ser asociado a través de un atributo *type* o *element*, dependiendo del tipo de dato que

se desea asociar. Hasta aquí, el componente *Interface* describe completamente la interfaz del servicio de manera abstracta.

El componente *Binding* describe un formato de mensaje y protocolo de transmisión concretos (SOAP, HTTP o MIME) para el enlace con un componente *Interface*. El componente *Endpoint* define el punto concreto de acceso al servicio. Este componente contiene un nombre y combina un componente *Binding* con la dirección donde se accede a la implementación del servicio. Finalmente, el componente *Service* contiene un nombre, que será el nombre del servicio, y uno o varios puertos *Endpoint* asociados.

Como se ha dicho anteriormente, el modelo de interfaz de servicio Web propuesto por SOD-M, define un conjunto de elementos de modelado que se corresponden con los componentes definidos por el estándar WSDL. En la Figura 3.9 se presenta el metamodelo de interfaz de servicio Web. Dicha figura muestra sombreados los nuevos elementos de modelado propuestos por SOD-M para la representación de la interfaz de un servicio Web mediante la técnica de diagrama de clases, así como sus relaciones con los elementos del modelo de clases de UML.

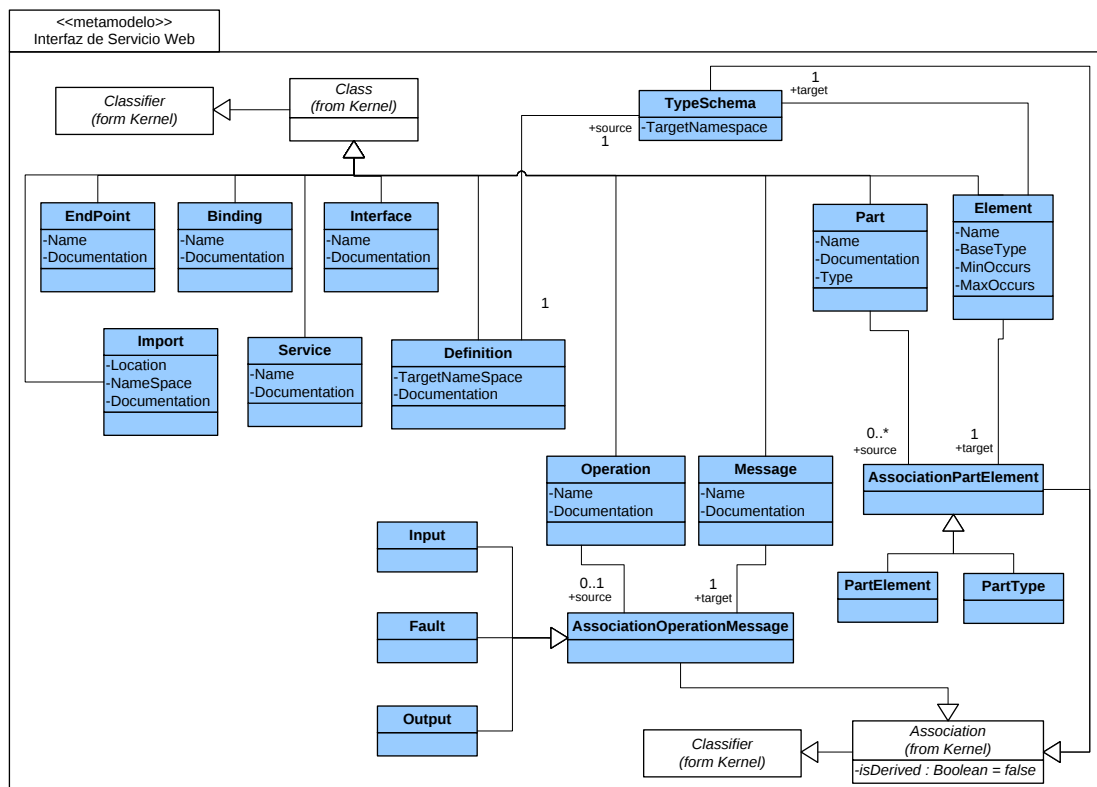


Figura 3.9. Metamodelo de Interfaz de Servicio Web

La Tabla 3.6 describe tales elementos, indicando su semántica, notación, restricciones y valores etiquetados. Los valores etiquetados se utilizan para

extender las propiedades de un estereotipo concreto. En este caso, y debido a que los elementos de este modelo deben representar los elementos del estándar WSDL y su semántica, ha sido necesario incluir valores etiquetados para algunos de los elementos de modelado propuestos, los cuales se utilizan para especificar ciertas propiedades de dichos elementos. Los valores etiquetados de cada nuevo elemento de modelado se muestran en la Figura 3.9 como atributos del mismo.

Tabla 3.6. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Interfaz de Servicios Web

Concepto	Semántica	Notación		Restricciones y Valores Etiquetados
		Meta-clase de UML	Estereotipo	
Definition	Representa el componente <i>Definition</i> del estándar WSDL.	Clase	<<DEFINITION>>	Valores Etiquetados: <i>TargetNameSpace</i> , es obligatorio y contiene una URI (Uniform Resource Identifier) que indica el espacio de nombres al que pertenecen todos los nombres de componentes definidos en el documento. <i>Documentation</i> , es opcional y contiene una descripción del componente.
Element	Representa un componente <i>Element</i> del estándar WSDL.	Clase	<<ELEMENT>>	Restricciones: esta relacionado mediante una composición estereotipada con <<Type_Schema>> a la clase <i>Definition</i> . Valores Etiquetados: <i>Name</i> , indica el nombre del elemento. <i>BaseType</i> , que indica el tipo base. <i>MinOccurs</i> y <i>MaxOccurs</i> , que indican el número mínimo y máximo de ocurrencias del elemento, respectivamente.
TypeSchema	Representa una relación entre un componente <i>Definition</i> y los tipos de datos definidos en componentes <i>Element</i> .	Asociación (tipo Composición)	<<Type_Schema>>	Restricciones: sólo se utiliza para relacionar una clase <i>Definition</i> con una clase <i>Element</i> . Valores Etiquetados: <i>TargetNameSpace</i> , que indica el espacio de nombre utilizado por el componente <i>Schema</i> del estándar WSDL.
Message	Representa un componente <i>Message</i> del metamodelo de WSDL.	Clase	<<MESSAGE>>	Restricciones: debe estar relacionado mediante una composición a la clase <i>Definition</i> y debe tener asociado, como mínimo, una clase <i>Part</i> . Valores Etiquetados: <i>Name</i> , es obligatorio e indica el nombre de componente. <i>Documentation</i> , es opcional y contiene una descripción del componente.
Part	Representa un componente <i>Part</i> del metamodelo de WSDL.	Clase	<<PART>>	Restricciones: debe estar relacionado mediante una composición sólo a una clase <i>Message</i> . Valores Etiquetados: <i>Type</i> es opcional y se utiliza cuando el componente <i>Part</i> utiliza un tipo base (int, string, etc.) y no un tipo de dato definido a través de un componente <i>Element</i> . <i>Name</i> , es obligatorio e indica el nombre de componente. <i>Documentation</i> , es opcional y contiene una descripción del componente.

Concepto	Semántica	Notación		Restricciones y Valores Etiquetados
		Meta-clase de UML	Estereotipo	
PartType	Representan una relación entre un componente <i>Part</i> del metamodelo de WSDL y un componente <i>Element</i> .	Asociación	<<Part_Type>>	Restricciones: sólo se utiliza para asociar una clase <i>Part</i> con una clase <i>Element</i> cuando el componente <i>Part</i> utiliza el atributo <i>type</i> .
PartElement		Asociación	<<Part_Element>>	Restricciones: sólo se utiliza para asociar una clase <i>Part</i> con una clase <i>Element</i> sólo cuando el componente <i>Part</i> utiliza un <i>Element</i> .
Interface	Representa un componente <i>Interface</i> del metamodelo de WSDL.	Clase	<<INTERFACE>>	Restricciones: debe estar relacionado mediante una composición a la clase <i>Definition</i> y debe tener asociado, como mínimo una clase, <i>Operation</i> . Valores Etiquetados: <i>Name</i> , es obligatorio e indica el nombre de componente. <i>Documentation</i> , es opcional y contiene una descripción del componente.
Operation	Representa un componente <i>Operation</i> del metamodelo de WSDL.	Clase	<<OPERATION>>	Restricciones: debe estar relacionado mediante una agregación a una clase <i>Interface</i> . Valores Etiquetados: <i>Name</i> , es obligatorio e indica el nombre de componente. <i>Documentation</i> , es opcional y contiene una descripción del componente.
Input	Representan una relación entre un componente <i>Operation</i> y componente <i>Message</i> del metamodelo de WSDL.	Asociación	<<Input>>	Restricciones: sólo se utiliza para asociar una clase <i>Operation</i> con una clase <i>Message</i> cuando ésta representa un mensaje de entrada.
Output		Asociación	<<Output>>	Restricciones: sólo se utiliza para asociar una clase <i>Operation</i> con una clase <i>Message</i> cuando ésta representa un mensaje de salida.
Fault		Asociación	<<Fault>>	Restricciones: sólo se utiliza para asociar una clase <i>Operation</i> con una clase <i>Message</i> cuando ésta representa un mensaje de fallo o error.
Binding	Representa un componente <i>Binding</i> del metamodelo de WSDL.	Clase	<<BINDING>>	Restricciones: debe estar relacionado mediante una composición a la clase <i>Definition</i> y debe tener asociado una clase <i>Interface</i> . Valores Etiquetados: <i>Name</i> , es obligatorio e indica el nombre de componente. <i>Documentation</i> , es opcional y contiene una descripción del componente.
Service	Representa un componente <i>Service</i> del metamodelo de WSDL.	Clase	<<SERVICE>>	Restricciones: debe estar relacionado mediante una composición a la clase <i>Definition</i> y debe tener asociado mediante una composición, como mínimo, una clase <i>EndPoint</i> . Valores Etiquetados: <i>Name</i> , es obligatorio e indica el nombre de componente. <i>Documentation</i> , es opcional y contiene una descripción del componente.

Concepto	Semántica	Notación		Restricciones y Valores Etiquetados
		Meta-clase de UML	Estereotipo	
EndPoint	Representa un componente <i>EndPoint</i> del metamodelo de WSDL.	Clase	<<ENDPOINT>>	Restricciones: debe estar relacionado mediante una composición, solo a una clase <i>Service</i> y debe tener asociado una clase <i>Binding</i> . Valores Etiquetados: <i>Name</i> , es obligatorio e indica el nombre de componente. <i>Documentation</i> , es opcional y contiene una descripción del componente.
Import	Representa un componente <i>Import</i> del metamodelo de WSDL.	Clase	<<IMPORT>>	Restricciones: debe estar relacionado mediante una composición a la clase <i>Definition</i> . Valores Etiquetados: <i>Location</i> , de tipo URI, es opcional e identifica la localización de alguna información adicional para el espacio de nombres. <i>Namespace</i> , de tipo URI, es obligatorio e indica que el documento WSDL que lo contiene puede contener referencias a las definiciones WSDL de ese espacio de nombres. <i>Documentation</i> , es opcional y contiene una descripción del componente.

3.4.3.2 Modelo de Composición de Servicio Extendido

Este modelo extiende el modelo de composición de servicio propuesto a nivel PIM. Por tanto, en este modelo se representa también el proceso de composición de los servicios de una manera detallada y teniendo en cuenta detalles específicos que faciliten la implementación de la composición de servicios en una plataforma basadas en servicios Web.

Este modelo se representa mediante la técnica de diagrama de actividad de UML (OMG, 2003c) pero incluyendo, además, un nuevo elemento de modelado que extiende el diagrama de actividad tradicional de UML, llamado *Servicio Web*.

Elementos de Modelado

Como ya se ha dicho, este modelo extiende el modelo de composición de servicio de nivel PIM definido previamente, y por ello, también permite representar el proceso de composición de las distintas acciones que son necesarias para llevar a cabo un servicio del negocio, identificando quienes son los colaboradores que participan en la realización del mismo. Pero además, y dado que SOD-M define modelos específicos para una plataforma basada en servicios Web, en este modelo se propone indicar explícitamente aquellas acciones que son (o serán, en el caso de que aún no están implementadas) soportadas por servicios Web. Para ello, se define un nuevo elemento de modelado llamado *Servicio Web*, que es un tipo

especial de acción. Un *Servicio Web* se representa en este modelo como un nodo de actividad (*ActivityNode* en la Figura 3.10). El elemento *Servicio Web* tiene como valor etiquetado el conjunto de operaciones que realiza, describiendo el nombre de la operación y los parámetros de entrada y salida de cada una de ellas.

La Figura 3.10 presenta el metamodelo de composición de servicio extendido; en ella se destaca el nuevo elemento de modelado propuesto, que es un tipo especial de acción, y su relación con los elementos de modelado que utiliza el diagrama de actividad de UML. La Tabla 3.7 describe el elemento *Servicio Web* propuesto, indicando su semántica, notación, restricciones y valores etiquetados.

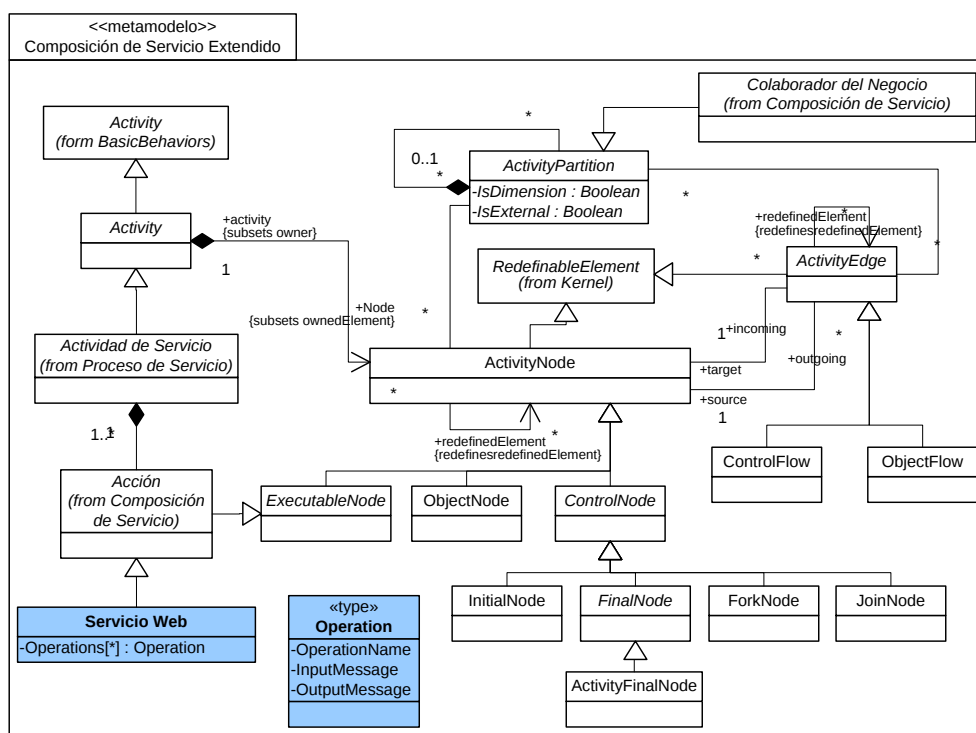


Figura 3.10. Metamodelo de Composición de Servicio Extendido

Tabla 3.7. Conceptos, semántica, notación y restricciones de los elementos del Modelo de Composición de Servicio Extendido

Concepto	Semántica	Notación		Restricciones y Valores Etiquetados
		Meta-clase de UML	Estereotipo	
Servicio Web	Representa una acción que es (o será) realizada por un Servicio Web.	Nodo de Actividad	<<WS>>	Restricciones: debe estar asociado a un colaborador del negocio. Valores Etiquetados: <i>Operations</i>, que incluye una lista de tipos de datos <i>Operation</i>. El tipo de dato <i>Operation</i> contiene los siguientes atributos: <i>OperationName</i> que describe el nombre de una operación que realiza el servicio Web, y <i>InputMessage</i>, <i>OutputMessage</i>, que indican los mensajes de entrada y de salida del servicio Web, respectivamente.

3.5 Perfil UML SOD-M

Un perfil UML es un paquete que contienen los elementos de modelado que han sido adaptados para un dominio o propósito específicos, usando los mecanismos de extensión como estereotipos, definición de valores etiquetados y restricciones (OMG, 2003c).

Todos los modelos propuestos por SOD-M a nivel PIM y PSM incluyen nuevos elementos de modelado específicos para el modelado del comportamiento de un SIW desde una aproximación orientada a servicios. Por ello, en este trabajo de Tesis Doctoral se define un perfil UML, llamado Perfil UML SOD-M, que agrupa todos estos elementos.

El perfil UML propuesto se estructura en dos paquetes, según se muestra en la Figura 3.11: *SOD-M/PIM*, que reúne los elementos de modelado a nivel PIM, y *SOD-M/PSM*, que reúne los elementos de modelado a nivel PSM.

- **Paquete SOD-M/PIM:** este paquete incluye todos los elementos de modelado propuestos por SOD-M para el modelado del comportamiento del sistema a nivel PIM. Dichos elementos extienden los elementos para el modelado de comportamiento propuestos por UML (OMG, 2003c). En la Figura 3.11 se muestran las dependencias de los elementos de modelado propuestos por SOD-M a nivel PIM, respecto de los paquetes que agrupan los elementos para el modelado del comportamiento en UML. Así, los nuevos elementos que forman parte del *metamodelo de casos de uso y de casos de uso extendido*, extienden la semántica de los elementos para el modelado de casos de uso de UML; y los nuevos elementos que forman parte del *metamodelo de procesos de servicio y de composición de servicio*, extienden los elementos para el modelado de actividades de UML.
- **Paquete SOD-M/PSM:** este paquete incluye todos los elementos de modelado propuestos por SOD-M para el modelado del comportamiento del sistema a nivel PSM. Dichos elementos extienden los elementos para el modelado estructural y de comportamiento propuestos por UML (OMG, 2003c). La Figura 3.11 muestra también las dependencias de los elementos de modelado propuestos por SOD-M a nivel PSM, respecto de los paquetes que agrupan los elementos estructurales y de comportamiento de UML. En este caso, los nuevos elementos que forman parte del *metamodelo de interfaz de servicio Web* extienden los elementos para el modelado de clases de UML, y los nuevos elementos que forman

parte del *metamodelo de composición de servicio extendido*, extienden los elementos para el modelado de actividades de UML.

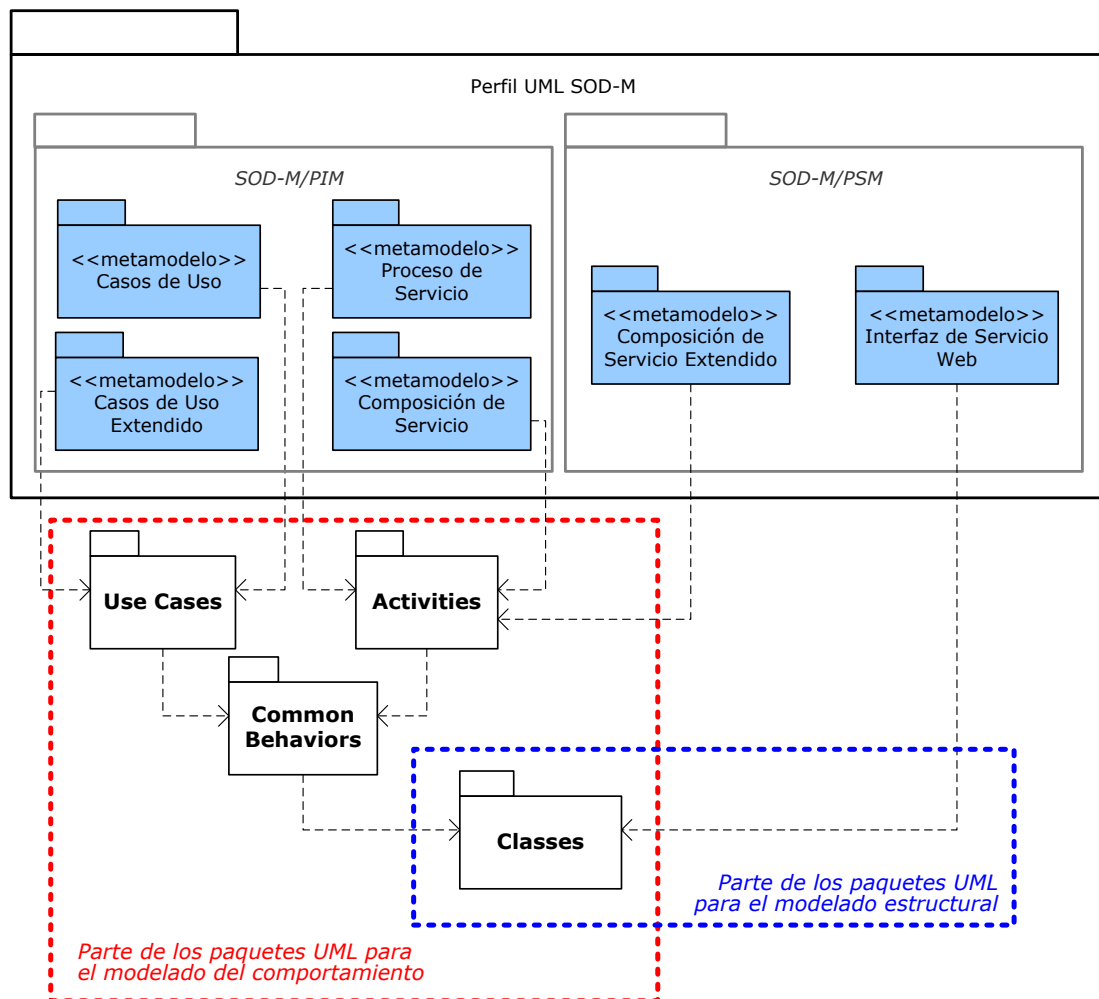


Figura 3.11. Dependencias de los elementos de modelado de SOD-M respecto de los paquetes UML para modelado del comportamiento y modelado estructural

La representación gráfica de un perfil UML se realiza utilizando la misma notación de un paquete UML y agregando el estereotipo *<<profile>>* (OMG, 2003c).

La Figura 3.12 presenta el perfil UML SOD-M completo; en ella se destacan cada uno de los nuevos elementos incluidos en el perfil, organizados en los dos paquetes mencionados previamente. La Figura 3.12 muestra, además, por cada uno de los elementos de modelado propuestos, la metaclase de UML que extiende, así como el estereotipo utilizado (sólo en los casos que sea necesario) para su representación en el modelo.

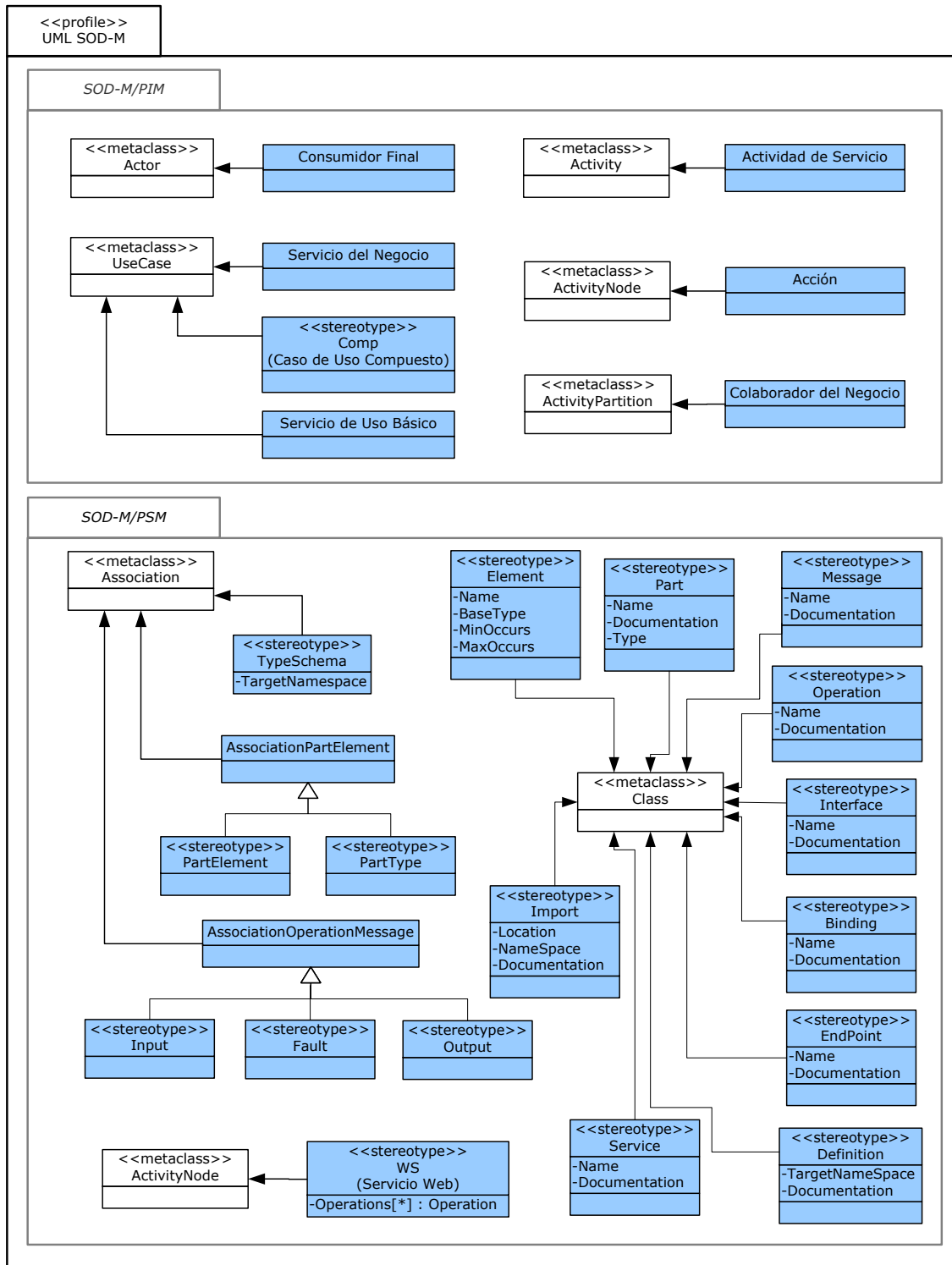


Figura 3.12. Perfil UML SOD-M

3.6 Proceso de SOD-M

SOD-M define un proceso dirigido por modelos para el desarrollo orientado a servicios de SIW. El proceso, recogido en la Figura 3.13 de forma esquemática, se inicia a partir del modelado de negocio de alto nivel y permite obtener como

resultado, modelos de comportamiento que facilitan la generación del SIW utilizando las tecnologías disponibles actualmente para desarrollo basado en servicios Web.

Este proceso de desarrollo, dirigido por modelos y basado en la arquitectura MDA, propone la *generación de un conjunto de modelos a distintos niveles de abstracción*, así como *transformaciones entre los mismos*. De esta manera, SOD-M permite alcanzar el principal objetivo de esta Tesis Doctoral, que es la alineación de procesos de negocio de alto nivel con plataformas específica basadas en servicios Web.

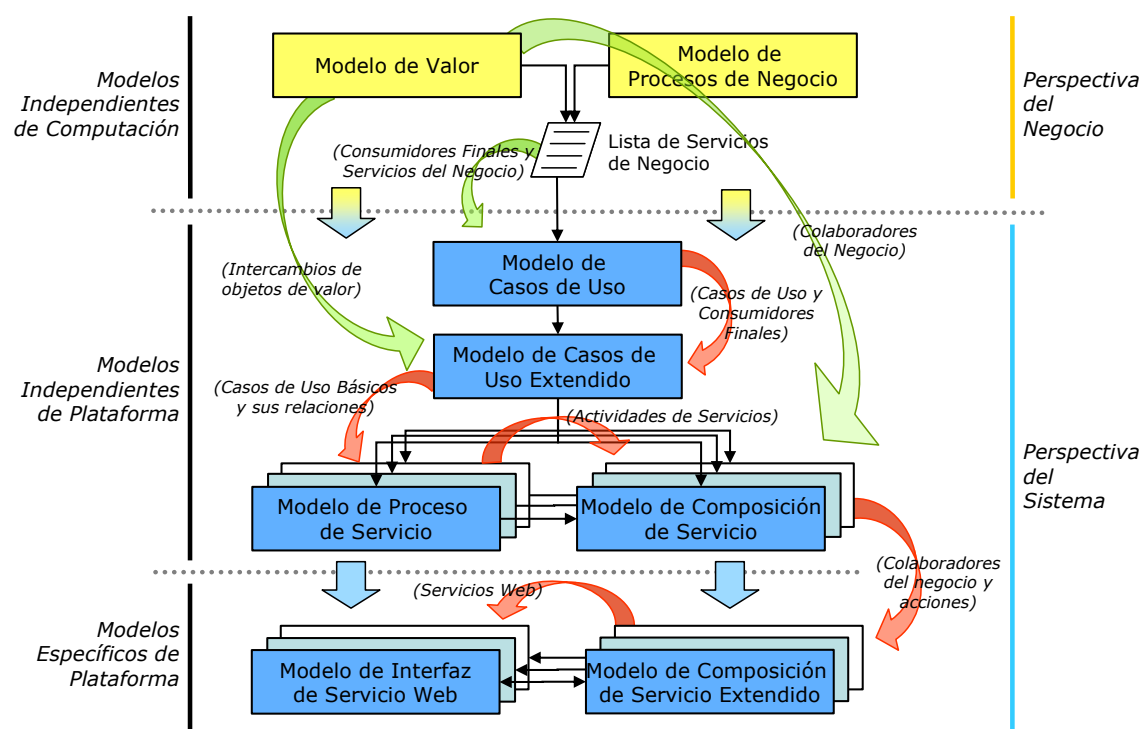


Figura 3.13. Proceso de SOD-M

La Figura 3.13 extiende la Figura 3.2 presentada en la sección 3.3 de este capítulo; en ella se muestran todos los modelos propuestos por SOD-M, tanto para el modelado del negocio como para el modelado del comportamiento, indicando, además, los conceptos de modelado que se trasladan de un modelo a otro (representados en los textos asociados a las flechas que van de un modelo a otro). Como puede verse en la Figura 3.13, algunos modelos se han representado como un conjunto de modelos; esto indica que durante el proceso de modelado del sistema se realizarán varios diagramas de ese modelo. Así, en el caso del *modelo de proceso de servicio*, del *modelo de composición de servicio*, y del *modelo de composición de servicio extendido*, se realizará un diagrama por cada servicio de

negocio que se implementará en el sistema; y, en el caso del *modelo de interfaz de servicios Web*, se realizará un diagrama por cada uno de los servicios Web que se implementarán en el sistema.

Además, como se ha explicado en la sección 2.1, MDA establece una diferencia entre las transformaciones de los modelos CIM al resto de los modelos, y las transformaciones entre los modelos de nivel PIM y PSM, tanto verticales como horizontales. Así, de acuerdo con esta diferencia, el proceso de SOD-M identifica dos tipos de transformaciones entre modelos: las *guías de transformación*, realizadas entre modelos de nivel CIM y PIM, que describen una relación de trazabilidad entre los conceptos del modelo de origen y destino, indicando cómo transformar los requisitos expresados en los modelos de nivel CIM en elementos de los modelos PIM (flechas verdes en la Figura 3.13); y, las *reglas de transformación entre modelos*, realizadas entre modelos de nivel PIM y PSM, que permiten transformar los elementos de un modelo origen en elementos de un modelo destino (flechas rojas en la Figura 3.13).

A continuación se presenta el proceso de SOD-M de manera detallada. Para ello, en primer lugar, se describen las actividades de generación de los distintos modelos propuestos por SOD-M. Cada actividad se divide en un conjunto de tareas asociadas, que, partiendo de un conjunto de elementos de entrada, genera un conjunto de elementos de salida. Posteriormente, en el apartado 3.6.2, se presenta el proceso de transformación entre modelos de SOD-M. En el se resumen las guías utilizadas para transformar los requisitos expresados en los modelos de nivel CIM en elementos de los modelos PIM, así como las reglas para la transformación entre modelos de nivel PIM y PSM. Tales reglas de transformación se especifican primero mediante lenguaje natural, y se describen después por medio de gramáticas de grafos.

3.6.1 Actividades de Generación de Modelos

Como se ha dicho en el apartado anterior, el proceso de SOD-M propone la generación de un conjunto de modelos a distintos niveles de abstracción, así como transformaciones entre los mismos. En este apartado se describen las tareas necesarias para la generación de los modelos propuestos por SOD-M. Cada una de las actividades de generación de modelos se presentará utilizando la siguiente estructura:

- *Artefactos de Entrada*, indicando qué artefactos (especificaciones o modelos) se utilizan como entrada para la generación del modelo.

- *Artefactos de Salida*, indicando qué artefactos (especificaciones o modelos) se generan como salida de la actividad.
- *Tareas*, especificando cada una de las tareas necesarias para la generación del modelo.

Cada una de estas actividades de generación de modelos se ilustra en la sección 4.3 donde se presenta uno de los casos de estudio realizado para la validación de esta Tesis Doctoral.

3.6.1.1 Generación del Modelo de Valor

Como se ha señalado en el apartado 3.4.1.1, donde se presentó el modelo de valor, este modelo se obtiene como resultado de la aplicación del método de modelado de negocio llamado *e³value*. Las tareas necesarias para la generación de este modelo se describen de manera detallada en (Gordijn y Akkermans, 2003). No obstante, se presenta aquí un resumen de las mismas utilizando la estructura antes definida.

- **Artefactos de Entrada:** especificaciones de la organización y el entorno de negocio.
- **Artefactos de Salida:** modelo de valor.
- **Tareas:** los autores de este modelo establecen las siguientes tareas para la construcción de un modelo de valor base:
 - *Identificación del escenario operacional.* Esta tarea puede realizarse describiendo de forma textual los productos, servicios o experiencias que desean los consumidores del negocio.
 - *Identificación de los actores.* Se propone realizar una lista tentativa de los actores basada, inicialmente, en los precursores del negocio y los consumidores finales del mismo.
 - *Identificar objetos del valor.* Esta tarea se realiza especificando qué ofrece o recibe cada uno de los actores identificados en el modelo: es decir, qué intercambio de objetos de valor ocurren entre los miembros del negocio.
 - *Identificar los caminos de dependencia.* Para ello será necesario identificar cuáles son las necesidades de los consumidores finales del negocio y qué intercambios de objetos de valor deben existir para poder satisfacer dicha necesidad.

3.6.1.2 Generación del Modelo de Proceso de Negocio

Este modelo se utiliza para entender y describir los procesos de negocio relacionados con el entorno en el que se utilizará el sistema que se desea construir. Para la representación de tales procesos se propone la utilización de un diagrama de actividad UML.

- **Artefactos de Entrada:** especificaciones de la organización y el entorno de negocio.
- **Artefactos de Salida:** modelo de proceso de negocio.
- **Tareas:**
 - *Identificación de las actividades del proceso de negocio.* Dichas actividades pueden ser muy variadas dependiendo tanto del entorno del negocio como de los procesos de negocio que se desean describir. Así, por ejemplo, las actividades podrían representar etapas de un proceso administrativo global de una organización, o bien los pasos concretos para la realización de una tarea u actividad específica de la organización.
 - *Identificación de flujos de trabajo.* Esta tarea se realizará especificando en el modelo el orden de realización entre cada una de las actividades, así como actividades que son alternativas o paralelas.

3.6.1.3 Generación de la Lista de Servicios de Negocio

Como una actividad previa a la generación de modelo de casos de uso, en el que se representan los servicios de negocio, SOD-M propone realizar una lista de servicios de negocio.

La lista de servicios de negocio es una descripción textual de todos los servicios que forma parte de un negocio y que se ofrecen para satisfacer una necesidad de un consumidor final del negocio. Así, esta lista incluirá el detalle de todos los consumidores de servicios, así como de los servicios de negocio que cada uno de ellos necesita.

- **Artefactos de Entrada:** modelo de valor y modelo de proceso de negocio.
- **Artefactos de Salida:** lista de servicios de negocio.

▪ **Tareas:**

- *Identificar los consumidores finales.* Los consumidores finales se identifican a partir de los caminos de dependencias representados en el modelo de valor. Los caminos de dependencias muestran los intercambios de objetos de valor que tiene lugar ante un determinado estímulo o necesidad de un consumidor final perteneciente al negocio. Así, gráficamente, los consumidores finales del negocio, son aquellos actores del modelo de valor que contienen un estímulo inicial (representado gráficamente como un círculo rojo), es decir aquellos que tienen una necesidad específica.
- *Identificar los servicios que requiere cada consumidor final del negocio.* Los servicios de negocio se pueden obtener a partir de los dos modelos de negocio propuestos, dado que ambos permite modelar servicios o actividades del negocio en el que se desarrollará el sistema que se va a construir. Dichos servicios se obtienen del siguiente modo:
 - *A partir del modelo de valor:* en este caso, los servicios de negocio a partir de los caminos de dependencia representados en el modelo. Cada estímulo inicial de un camino de dependencia, en el modelo de valor, representa una necesidad de un consumidor final; y cada necesidad un consumidor final da origen a un servicio del negocio. Así, los servicios del negocio, se identifican a partir de las necesidades o estímulos iniciales de los camino de dependencia, en el modelo de valor.
 - *A partir del modelo de proceso de negocio:* en este caso, es necesario identificar aquellas actividades del proceso de negocio permiten satisfacer una o varias necesidades de consumidores finales del negocio.

3.6.1.4 Generación del Modelo de Casos de Uso

El modelo de casos de uso se realiza mediante un diagrama de casos de uso UML y se utiliza para la representación de los servicios de negocio que requiere cada uno de los consumidores finales o usuarios del SIW. Dicho modelo se obtiene tomando como entrada la lista de servicios de negocio descrita anteriormente.

Cabe mencionar en este punto, que este modelo es el primer modelo del sistema que se construye, dado que, tanto el modelo de valor como el modelo de proceso de negocio y la lista de servicios de negocio, son modelos de negocio y, como tal, se centran en el análisis de las características del negocio en el que se desenvolverá el sistema, pero no en el sistema en sí mismo. Por tanto, para poder comenzar con el modelado del sistema, en este caso del comportamiento del mismo, será necesario primero identificar el alcance o límites del SIW que se quiere desarrollar. Esto implica determinar: *¿cuáles de los servicios del negocio identificados en la lista de servicios del negocio, serán soportados por el SIW?, y ¿cuáles de los consumidores finales identificados en la lista de servicios del negocio, serán consumidores finales o usuario de los servicios SIW?*. Respondiendo a estas preguntas es posible identificar los elementos de modelado que forman parte del modelo de casos de uso.

- **Artefactos de Entrada:** lista de servicios de negocio, especificación de requisitos del SIW.
- **Artefactos de Salida:** modelo de casos de uso.
- **Tareas:**
 - *Identificar los servicios de negocio como casos de uso.* Para realizar esta tarea será necesario determinar, cuáles de los servicios identificados en la lista de servicios del negocio serán soportados o provistos por el SIW; o lo que es lo mismo, determinar qué necesidades de los consumidores finales o usuarios del SIW podrán ser satisfechas a través del SIW.
 - *Identificar los consumidores finales del SIW como actores del modelo.* Para realizar esta tarea será necesario determinar, cuáles de los consumidores finales identificados en la lista de servicios de negocio, son consumidores de los servicios de negocio que se ofrecerán a través del SIW.
 - *Relacionar los actores con los casos de uso.* Esta relación se representará teniendo en cuenta cuáles de los servicios del negocio, representados como casos de uso, consumirá cada actor.

3.6.1.5 Generación del Modelo de Casos de Uso Extendido

Este modelo se utiliza para el modelado de las funcionalidades requeridas por el sistema para llevar a cabo los servicios de negocio, y se representa mediante un

diagrama de casos de uso UML que ha sido extendido para la representación de conceptos propios de SOD-M.

- **Artefactos de Entrada:** modelo de casos de uso, modelo de valor y especificación de requisitos del SIW.
- **Artefactos de Salida:** modelo de casos de uso extendido.
- **Tareas:**
 - *Identificar los casos de uso, tanto básicos como compuestos.* Para ello será necesario identificar, para cada uno de los servicios de negocio del modelo de casos de uso anterior, aquellas funcionalidades que son requeridas por el sistema para llevar a cabo el servicio de negocio. Dichas funcionalidades se pueden obtener de dos maneras complementarias:
 - *A partir del modelo de valor:* identificando las funcionalidades que son necesarias realizar en el sistema para llevar a cabo cada intercambio de objetos de valor que se produce ante una necesidad de un consumidor final.
 - *A partir de la especificación de requisitos del SIW que se va a construir:* teniendo en cuenta qué tareas es necesario realizar en el sistema para poder llevar a cabo los servicio de negocio.
 - *Descomponer los casos de uso compuestos.* Esta tarea se realiza mediante un proceso iterativo de descomposición de casos de usos compuestos en caso de usos más simples (básicos o compuestos). Dicha descomposición se repetirá hasta que sólo existan casos de uso básicos en el modelo.
 - *Identificar las relaciones de extensión e inclusión entre los casos de uso básicos.* Se representará una relación *include* entre dos casos de uso, cuando exista un flujo de eventos en el cual el caso de uso base incluya el comportamiento del otro; y, se representará una relación *extend*, cuando el comportamiento del caso de uso base pueda ser opcionalmente extendido por el comportamiento de otro caso de uso. Las relaciones de inclusión y extensión tienen aquí la misma semántica que en el modelo de casos de uso tradicional de UML.

3.6.1.6 Generación del Modelo de Proceso de Servicio

El modelo de proceso de servicio se utiliza para representar el flujo de trabajo necesario para la realización de un servicio de negocio en el sistema. Este modelo se representa mediante la técnica de diagrama de actividad de UML y, como se ha dicho antes, SOD-M propone realizar varios de estos diagramas, uno por cada uno de los servicios de negocio que se implementarán en el sistema.

- **Artefactos de Entrada:** modelo de casos de uso extendido, especificación de requisitos del SIW.
- **Artefactos de Salida:** modelo de proceso de servicio.
- **Tareas:**
 - *Identificar las actividades de servicio del proceso.* Las actividades de servicio se obtienen a partir de los casos de uso básicos representados en el modelo de casos de uso extendido. Cada caso de uso básico da lugar a una actividad de servicio en este modelo; de esta manera, cada actividad del modelo representa un comportamiento que forma parte del flujo de trabajo necesario para la realización de un servicio de negocio.
 - *Identificar el flujo de trabajo entre las actividades de servicios.* Esta tarea se realizará especificando el orden de realización entre cada una de las actividades de servicio, así como actividades alternativas o paralelas.

3.6.1.7 Generación del Modelo de Composición de Servicio

Este modelo extiende modelo de proceso de servicio descrito anteriormente y se utiliza para la representación del proceso de composición de las distintas acciones que son necesarias para llevar a cabo un servicio del negocio, indicando, además, cuáles son los miembros del negocio que realizan cada acción. Este modelo se representa mediante la técnica de diagrama de actividad de UML, y se propone realizar varios de estos diagramas, uno por cada uno de los servicios de negocio.

- **Artefactos de Entrada:** modelo de proceso de servicio, modelo de valor y especificación de requisitos del SIW.
- **Artefactos de Salida:** modelo de composición de servicio.

▪ **Tareas:**

- *Identificar los colaboradores del negocio.* Un colaborador del negocio representa una entidad que realiza alguna de las acciones que son necesarias para llevar a cabo un servicio de negocio. Los colaboradores del negocio se identifican a partir del modelo de valor, teniendo en cuenta aquellos actores, segmentos de mercado y actividades de valor que ofrecen o reciben objetos de valor ante una necesidad de un consumidor. Así, gráficamente, los colaboradores del negocio que participan en la realización de un servicio de negocio, pueden verse en el modelo de valor, como aquellos actores que son atravesados por un camino de dependencia. Los colaboradores del negocio se representa como particiones en el diagrama de actividad.
- *Identificar colaboradores del negocio que son externos al sistema.* Aquellos colaboradores que son externos al sistema que se está modelando, se representarán en el diagrama de actividad como particiones estereotipadas con <<external>>.
- *Identificar las acciones del modelo.* Una acción es una unidad de comportamiento fundamental que forma parte de una actividad de servicio, y que representa alguna transformación o procesamiento en el sistema que se está modelando. Las acciones de este modelo se obtienen a partir de las actividades de servicio, que se representan en el modelo de proceso de servicio, identificando el conjunto de acciones que son necesarias para la realización de cada una ellas. Cada acción se representará como una actividad en el diagrama de actividad, y se ubicará en la partición que corresponda según quien sea el colaborador del negocio que realiza la acción.
- *Identificar el flujo de datos y de control entre las acciones del modelo.* Determinando flujos secuenciales, alternativos, paralelos, bifurcaciones, etc. entre las acciones. Tales flujos pueden ser derivados a partir del modelo de proceso de servicios realizado previamente.

3.6.1.8 Generación del Modelo de Composición de Servicio Extendido

Este modelo extiende el modelo de composición de servicio que se realiza a nivel PIM, y permite la representación del proceso de composición de un servicio de negocio pero teniendo en cuenta detalles específicos de una plataforma basada en

servicios Web. El modelo permite representar aquellas acciones que serán implementadas como servicios Web, indicando además el nombre la operación y los parámetros de los mensajes de entrada y salida que utilizará. Este modelo se representa mediante un diagrama de actividad de UML extendido. En este caso, también se realizan varios diagramas, uno por cada uno de servicio de negocio que se implementará en el sistema.

- **Artefactos de Entrada:** modelo de composición de servicio, especificación de requisitos del SIW.
- **Artefactos de Salida:** modelo de composición de servicio extendido.
- **Tareas:**
 - *Identificar las acciones que serán implementadas como servicios Web.* En este caso, será el diseñador del sistema quien decida qué acciones del proceso de composición de un servicio de negocio se implementarán como servicios Web; para ello, se podrán tener en cuenta los siguientes criterios principales: la *reutilización de cada acción* (por ejemplo, una acción que ejecute el login de un cliente), la *posibilidad de comunicación* con colaboradores externos (por ejemplo, una acción que realiza un colaborador externo al sistema), y *la posibilidad de realizar llamadas a servicios Web* ya implementados (por ejemplo, una acción que comprueba la validez de una tarjeta de crédito). Tales acciones se representarán en este modelo como *Servicios Web* utilizando el estereotipo <<WS>>. Se especificará, además, las operaciones que realizará cada servicio Web, indicando por cada una de ellas, el nombre y los mensajes de entrada y de salida que utilizará.

3.6.1.9 Generación del Modelo de Interfaz de Servicio Web

Este modelo, que se utiliza para representar de manera gráfica la interfaz un servicio Web, se realiza a través de un diagrama de clases de UML que ha sido extendido para la representación de conceptos propios del estándar WSDL. Como se ha dicho antes, SOD-M propone realizar varios de estos diagramas, en concreto, uno por cada uno de los servicios Web que se implementarán en el sistema; es decir, uno por cada elemento Servicio Web identificado en el modelo de composición de servicios extendido.

- **Artefactos de Entrada:** modelo de composición de servicio extendido, especificación de requisitos del SIW.

- **Artefactos de Salida:** modelo de interfaz de servicio Web.
- **Tareas:**
 - *Identificar un elemento Operation por cada operación que realiza el Servicio Web.* Este elemento se obtiene a partir del modelo de composición de servicio extendido. En concreto, se representará un elemento *Operation* en este modelo, por cada conjunto de valores (*OperationName*, *InputMessage*, *OutputMessage*) que aparezcan en el elemento *Servicio Web* del modelo de composición de servicio extendido. Cada elemento *Operation* tendrá como nombre el contenido del atributo *OperationName*.
 - *Identificar los elementos Message y Part para cada operación que realiza el Servicio Web.* Los elementos *Message* se obtienen también a partir del modelo de composición de servicio extendido. Así, por cada conjunto de valores (*OperationName*, *InputMessage*, *OutputMessage*) se representarán dos elementos *Message*. Uno de ellos tendrá como nombre el contenido del atributo *InputMessage* y estará relacionado con el elemento *Operation* asociado, mediante una asociación *Input*; el otro elemento tendrá como nombre el contenido del atributo *Output Message* y estará relacionado con el elemento *Operation* asociado, mediante una asociación *Output*. Posteriormente se representarán los elementos *Part*, que se corresponderán con los parámetros de entrada y salida de cada operación. Dichos elementos *Part* estarán relacionados a los elementos *Message* según corresponda.
 - *Identificar el resto de los elementos del modelo.* Para ello se tendrán en cuenta las especificaciones de cada uno de los elementos presentada en la Tabla 3.6 del apartado 3.4.3.1.

Para finalizar este apartado, cabe mencionar que, en todos los casos, la construcción de los modelos es un proceso cíclico. Esto significa que las tareas necesarias para la generación de un modelo pueden repetirse varias veces con el objetivo de ir refinando o corrigiendo el modelo.

3.6.2 Transformación de Modelos en SOD-M

Según la propuesta de MDA (OMG, 2001), un proceso de desarrollo dirigido por modelos precisa, no sólo de modelos a diferentes niveles de abstracción, sino también de transformaciones entre los mismos, las cuales permitan transformar los elementos de un modelo de entrada en elementos de otro, u otros modelos de salida, diferentes entre sí.

Como ya se ha explicado antes, en MDA, las reglas de transformación se aplican entre modelos de nivel PIM y PSM, pudiéndose definir transformaciones tanto verticales (PIM-PSM, PSM-PIM), como horizontales (PIM-PIM, PSM-PSM). Pero además, tal como propone MDA, debe existir una trazabilidad entre los requisitos del sistema expresados en los modelos CIM, y los elementos o artefactos representados en los modelos PIM y PSM que permiten llevar a cabo tales requisitos, aunque en este caso, no sea posible la definición de reglas que permitan automatizar la transformación.

Por esta razón, el proceso de SOD-M identifica dos tipos de transformaciones entre modelos: las *guías de transformación*, realizadas entre modelos de nivel CIM y PIM, y las *reglas de transformación*, realizadas entre modelos de nivel PIM y PSM (tanto verticales como horizontales). Ambos tipos de transformaciones se describen en los siguientes apartados de esta sección.

3.6.2.1 Guías de Transformación

De acuerdo con la propuesta de MDA, SOD-M define guías para que los requisitos expresados en los modelos de nivel CIM puedan ser posteriormente identificados mediante los elementos de los modelos PIM.

Las guías de transformación definidas por SOD-M se presentan en la Tabla 3.8; en ella se incluyen cada uno de los elementos de modelado de nivel PIM que se obtienen a partir de los modelos de nivel CIM. Las guías de transformación que se recogen en esta tabla, se han ido describiendo ya, de manera informal, junto con las actividades de generación de algunos de los modelos de nivel PIM.

Tabla 3.8. Guías de Transformación en SOD-M.

Modelos de Nivel CIM		Modelos de Nivel PIM		
Modelos	Elementos de Modelado	Modelo de Casos de uso	Modelo de Proceso de Servicio	Modelo de Composición de Servicio
<i>Modelo de Valor</i>	Actor (que es el origen de un camino de dependencia)	Consumidor Final		Colaborador del Negocio
	Estímulo inicial (de cada camino de dependencia)	Servicio de Negocio		
	Intercambios de Objetos de Valor (que se producen ante una necesidad de un consumidor final)		Actividad de Servicio	
	Segmento de Mercado			Colaborador del Negocio
	Actividad de Valor			Colaborador del Negocio
<i>Modelo de Proceso de Negocio</i>	Actividad	Servicio de Negocio		

3.6.2.2 Reglas de Transformación: Lenguaje Natural

En este apartado se presentan las reglas de transformación propuestas por SOD-M, incluyendo tanto las transformaciones horizontales, es decir, entre modelos de nivel PIM y entre modelos de nivel PSM, como las transformaciones verticales, de nivel PIM a PSM.

Tales reglas, se describen en lenguaje natural en la Tabla 3.9. Dicha tabla incluye, además, una columna en la que se indica el grado de automatización de la regla: completo (C) o parcial (P). Se dice que el grado de automatización de una regla es parcial, cuando se requiere la intervención de un diseñador para su aplicación; por el contrario, se dice que el grado de automatización de una regla es completo, cuando dicha intervención no es necesaria.

Tabla 3.9. Reglas de Transformación en SOD-M

Modelo Origen	Modelo Destino	Reglas de Transformación	Grado de Automatización
Transformaciones de Modelos PIM-PIM			
Modelo de Casos de Uso	Modelo de Casos de Uso Extendido	1. Cada Servicio de Negocio (SN) del modelo de casos de uso se deberá descomponer en uno o más Casos de Uso Básicos (CUB) o Compuestos (CUC).	P
		2. Cada CUC que se obtenga se volverá a descomponer en uno o más CUB.	P
Modelo de Casos de Uso Extendido	Modelo de Proceso de Servicio	3. Cada CUB que se correspondiente a un mismo SN, se representará como una Actividad de Servicio (AS) en el diagrama correspondiente a ese SN.	C
		4. Cada asociación Extend identificada en el modelo de casos de uso extendido se representará en el modelo de proceso de servicio como una bifurcación. La AS correspondiente al CUB origen de la asociación Extend será previa a la AS correspondiente al CUB destino de la asociación Extend.	C
		4.1 Si la asociación Extend tiene solo un CUB origen, la bifurcación presentará una AS como una alternativa a otro flujo que no tiene AS. Posteriormente ambos flujos se encontrarán.	C
		4.2 Si la asociación Extend tiene varios CUB origen, la bifurcación presentará las diferentes AS como diferentes alternativas a otro flujo sin AS. Posteriormente todos los flujos se encontrarán.	C
		5. Cada asociación Incluye identificada en el modelo de casos de uso extendido se representará en el modelo de proceso de servicio como un flujo. La AS correspondiente al CUB origen de la asociación Incluye será la siguiente a la AS correspondiente al CUB destino de la asociación.	C
		5.1 Si la asociación Incluye tiene varios destinos, el diseñador debe decidir la secuencia apropiada para las diferentes AS correspondientes a los CUB de destino (que serán, obviamente, previos a la AS correspondiente al CUB origen).	P
Modelo de Proceso de Servicio	Modelo de Composición de Servicio	6. Cada AS del modelo de proceso de servicio de un SN se deberá descomponer en una o más Acciones en el modelo de composición de servicio del mismo SN.	P
		7. El flujo de control entre las Acciones se corresponderá con el flujo de sus respectivas AS en el modelo de proceso de servicio.	C
		7.1 Si existen dos o más acciones pertenecientes a una misma AS, será el diseñador quien decida el orden particular en las acciones identificadas.	P
Transformaciones de Modelos PIM-PSM			
Modelo de Composición de Servicio	Modelo de Composición de Servicio Extendido	8. Cada acción que vaya a implementarse como un servicio Web (SW) será representada en el modelo de composición de servicio extendido como un SW.	P
		8.1 Se especificará las operaciones que realizará cada SW indicando por cada una de ellas, el nombre y los mensajes de entrada y de salida que utilizará.	P

Modelo Origen	Modelo Destino	Reglas de Transformación	Grado de Automatización
Transformaciones de Modelos PSM-PSM			
Modelo de Composición de Servicios Extendido	Modelo de Interfaz de Servicio Web	9. Por cada conjunto de valores (OperationName, InputMessage, OutputMessage), contenido en el valor etiquetado Operations de cada elemento SW del modelo de composición de servicio extendido, se representará en el modelo de interfaz de servicio Web: un elemento Operation (OP) y dos elementos Message (MSS).	C
		9.1 Cada elemento OP tendrá como nombre el contenido del atributo OperationName.	C
		9.2 Se representará un elemento MSS que tendrá como nombre el contenido del atributo InputMessage y estará relacionado con el elemento OP asociado, mediante una asociación Input.	C
		9.3 Se representará un elemento MSS que tendrá como nombre el contenido del atributo OutputMessage y estará relacionado con el elemento OP asociado, mediante una asociación Output.	C

Con el fin de clarificar las reglas asociadas a las transformaciones de las relaciones *Include* y *Extend* procedentes del modelo de casos de uso extendido (reglas 4 y 5), en la Tabla 3.10 se recoge la especificación de las mismas, indicando los elementos de los modelos de origen y destino involucrados en las transformaciones, así como diagramas parciales que facilitan la comprensión de estas reglas.

Tabla 3.10. Reglas de transformación de las relaciones *extend* e *include* del Modelo de Casos de Uso Extendido al Modelo de Proceso de Servicio

Reglas	Representación	
	Elementos del Modelo de Casos de Uso Extendido	Elementos del Modelo de Proceso de Servicio
Regla 4.1 Si la asociación Extend tiene solo un CUB origen, la bifurcación presentará una AS como una alternativa a otro flujo que no tiene AS. Posteriormente ambos flujos se encontrarán.		
Regla 4.2 Si la asociación Extend tiene varios CUB origen, la bifurcación presentará las diferentes AS como diferentes alternativas a otro flujo sin AS. Posteriormente todos los flujos se encontrarán.		
Regla 5 Cada asociación Include identificada en el modelo de casos de uso extendido se representará en el modelo de proceso de servicio como un flujo. La AS correspondiente al CUB origen de la asociación Include será la siguiente a la AS correspondiente al CUB destino de la asociación.		

Reglas	Representación	
	Elementos del <i>Modelo de Casos de Uso Extendido</i>	Elementos del <i>Modelo de Proceso de Servicio</i>
Regla 5.1 Si la asociación Include tiene varios destinos, el diseñador debe decidir la secuencia apropiada para las diferentes AS correspondientes a los CUB de destino (que serán, obviamente, previos a la AS correspondiente al CUB origen).		

3.6.2.3 Reglas de Transformación: Gramáticas de Grafos

De acuerdo con los principios de MDA, las transformaciones entre modelos deben ser automatizadas, al menos en cierta medida. Así, de cara a poder automatizar en un futuro las reglas de transformación propuestas por SOD-M, en este trabajo de Tesis Doctoral se ha considerado, como una primera aproximación, la posibilidad de formalizar dichas reglas de transformación por medio de gramáticas de grafos (Czarnecki y Helsen, 2003; Tratt, 2005); dado que resultan bastante intuitivas para comprender y describir las transformaciones entre modelos y, además, debido a que proyectos como *AGG System* (Taentzer, 2000) proporcionan la funcionalidad necesaria para automatizar dichas transformaciones.

Un diagrama de clases de UML se puede considerar como un grafo, dado consta de un conjunto de clases (nodos) que se conectan mediante asociaciones (aristas). Debido a que SOD-M propone la utilización de modelos UML para los niveles PIM y PSM y, que todos los metamodelos de dichos modelos se describen a través de diagramas de clases, resulta intuitivo expresar las reglas de transformación definidas previamente entre los modelos mediante grafos.

La expresión de las transformaciones de modelos como gramáticas de grafos debe realizarse mediante la definición de un conjunto de reglas con la estructura *LHS:=RHS*. En la parte izquierda (LHS) se define un (sub)grafo patrón y en la parte derecha (RHS) se define un (sub)grafo de sustitución. De esta forma, cuando se aplica la regla, el grafo origen (o modelo UML origen) se sustituye, en base al patrón definido en LHS, por el grafo de sustitución (o modelo UML destino) (Sendall y Kozaczynski, 2003).

La especificación de las reglas de transformación mediante grafos se realizará en base a las siguientes directrices (Cáceres *et al.*, 2006a):

- Los nodos en el LHS se identifican con números consecutivos, lo que hace posible la correspondencia con sus respectivos nodos en el RHS.
- La referencia a un nodo del LHS, en el RHS, se realiza a través de la expresión `match(x)`, donde `x` es el número del nodo del LHS.
- Todas las propiedades de los diferentes nodos siempre tendrán asignado un valor inicial. Para indicar que es válido cualquier valor, se utiliza "???".
- Para referenciar a un atributo de un nodo LHS, se usará la notación `"."`, por ejemplo `match(x).name`.
- Igual que los nodos pertenecientes al LHS, los nodos pertenecientes al RHS también han de ser numerados. Hay que tener en consideración las siguientes situaciones:
 - Si el número de nodo del LHS aparece en el RHS, el tipo de nodo de un lado es igual al otro.
 - Si un número de nodo aparece en el LHS pero no en el RHS, el nodo del LHS deberá ser eliminado en el modelo destino así como las relaciones en las que participa.
 - Si un número de nodo aparece en el RHS, que no estaba en el LHS, hay que añadir este nuevo nodo en el modelo destino.
 - Si un número de nodo que aparece en el LHS aparece en el RHS con apóstrofe (`x'`), significa que el tipo de nodo en el RHS es diferente, aunque se preservan las conexiones con el resto de elementos.

De acuerdo con estas directrices, en este trabajo de Tesis Doctoral se define un conjunto de reglas de transformación basadas en grafos para las transformaciones automatizables entre modelos propuestos por SOD-M. Tales reglas de transformación, tanto verticales (entre modelos de nivel PIM y PSM) como horizontales (entre modelos de nivel PIM, y entre modelos de nivel PSM), se incluyen en el apéndice C de esta memoria de Tesis Doctoral.

En la Figura 3.14 se presenta, a modo de ejemplo, dos de las reglas de transformación entre modelos de nivel PIM, las reglas 1 y 2 de la Tabla 3.9.

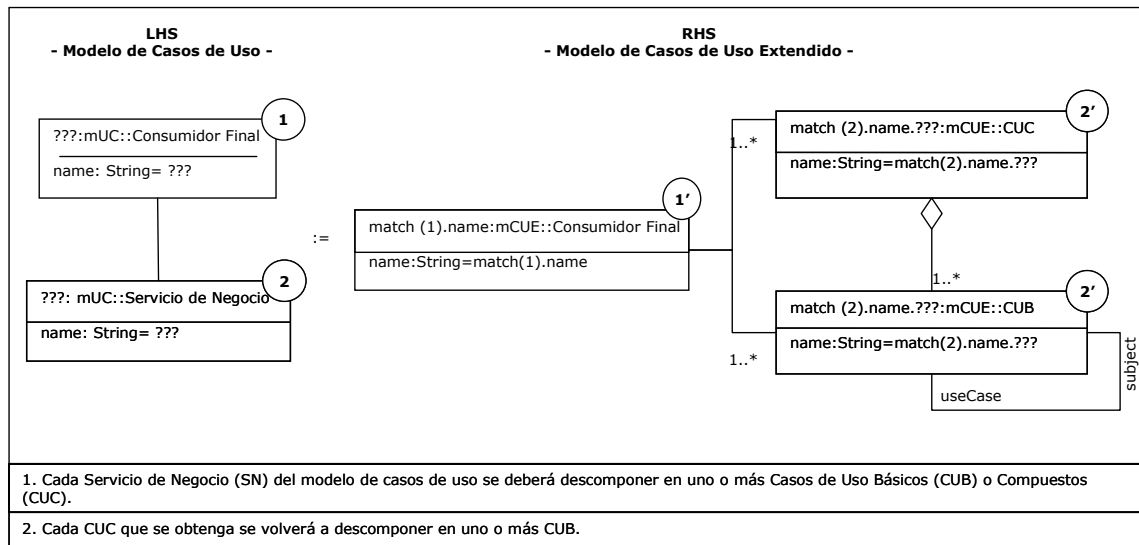


Figura 3.14. Transformación de *Servicios de Negocios* y *Consumidor Final* del metamodelo de Casos de Uso a *Casos de Uso Básicos* y *Compuestos* en el metamodelo de Casos de Uso Extendido

3.7 Integración de SOD-M en la Arquitectura de MIDAS

SOD-M se integra en la arquitectura de MIDAS, una arquitectura basada en MDA para el desarrollo de SIW (Marcos *et al.*, 2004; Cáceres *et al.*, 2003). Como se descrito en la sección 1.1 del capítulo 1, la arquitectura de modelos de MIDAS (Figura 3.15), contempla tres aspectos básicos para el modelado y desarrollo de SIW: *hipertexto*, *contenido* y *comportamiento*. SOD-M se centra en el aspecto del comportamiento. En esta sección se describirán las principales guías para la integración de SOD-M en el marco de la arquitectura de MIDAS, teniendo en cuenta los demás aspectos contemplados en dicha arquitectura para el desarrollo de SIW.

Como puede verse en la Figura 3.15, la arquitectura de MIDAS considera el aspecto de hipertexto como un eje vertebrador para el desarrollo del SIW. Dado que, en el marco de esta Tesis Doctoral, se propone un enfoque orientado a servicios para la construcción de todo el SIW, se hace necesario describir cómo se realiza la integración entre los modelos del aspecto de comportamiento propuestos por SOD-M y los modelos propuestos para el modelado del hipertexto en MIDAS.

En la Figura 3.15 se muestran todos los modelos que son necesarios para el desarrollo del hipertexto en el marco de la arquitectura de MIDAS. En ella se incluyen los siguientes modelos: *modelo de valor* y *modelo de procesos de negocio*, propuestos para el modelado del negocio a nivel CIM; *modelo de casos de uso*, *modelo de casos de uso extendido*, *modelo de proceso de servicio*, *modelo de composición de servicio*, para el modelado aspecto del comportamiento a nivel PIM; *modelo conceptual de datos*, para el modelado del aspecto del contenido a nivel

PIM; y finalmente, *modelo de fragmentos extendidos* y *modelo de navegación extendidos*, para el modelado del aspecto del hipertexto a nivel PIM.

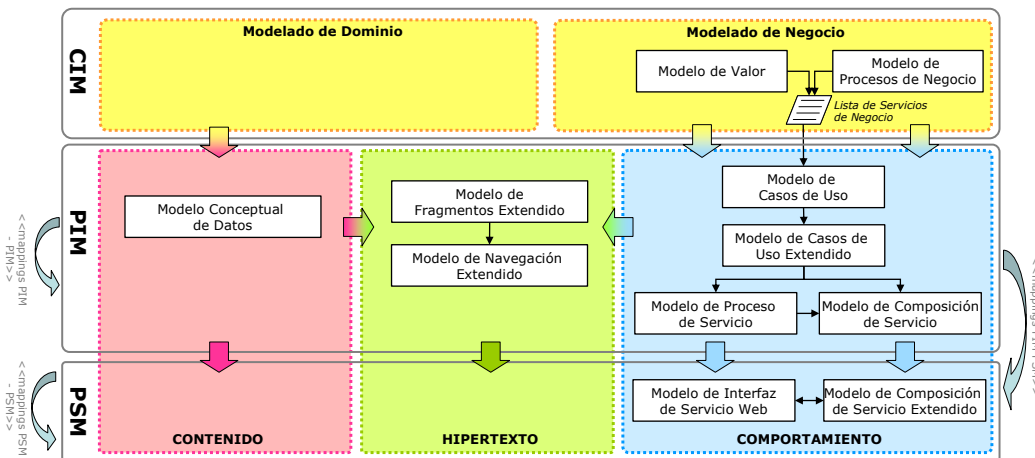


Figura 3.15. Arquitectura de Modelos de MIDAS

En el apartado 3.7.1, se describen las principales características del método para el modelado del hipertexto en MIDAS y se presentan los modelos del aspecto del hipertexto. Posteriormente, en el apartado 3.7.2, se describen las guías para la obtención de tales modelos a partir de los modelos del comportamiento propuestos por SOD-M.

3.7.1 Modelado del Hipertexto de MIDAS

El objetivo principal del método para el modelado del hipertexto de MIDAS es la obtención de un modelo navegacional que facilite al usuario la navegación a través del SIW. Por este motivo, en el marco de MIDAS, se propone un método para el desarrollo del hipertexto de SIW basado en los servicios que el usuario requiere del sistema.

Como se ha explicado en el capítulo 2, la mayoría de los métodos tradicionales para el desarrollo de SIW proponen la construcción del hipertexto en base a los objetos "cosas" del dominio (Isakowitz *et al.*, 1998; Ceri *et al.*, 2000; Koch *et al.*, 2004; Schwabe *et al.*, 2001; Pastor *et al.*, 2001; Fraternali y Paolini, 2000). De este modo, se obtienen modelos de hipertexto con una perspectiva estructural donde, en el mejor de los casos, los servicios y la funcionalidad asociada al sistema se añaden posteriormente al modelo estructural de hipertexto. A diferencia de estos enfoques, MIDAS define una nueva perspectiva orientada a servicios para el modelado del hipertexto (Cáceres *et al.*, 2004; De Castro *et al.*, 2004). Esta aproximación propone modelar el hipertexto en base a los servicios que

el usuario requiere del sistema, así como en los procesos necesarios para llevar a cabo los mismos.

Para ello, el método del modelado del hipertexto propuesto por MIDAS, parte del *modelo de casos de uso* (definido en la sección 3.4.2.1), en el que se representan los servicios de negocio, es decir, los servicios que son requeridos por los usuarios o consumidores del SIW. Además, MIDAS propone obtener, a partir del *modelo del proceso de servicio* (definido en la sección 3.4.2.3), las rutas que el usuario del SIW debe seguir para llevar a cabo con éxito cada servicio que espera del mismo. Dichas rutas se representan en el modelo de navegación, permitiendo de esta manera el desarrollo de SIW más intuitivos y fáciles de navegar por parte de los usuarios (De Castro *et al.*, 2006b).

A continuación, se resumen las principales características de los modelos del aspecto del hipertexto propuestos por MIDAS: el *modelo de fragmentos extendido* y el *modelo de navegación extendido*.

3.7.1.1 Modelo de Fragmentos Extendido

El modelo de fragmentos fue propuesto inicialmente por la metodología RMM (Isakowitz *et al.*, 1995), y se define como la descomposición del sistema en unidades de información significativas, denominadas fragmentos, y los hiperenlaces entre dichos fragmentos. El modelo de fragmentos extendido, definido en el marco de MIDAS, extiende los conceptos del modelo de fragmentos de RMM y, además, a diferencia de este, se representa utilizando notación UML.

El modelo de fragmentos extendido se representa a través de un diagrama de clases, en el cual los fragmentos se muestran como clases estereotipadas y los hiperenlaces como relaciones entre clases. En dicho modelo, cada fragmento representa una unidad de información que será mostrada en la Web.

En el modelo de fragmentos extendido se identifican dos tipos de fragmentos: estructurales y funcionales. Un *fragmento estructural* es un fragmento tal y como se mencionó en apartado anterior, por lo que representa una unidad de información que será mostrada en la Web. Un *fragmento funcional* es un fragmento en el que, además de la información que será mostrada en la Web, pueden aparecer distintas funcionalidades asociadas al fragmento e información necesaria para llevar a cabo las mismas; así, un fragmento funcional permite la representación de una interacción del usuario con el SIW. Los diferentes tipos de fragmentos se muestran en este modelo como clases estereotipadas con <<SS>> y <<FS>>, las cuales

permiten representar respectivamente los fragmentos estructurales (*Structural Slice*) y fragmentos funcionales (*Functional Slice*).

En el modelo de fragmentos extendido se identifican, además, las *rutas*. Una ruta se define en MIDAS como la secuencia de pasos que el usuario debe seguir para llevar a cabo con éxito cada servicio que espera del mismo; por lo que en este modelo, se representa una ruta por cada servicio de negocio que se implementará en el SIW. Tales rutas se obtienen a partir del modelo de proceso de servicio asociado a cada servicio de negocio, y se representa en el modelo de fragmentos extendido como un conjunto de hiperenlaces que enlazan, con una relación dirigida, los fragmentos por los que el usuario tendrá que pasar. Cada ruta queda identificada en este modelo con el estereotipo <<route>> y el valor etiquetado {*NombreRuta*}. Una ruta puede además tener caminos alternativos o subrutas. En ese caso, si la ruta *X* tiene la subruta *Y*, dicha subruta se estereotipa como <<route>> y el valor etiquetado {*X.Y*}.

El hecho de señalar las rutas que un usuario deberá seguir, para la realización de cada servicio de negocio en la estructura navegacional del SIW, le permite al usuario saber en cada momento qué hiperenlace debe tomar, cuál es la próxima actividad a realizar y cuántas actividades le quedan por realizar; evitando así que el usuario se pierda en el espacio de navegación. De esta manera, es posible construir SIW más intuitivos y fáciles de navegar por parte de los usuarios.

3.7.1.2 Modelo de Navegación Extendido

El modelo de navegación extendido se obtiene añadiendo las estructuras de navegación (por ejemplo: menús, índices, etc.) al modelo de fragmentos extendido descrito previamente. Las estructuras de navegación que se utilizan en este modelo se representan utilizando los estereotipos definidos en la metodología UWE (Koch *et al.*, 2000). De este modo, el modelo de navegación extendido incluye: un conjunto de *fragmentos estructurales y funcionales*; las *rutas* que permiten enlazar los fragmentos; y las siguientes *estructuras de navegación*: *menu* (que se representa un menú de la aplicación Web), *index* (que representa un índice que se muestra en la aplicación Web), *query* (que representa una consulta que se realiza para mostrar una unidad de información en la aplicación Web) y *guide tours* (que representa un acceso secuencial sobre distintas unidades de información en la aplicación Web).

El método del modelado del hipertexto de MIDAS propone introducir un menú principal en el que se represente una entrada por cada servicio de negocio identificado en el SIW. Este menú supone el inicio de cada una de las rutas de servicio identificadas en el modelo de fragmentos extendido. Cada ruta, que en este

modelo también debe quedar identificada, debe ser tenida en cuenta en la implementación. Así, las paginas Web del SIW, obtenidas a través de las clases del modelo navegacional estereotipadas con <<FS>> y <<SS>>, deberán mostrar la ruta que el usuario está realizando; de esta manera el SIW estará preparado para que el usuario realice cada uno de los servicios que desea, siguiendo la ruta de ese servicio, sin perderse durante la navegación.

Este modelo y el anterior se ilustrarán también en la sección 4.3, en la que se presenta un SIW desarrollado como casos de estudios de esta Tesis Doctoral.

3.7.2 Obtención de Modelos del Hipertexto a partir de Modelos Propuestos por SOD-M

Como se ha explicado en la sección 3.7.1, el método del modelado del hipertexto de MIDAS propone construir los modelos de nivel PIM para el aspecto del hipertexto partiendo de los modelos de nivel PIM propuesto por SOD-M.

La Figura 3.16 muestra el proceso para el modelado del hipertexto de MIDAS. En ella se representan, además de los todos modelos necesarios para el desarrollo del hipertexto, los conceptos de modelado que se trasladan de un modelo a otro (representados en los textos asociados a las flechas). En este caso, de la misma manera que se presentó en la Figura 3.13 para el proceso propuesto por SOD-M, se distinguen los dos tipos de transformaciones propuestas entre modelos: las guías de transformación, representadas a través de las flechas verdes en la Figura 3.16; y, las reglas de transformación, representadas a través de las flechas rojas en la Figura 3.16.

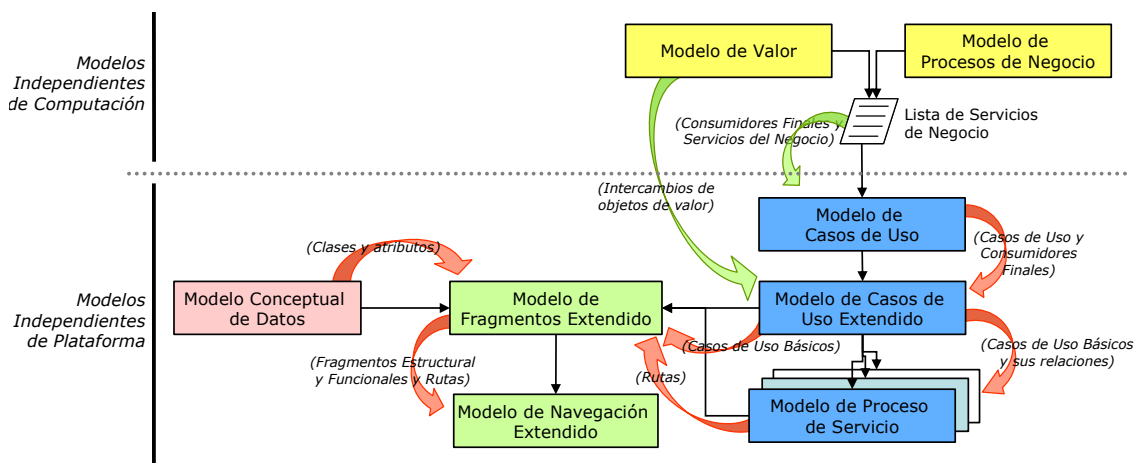


Figura 3.16. Proceso para el Modelado del Hipertexto de MIDAS

De la misma manera que en el caso de SOD-M, el proceso para el modelado del hipertexto de MIDAS propone la generación de un conjunto de modelos de nivel PIM, así como diferentes reglas de transformación entre los mismos.

Dado que las actividades de generación de los modelos de comportamiento ya se han descrito en el apartado 3.6.1, en este apartado sólo se presentan las actividades de generación de los dos modelos propuestos por MIDAS para el aspecto del hipertexto: modelo de fragmentos extendido y modelo de navegación extendido. Tales actividades se describen utilizando la misma estructura utilizada para describir el proceso de SOD-M en el apartado 3.6.1.

Generación del Modelo de Fragmentos Extendido

- **Artefactos de Entrada:** modelo de casos de uso extendido, modelo de datos conceptual y modelo de proceso de servicio.
- **Artefactos de Salida:** modelo de fragmentos extendido.
- **Tareas:**
 - *Identificar, en el modelo de casos de uso extendido, casos de uso básicos funcionales o estructurales.* Un caso de uso básico es estructural cuando el resultado observable para el actor que lo realiza es proporcionar una vista de los datos que conforman el SIW; por ejemplo, "ver vuelos disponibles". Un caso de uso básico es funcional cuando implica una interacción con el actor que lo realiza, generalmente cuando se requiere algún dato de entrada; por ejemplo, "búsqueda de vuelos" o "registro de clientes"
 - *Transformar directamente cada caso de uso básico estructural en un fragmento estructural.*
 - *Transformar directamente cada caso de uso básico funcional en un fragmento funcional.*
 - *Asignar a cada fragmento identificado los correspondientes atributos.* Esto se realiza utilizando los atributos procedentes de las clases identificadas en el modelo de datos conceptual, y asignándolos a cada fragmento estructural o funcional según los datos que sean necesarios mostrar o introducir en cada uno de ellos.
 - *Asignar a cada modelo de proceso de servicio un nombre de ruta para identificar la ruta del servicio de negocio que este representa.*

Cada bifurcación existente en el modelo dará lugar a una subruta de la ruta y habrá de ser igualmente nombrada.

- *Enlazar en el modelo de fragmentos extendido, los fragmentos asociados a cada servicio de negocio, teniendo en cuenta la ruta identificada en el modelo de proceso de servicio.*
- *Repetir este proceso para cada uno de los servicios de negocio.*

Generación del Modelo Navegación Extendido

- **Artefactos de Entrada:** modelo de fragmentos extendido.
- **Artefactos de Salida:** modelo de navegación extendido.
- **Tareas:**
 - *Identificar y agregar a los elementos ya identificados en el modelo de fragmentos extendido, las estructuras de navegación tales como índices, menú, etc.*

En este caso, también se han representado las reglas de transformación entre los modelos de SOD-M y los modelos del aspecto de hipertexto. Tales reglas se presentan primero en lenguaje natural y posteriormente en gramática de grafos, de la misma manera que se realizó con SOD-M en el apartado 3.6.2.

En la Tabla 3.11 se recoge el conjunto de reglas de transformaciones entre los modelos de SOD-M (modelos de casos de uso extendido y modelo de proceso de servicio) y el modelo de datos conceptual, y, el modelo de fragmentos extendido; así como también, las reglas entre este último y el modelo de navegación extendido. Dicha tabla recoge también en este caso su grado de automatización: completo (C) o parcial (P).

Tabla 3.11. Reglas de Transformación entre los modelo de SOD-M y modelos del hipertexto

Modelo Origen	Modelo Destino	Reglas de Transformación	Grado de Automatización
Modelo de Casos de Uso Extendido, Modelo de Proceso de Servicio, y Modelo de Datos Conceptual	Modelo de Fragmentos Extendido	1. Cada Caso de Uso Básico Estructural del modelo de casos de uso extendido se transformará en un Fragmento Estructural en el modelo de fragmentos extendido.	C
		2. Cada Caso de Uso Básico Funcional del modelo de casos de uso extendido se transformará en un Fragmento Funcional en el modelo de fragmentos extendido.	C
		3. Los atributos de cada Fragmento del modelo de fragmentos extendido serán obtenidos directamente de las respectivas clases del modelo de datos conceptual.	C

Modelo Origen	Modelo Destino	Reglas de Transformación	Grado de Automatización
		4. Cada modelo de proceso de servicio será representado como una ruta en el modelo de fragmentos extendido con el nombre del servicio de negocio que representa.	P
		5. Cada bifurcación existente en cualquiera de los modelo de proceso de servicio se representará como una subruta en el modelo de fragmentos extendido. El nombre completo de dicha subruta será el nombre del servicio de negocio (SN) más un nombre nuevo para identificar de forma única a la subruta. Es decir, SN.nombre.IDSubroute.	P
Modelo de Fragmentos Extendido	Modelo de Navegación Extendido	6. El modelo de navegación extendido debe incluir un menú inicial con una entrada para cada ruta identificada en el modelo de fragmentos extendido.	C
		7. Cada bifurcación representada en el modelo de fragmentos extendido, será precedida por un índice representado en el modelo de navegación extendido.	C
		8. Cada asociación que en el modelo de fragmentos extendido tenga como destino un Fragmento Estructural, será transformado a un <i>index</i> o <i>query</i> en el modelo de navegación extendido. El diseñador será quien tome la decisión en cada caso.	P

Las siguientes figuras, Figura 3.17 y Figura 3.18, muestran las reglas de transformación basadas en grafos para todas las transformaciones completamente automatizables que son necesarias para la generación del modelo de fragmentos extendido. La Figura 3.17 muestra las reglas 1 y 2 de la Tabla 3.11, mientras que la Figura 3.18 muestra la regla 3.

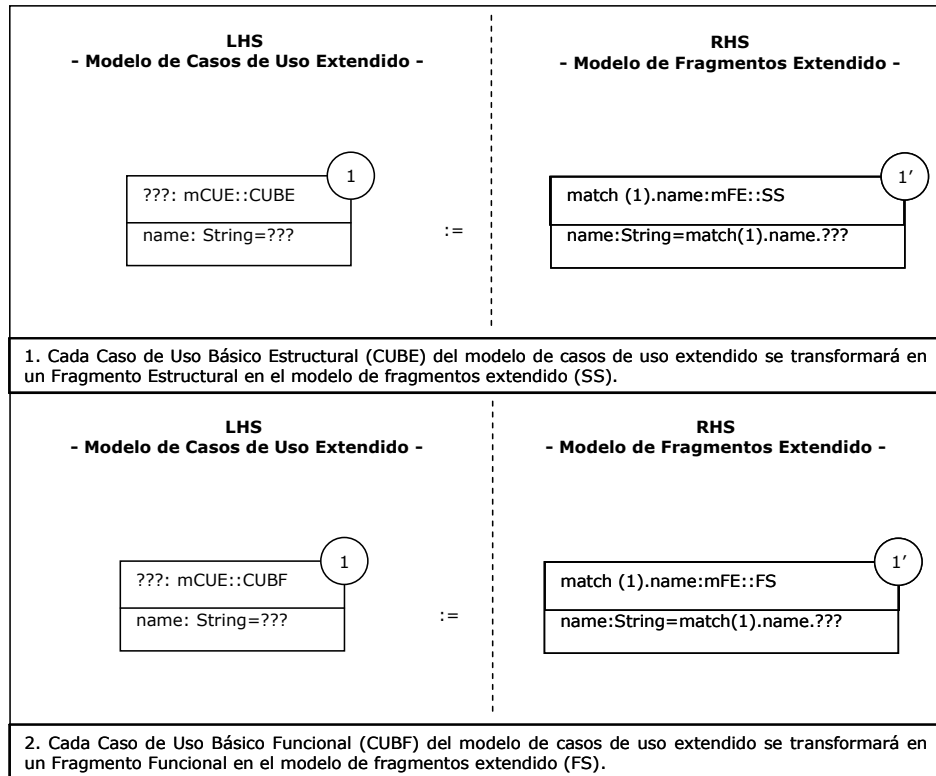


Figura 3.17. Transformación de los *Casos de Uso Básicos* del metamodelo de Casos de Uso Extendido al metamodelo de Fragmentos Extendido

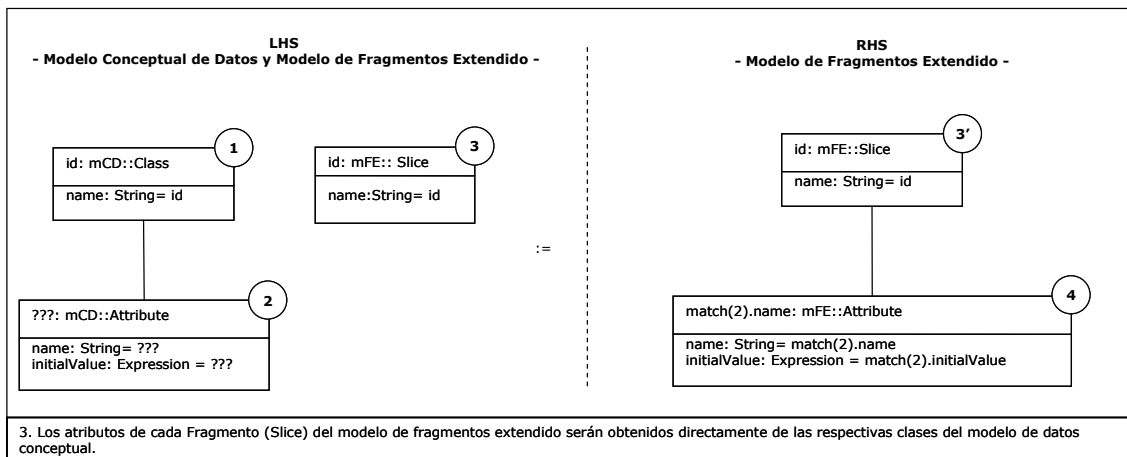


Figura 3.18. Transformación de los atributos de las clases del metamodelo de Datos Conceptual al metamodelo de Fragmentos Extendido

VALIDACIÓN

En el presente capítulo se presenta la validación del trabajo de investigación realizado en esta Tesis Doctoral, que se ha abordado aplicando el método de investigación en acción descrito en la sección 1.4 del capítulo 1. Este capítulo se estructura de la siguiente forma: en la sección 4.1 se indica el proceso seguido para la realización de la validación de este trabajo de investigación, indicando los casos de estudio, de laboratorio y reales, que se han desarrollado; posteriormente, en la sección 4.2, se describen los distintos sistemas que se han desarrollado como casos de estudio; para finalizar, en la sección 4.3 se presenta un caso de estudio completo que permite ofrecer una visión global del método de desarrollo presentado en esta Tesis Doctoral. El resto de casos de estudio se presentan en el apéndice B de esta memoria.

4.1 Método de Validación

El método de investigación utilizado en este trabajo de Tesis Doctoral, que ha sido presentado en el capítulo 1, es el resultante de la combinación de un método propuesto en (Marcos y Marcos, 1998) para investigación en el campo de la Ingeniería del Software, y de Investigación en Acción (*Action Research*) propuesto en (Avison *et al.*, 1999).

En el método de Investigación en Acción, la validación se realiza junto con la etapa de resolución, como parte de un proceso iterativo y cíclico de realimentación, lo que permite ir resolviendo el problema al mismo tiempo que se va validando. El proceso de investigación definido en el método de investigación en acción no es por tanto un proceso lineal, sino que va avanzando mediante la realización de ciclos, llamados ciclos de investigación en acción, proporcionando así la oportunidad de ir fortaleciendo los resultados y el aprendizaje obtenido de cada ciclo de la investigación en acción. Este fortalecimiento de los resultados puede obtenerse, también, expandiendo el alcance de la investigación; por ejemplo, aplicando la investigación a diferentes campos relacionados (Kock *et al.*, 1997), aumentado, de este modo, la generalidad de los resultados obtenidos en cada ciclo de la Investigación en Acción.

En este trabajo de Tesis Doctoral, el proceso de validación se ha ido realizando de forma continua y progresiva junto con la etapa de resolución. El alcance de la investigación propuesta se ha ido expandiendo, aplicando las técnicas y versiones del método al desarrollo de diversos proyectos software pertenecientes

a diferentes dominios de aplicación. Así, cada ciclo de investigación en acción se ha aplicado iterativamente sobre diferentes casos de estudio que se han propuesto, siendo dos de ellos de laboratorio y cuatro de ellos casos de estudio reales (ver Tabla 1.1 de esta memoria).

La Figura 4.1 muestra, de forma completa, el proceso de validación de este trabajo de investigación.

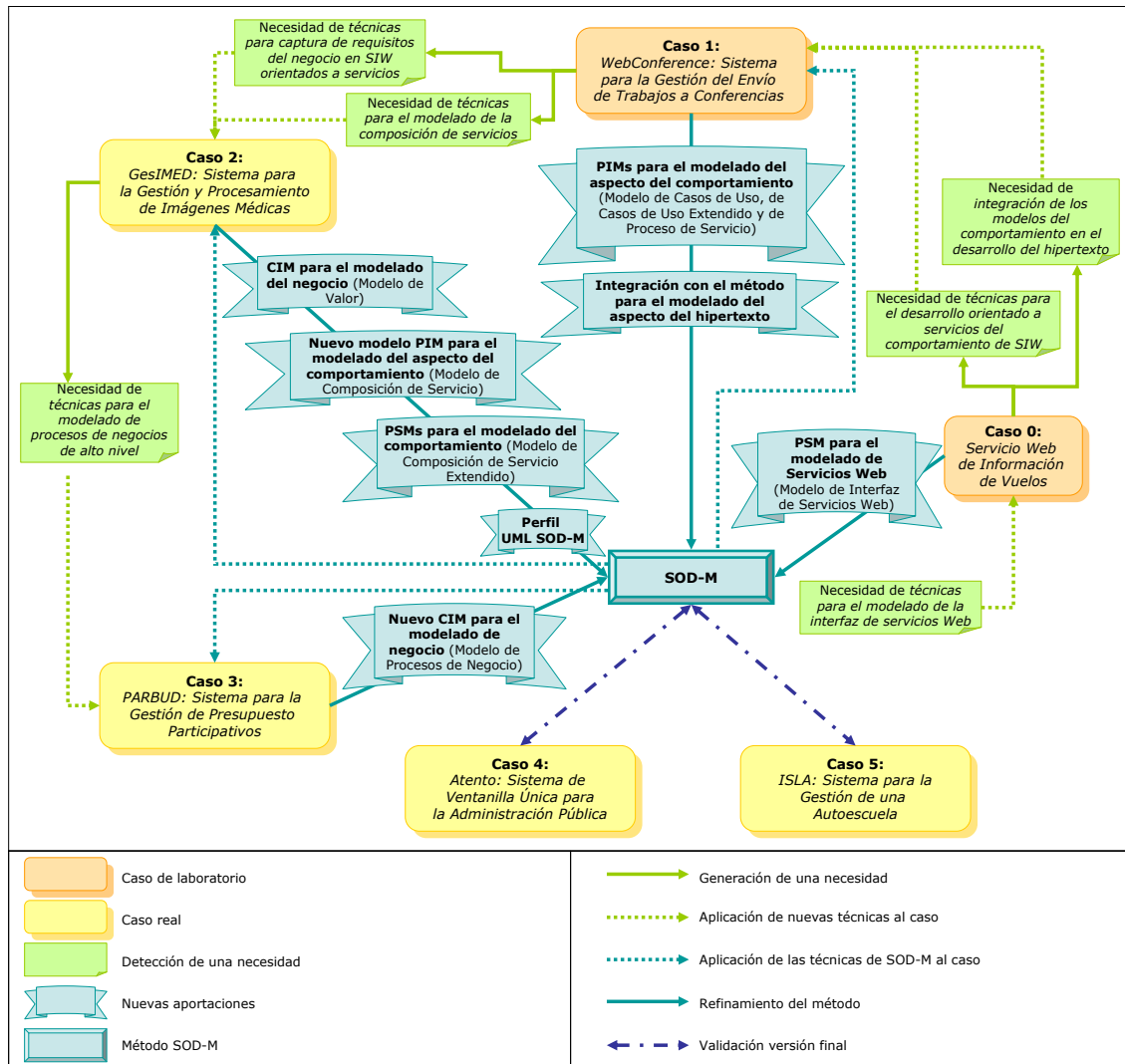


Figura 4.1. Proceso de Validación de este trabajo de Tesis Doctoral

Como puede verse en la Figura 4.1, en primer lugar se desarrollaron dos casos de estudio de laboratorio. A través de estos casos, que han permitido la validación de algunas de las técnicas propuestas, se han ido detectando las primeras necesidades del método que se pretendía definir, dando así lugar a las primeras propuestas de solución. Posteriormente, dichas propuestas se han ido refinando y validando a través de la realización de nuevos ciclos de investigación en

acción sobre los distintos casos de estudio reales contemplados. De este modo, gracias al desarrollo de dichos casos y al continuo análisis de necesidades, se han ido integrando las diferentes propuestas que han dado forma a este trabajo de Tesis Doctoral. Para finalizar el proceso de validación, se utilizaron dos casos de estudios reales que han sido desarrollados aplicando la metodología completa, permitiendo de este modo la validación de la propuesta completa, y no solo de ciertos aspectos de la misma.

A continuación se presentan cada uno de los casos de estudio desarrollados, indicando tanto las necesidades detectadas, como las nuevas aportaciones que fueron surgiendo a partir de ellos, las cuales se muestran también de forma gráfica en la Figura 4.1. Los casos de estudios en sí mismos, es decir, el servicio Web y los SIW desarrollados, se describen detalladamente en la sección 4.2 de este capítulo.

Caso de Estudio de Laboratorio

- **Caso 0: Servicio Web de Información de Vuelos.** Este caso se ha diseñado específicamente para la validación de los artefactos y estereotipos definidos para el modelado de la interfaz de servicios Web. Como resultado de este caso se ha realizado la siguiente propuesta:
 - El *modelo UML para la representación de servicios Web a nivel PSM*, denominado modelo de interfaz de servicio Web (Marcos *et al.*, 2003).

A partir de la realización de este caso, han surgido nuevas necesidades relacionadas con el desarrollo orientado a servicios de SIW. En concreto, se detectó la necesidad de *técnicas que permitan modelar el comportamiento de SIW desde un enfoque orientado a servicios*, de modo que permitan construir aplicaciones en base a servicios. Además, se ha detectado la necesidad de *contemplar los modelos del aspecto del comportamiento en el desarrollo del hipertexto de los SIW*, dado que tradicionalmente sólo se contemplaba el aspecto del contenido en el desarrollo del mismo.

- **Caso 1: WebConference: Sistema para la Gestión del Envío de Trabajos a Conferencia.** Este caso se ha diseñado para validar una primera versión de los modelos propuestos a nivel PIM para el modelado del aspecto del comportamiento, así como la integración de tales modelos en el desarrollo del hipertexto. Dicha aplicación se ha desarrollado siguiendo el método para el modelado del hipertexto de MIDAS. A través

de este caso se han validado los modelos propuestos, demostrando, además, que a partir de tales modelos, es decir, mediante la utilización de un enfoque orientado a servicios para el modelado y desarrollo de SIW, es posible obtener sistemas que faciliten la navegación de los usuarios a través del mismo (De Castro *et al.*, 2006b). Así, como resultado de este caso, se realizaron las siguientes aportaciones:

- Se propusieron los primeros *modelos para el desarrollo del aspecto del comportamiento de SIW a nivel PIM* de la arquitectura de MIDAS. En concreto, se propusieron primeras versiones de los modelos de casos de uso, de casos de uso extendido y de proceso de servicio (De Castro *et al.*, 2004).
- Se propusieron las guías para la *integración*, a través de la transformación entre modelos, *con el método propuesto para el modelado del aspecto del hipertexto* en MIDAS (Cáceres *et al.*, 2004; Cáceres *et al.*, 2004b).

Casos de Estudio Reales

- **Caso 2: GesIMED: Sistema para la Gestión y Procesamiento de Imágenes Médicas.** Para el desarrollo de este caso (Acuña *et al.*, 2004, Hernández *et al.*, 2006), se aplicaron las técnicas propuestas por SOD-M para el modelado del aspecto del comportamiento. El objetivo que se buscó a través de la aplicación del método a este caso fue probar la utilidad de los modelos propuestos en un entorno diferente a los SIW típicos de gestión, además de, aplicar el método a un caso de desarrollo real, dónde tanto los usuarios del sistema como los desarrolladores no coincidan con la propia doctoranda.

Para la realización de este caso se detectó la *necesidad de utilizar técnicas específicas para la captura de requisitos de los servicios del negocio en SIW*. Dado que hasta el momento no se contaba en SOD-M con una técnica para la captura de requisitos que permitiera identificar tanto los servicios del negocio, como los miembros que colaboran en cada uno de ellos. Además, también se detectó la necesidad de definir *técnicas específicas para el modelado de la composición de servicios*, de modo que permitiera luego facilitar la implementación de la composición a través de servicios Web.

Como resultado de la realización de este caso de estudio, y a través de un proceso de identificación, definición, validación y refinamiento de técnicas propuestas sobre el mismo caso, se realizaron las siguientes propuestas:

- Se propuso la utilización del método e³value para el *modelado de negocio a nivel CIM* de la arquitectura de MIDAS.
- Se definieron nuevos *modelos para el desarrollo del aspecto del comportamiento de los SIW* a nivel PIM de la arquitectura de MIDAS. En concreto, se propuso el modelo de composición de servicio, a la vez que se refinaron los modelos definidos previamente (De Castro et al., 2006a).
- Se definió un *modelo de nivel PSM*, llamado de composición de servicio extendido.
- Se definió además un *Perfil UML* agrupando todos los elementos para el modelado del comportamiento de un SIW desde un enfoque orientado a servicios.

- **Caso 3: PARBUD: Sistema para la Gestión de Presupuestos Participativos.** Este caso se llevó a cabo aplicando las técnicas y modelos propuestos al desarrollo del sistema PARBUD. Este caso se desarrolló también con el objetivo de detectar posibles mejoras o nuevas necesidades de un dominio particular como es el desarrollo de aplicaciones de democracia electrónica. Además, también en este caso se trató con usuarios externos, con los que se trabajó de manera conjunta para el desarrollo de la aplicación.

Para la realización de este caso se detectó que, además del modelo de negocio propuesto por el método e³value, *era necesario contar con un modelo de negocio que facilite al diseñador de un SIW la comprensión de los procesos de negocios de la organización en la que se desenvuelve el sistema que se desea desarrollar*. Es necesidad surgió a partir de que, en este caso concreto, se notó que todos los servicios que ofrecía la aplicación a los usuarios eran muy dependientes unos de otros, y existía un orden de precedencia entre cada uno de ellos; así, por ejemplo, había servicios que no se podían llevar a cabo sin que otros se realicen primero.

Como resultado final del desarrollo de este caso de estudio, y también, a través de un proceso definición, validación y refinamiento, se realizó la siguiente propuesta:

- Una *nueva técnica para el modelado de proceso de negocio a nivel CIM* de la arquitectura de MIDAS (Cáceres *et al.*, 2006b).

- **Caso 4: Atento: Sistema de Ventanilla Única para la Administración Pública.** Este caso se ha realizado aplicando el método completo sobre un sistema real para la gestión de la administración pública. En este caso se analizaron, en particular, las distintas posibilidades de implementación de la composición de servicios a partir de los modelos propuestos en SOD-M.

A través de la realización de este caso, se han detectado algunas pequeñas inconsistencias en los modelos propuestos que se han corregido oportunamente. Además, se han determinado las posibles formas de llevar a cabo la implementación la composición de servicios a partir de los modelos definidos (López *et al.*, 2006a; López *et al.*, 2006b).

- **Caso 5: ISLA: Sistema para la Gestión de una Autoescuela.** Este último caso se llevó a cabo aplicando también el método completo sobre un caso de estudio real.

Este caso fue especialmente útil para la validación de las transformaciones entre los distintos modelos propuestos a nivel CIM, PIM y PSM, las cuales se fueron definiendo en lenguaje natural a medida que se proponían los diversos modelos.

En la siguiente sección se describen brevemente el servicio Web y los distintos SIW que se han desarrollado como casos de estudio para la validación de este trabajo de investigación. En la sección 4.3 de este capítulo se presenta, a modo de ejemplo, el desarrollo completo de caso de estudio 2, GesIMED: Sistema para la Gestión y Procesamiento de Imágenes Médicas. El desarrollo completo del resto de los casos de estudios realizados se presenta en el apéndice B de esta memoria.

4.2 Descripción de los Casos de Estudio desarrollados

4.2.1 Casos de Estudio de Laboratorio

Caso 0: Servicio Web de Información de Vuelos

En este caso, se aborda el modelado y desarrollo de una interfaz de un servicio Web para proporcionar información acerca de los vuelos de un aeropuerto, tales como fecha y hora de salida de un vuelo determinado. El servicio Web objeto del caso de estudio, consta de dos operaciones: una que

proporciona información acerca de los vuelos de un aeropuerto y otra que permite llevar a cabo el *checking* de un vuelo.

Este caso se desarrolló analizando en primer lugar las características de las interfaces de los servicios Web, a través del estándar WSDL, y definiendo posteriormente todos los estereotipos, valores etiquetados y restricciones necesarias para la representación, con UML, del servicio Web tomado como caso de estudio.

Caso 1: WebConference: Sistema para la Gestión del Envío de Trabajos a Conferencias

 <http://kybele.escet.urjc.es/webconference/>

En este caso se aborda el diseño y la implementación de un SIW para la gestión y envío de trabajos a conferencias. El SIW ofrece tres servicios específicos a los usuarios, en este caso, los autores:

- *Envío de artículo:* para llevar a cabo este servicio es necesario indicar los datos del artículo a enviar (título, resumen, autores, etc.) y, posteriormente, realizar el envío del trabajo a través de la aplicación.
- *Edición de los datos de autor:* en este caso el autor puede visualizar sus datos y realizar las modificaciones que desee.
- *Visualización de artículos enviados por un autor:* contempla la visualización, por parte del autor, de los diferentes artículos enviados a una conferencia, así como la posibilidad de actualización de los datos de éstos, tales como título, resumen, autores, etc.

Para llevar a cabo estas tareas, un autor debe siempre identificarse, indicando nombre de usuario y contraseña. Además, el sistema ofrece la posibilidad de consultar su contraseña en caso de olvido.

Este SIW se desarrolló utilizando una primera aproximación de los modelos propuestos por SOD-M para el modelado del comportamiento a nivel PIM y aplicando el método del modelado del hipertexto de MIDAS para el desarrollo de hipertexto de la aplicación Web.

4.2.2 Casos de Estudio Reales

Caso 2: GesIMED: Sistema para la Gestión y Procesamiento de Imágenes Médicas

 <http://ariadna.escet.urjc.es/gesimed/>

Este caso aborda el desarrollo de un SIW que permite almacenar y gestionar imágenes médicas, así como el procesamiento de las mismas. Este SIW se ha desarrollado en conjunción con el grupo GTEBIM de la Universidad Rey Juan Carlos.

El SIW ha sido diseñado para ser utilizado, principalmente, por investigadores en neurociencias, a los que el sistema le ofrece tres servicios:

- *Almacenamiento y consulta de imágenes médicas:* ofrece a los investigadores en neurociencias la posibilidad de realizar diferentes tipos de consultas sobre las imágenes como, por ejemplo, por estudio, por modalidad de obtención de las imágenes, por patología, etc.
- *Procesamiento de imágenes médicas:* ofrece a los investigadores en neurociencias la posibilidad de realizar diversos tipos de procesos sobre imágenes médicas, pudiendo proporcionar para ello sus propias imágenes, o bien, acceder a las imágenes almacenadas en la base de datos.
- *Visualizaciones de imágenes médicas:* ofrece a los investigadores en neurociencias la posibilidad de realizar procesos de la visualización sobre las imágenes médicas. Para la realización de este servicio, los investigadores en neurociencias pueden también proporcionar sus propias imágenes, o bien acceder a las imágenes almacenadas en la base de datos.

Este SIW se desarrolló en el marco de MIDAS, aplicando los diversos modelos propuestos por SOD-M para el desarrollo orientado a servicio de los SIW. Este caso se presenta de forma completa en sección 4.3 de esta memoria.

Caso 3: PARBUD: Sistema para la Gestión de Presupuestos Participativos

 [Enlace pendiente](#)

Este caso aborda el desarrollo de un SIW para la gestión de presupuestos participativos. Este SIW se está desarrollando actualmente en colaboración con el Grupo de Estadística de la Universidad Rey Juan Carlos.

La idea de presupuestos participativos consiste en permitir a los ciudadanos participar en la decisión de asignación de un presupuesto. Así, por ejemplo, los miembros de un barrio podrían decidir cómo y en qué invertir parte del presupuesto que el ayuntamiento tiene para el desarrollo del barrio.

Para el diseño de esta aplicación se hizo necesaria la definición del modelo de proceso de negocio, puesto que permitía comprender y determinar explícitamente qué pasos o etapas sería necesario seguir para completar el ciclo de un presupuesto participativo. El proceso para la elaboración de un presupuesto participativo incluye las siguientes etapas: *(a) preparación*, en la que se establecen cuestiones como dinero que se asignará, quiénes decidirán, alternativas de asignación, etc.; *(b) comunicación de preferencias*, en la que los participantes indican sus preferencias de asignación del presupuesto; *(c) negociación*, que se realiza en caso de no haber una propuesta de asignación que satisfaga a los distintos participantes y que se lleva a cabo a través un método de negociación determinado por el mediador; *(d) votación*, que se realiza en caso de no llegar a un acuerdo a través de la negociación, y en la que cada participante vota por su preferencia; y, finalmente, se incluye una etapa de *(e) pos-acuerdo*, en la que se analiza en qué medida la asignación de presupuesto acordada satisface al conjunto de participantes.

En una primera etapa del diseño de este SIW, se ha abordado la etapa comunicación de preferencias, en la se contemplan dos tipos de usuarios: participante y mediador. Los servicios ofrecidos por el SIW a dichos usuarios son:

- *Solicitud de participación en presupuestos*: ofrece a los participantes la posibilidad de participar en la asignación de un determinado presupuesto, debiendo introducir para ello, sus datos personales y de contacto.
- *Autorización de la participación en presupuestos*: ofrece a los mediadores la posibilidad de gestionar los distintos miembros que participan en una asignación de presupuesto, autorizando o no el acceso a los mismos.
- *Comunicación de preferencias*: ofrece a los participantes la posibilidad de comunicar sus preferencias entre las distintas alternativas de asignación de un presupuesto.
- *Visualizar presupuestos de un participante*: ofrece a los participantes la posibilidad de consultar los presupuestos en los que esté participando.

Este SIW se ha modelado siguiendo el método SOD-M para el desarrollo orientado a servicios de SIW y en la actualidad está siendo implementado. Cabe mencionar, además, que para la realización de este SIW se partió de una aplicación Web existente, llamada PARBUD (<http://parbud.escet.urjc.es/>), sobre la se han ido detectando los diversos requisitos del sistema.

Caso 4: Atento: Sistema de Ventanilla Única para la Administración Pública

 [Enlace pendiente](#)

En este caso se aborda el desarrollo del SIW Atento, que es un sistema para la gestión de las relaciones del ciudadano con la administración pública. El SIW Atento permite acceder, a través de la intranet de la administración, a la información sobre las peticiones, y sus correspondientes respuestas, que se han producido entre un ciudadano y la administración, sea cual sea su canal de entrada (telefónico, presencial, Web, e-mail) o el área de la administración que la haya atendido. Así, el SIW ofrece una visión de conjunto de todas las relaciones mantenidas entre la administración pública y los ciudadanos, mejorando de esta manera el nivel de atención que se presta al mismo. Un ejemplo de relación entre el ciudadano y la administración es la solicitud y emisión de un certificado oficial o la consulta sobre los trámites a seguir para obtener una determinada prestación de la administración. Esta información sobre relaciones entre ciudadanos y administración se encuentra almacenada en el “sistema de gestión de las relaciones con el ciudadano” que es externo a Atento y que, además de gestionar estas relaciones, puede acceder a información útil durante el proceso de atención que se encuentra almacenada en otras bases de datos corporativas. Este otro sistema es utilizado por los Agentes de Atención Telefónica. Atento tiene una funcionalidad más limitada que el “sistema de gestión de las relaciones con el ciudadano”.

Los usuarios del sistema Atento son los agentes de atención presencial (empleado público) los cuales pueden acceder, a través de la intranet de la entidad pública, a los siguientes servicios:

- *Editar datos de un ciudadano:* ofrece a los usuarios de Atento la posibilidad de editar cualquiera de los datos de un ciudadano, como por ejemplo, realizar un cambio de domicilio. Para ello, el usuario deberá en primer lugar buscar al ciudadano en el sistema y, una vez seleccionado, podrá modificar los datos de contacto del ciudadano.

- *Grabar interacción de un ciudadano:* para ello, el usuario también deberá buscar al ciudadano en el sistema, después podrá indicar que se inicia la interacción y el sistema se encargará de crear una nueva interacción asociada a ese ciudadano, indicando datos como el código del usuario que lo está atendiendo, el estado inicial, la fecha de apertura, etc. Una interacción de un ciudadano podría ser, por ejemplo, una solicitud de un informe de vida laboral.
- *Consultar interacciones de ciudadanos:* ofrece a los usuarios la posibilidad de consultar los datos de una interacción existente. Para ello, el usuario deberá buscar la interacción existente aportando alguno de los datos que la identifican tales como, número de la interacción, nombre del ciudadano al que pertenece, fecha de apertura, etc.
- *Grabar encuesta de un ciudadano:* ofrece la posibilidad de realizar una encuesta de satisfacción al ciudadano para preguntarle sobre el proceso de atención. Para ello, el usuario deberá localizar al ciudadano en la base de datos para después grabar una nueva encuesta asociada a el.

Este SIW se ha modelado también aplicando el método SOD-M completo y en la actualidad está siendo implementado.

Caso 5: ISLA: Sistema para la Gestión de una Autoescuela

 Prototipo disponible en <http://isla-pfc.selfip.org>

En este caso se aborda el desarrollo de un SIW que contempla todas las actividades de gestión relacionadas con los servicios que proporciona una Autoescuela.

El negocio de la autoescuela involucra a distintos miembros que interaccionan con la autoescuela: *los alumnos*, que son los consumidores de los servicios de la autoescuela; *el profesor*, encargado de la impartición de las clases prácticas y teóricas a los alumnos; *la asociación de autoescuelas*, que aporta información fundamental para el desarrollo de la autoescuela, tales como nuevas disposiciones; *las empresa de distribución de material*, encargadas de proporcionar el material necesario para el funcionamiento de la autoescuela; y la *Dirección General de Tráfico (DGT)*, el organismo público que se encarga de conceder los carnés de conducir a los alumnos que lo solicitan, después de realizar un examen y ser declarados aptos.

Los servicios que la autoescuela ofrece a los alumnos son los siguientes:

- *Obtención del carné de conducir:* este es el principal servicio que da la autoescuela, y proporciona a los alumnos la posibilidad de obtener el carné de conducir. Para ello, el alumno debe primero contratar los servicios de la autoescuela, adquirir el material que le proporciona la misma y, posteriormente, apuntarse a clases prácticas y teóricas. Después de haber aprobado los exámenes teórico y práctico en la DGT, el alumno podrá solicitar la expedición del carné y la autoescuela dará por finalizado el contrato con el mismo.
- *Solicitud de clases prácticas de conducción:* proporciona al alumno la posibilidad de apuntarse a clases prácticas de conducción, por ejemplo, si lleva muchos años sin conducir, o bien para una convalidación de carné extranjero. Para ello, el alumno deberá también formalizar un contrato con la autoescuela y, posteriormente, apuntarse a las clases prácticas, seleccionando horas disponibles y realizando el pago de las mismas.
- *Solicitud de compra de material:* proporciona al alumno a posibilidad de adquirir el material que proporciona la autoescuela, tales como libros de teoría, libros de test prácticos, placas de conductor novel (L), etc. Para ello, el alumno deberá seleccionar el material que desea adquirir y realizar el pago del mismo.
- *Recuperación del carné de conducir:* ofrece al alumno la posibilidad de asistir a un curso teórico para recuperar el carné de conducir que le ha sido retirado por una autoridad competente, por ejemplo, por haber agotado los puntos del mismo. Para recuperar el carné, el alumno debe apuntarse y asistir a todas las clases de un curso de reciclaje impartido por la autoescuela, por lo que se deberá llevar un control de asistencia.

Para el desarrollo de este SIW se ha aplicado el método SOD-M completo y en la actualidad está siendo implementado.

4.3 Aplicación a un Caso de Estudio Real

En esta sección se presenta, de manera detallada, uno de los casos de estudio reales realizados para la validación del método propuesto en este trabajo de Tesis Doctoral, el SIW GesIMED.

El SIW GesIMED se ha desarrollado utilizando los modelos propuestos por SOD-M tanto a nivel CIM, como a nivel PIM y PSM. Cabe destacar que este caso de estudio se ha desarrollado previo a la definición del modelo de proceso de negocio

de nivel CIM; sin embargo, con el objetivo de mostrar el conjunto completo de los modelos propuestos, dicho modelo se ha representado también para este caso y se incluye dentro de esta sección.

A continuación se presentan todos los modelos desarrollados para este caso y posteriormente se describen los principales aspectos de la implementación del mismo.

4.3.1 Descripción del SIW GesIMED

El sistema GesIMED, es un SIW para la gestión y el procesamiento de imágenes médicas a través de la Web. Este SIW se ha desarrollado en colaboración con el grupo GTEBIM de la Universidad Rey Juan Carlos (GesIMED System, Hernández *et al.*, 2006). GesIMED permite gestionar la información referente a la creación y mantenimiento de estudios científicos para la investigación en neurociencias. El SIW ha sido diseñado para ser utilizado, principalmente, por investigadores en neurociencias, tales como neurólogos, neuropsicólogos, neurorradiólogos, y diversos médicos que realizan investigaciones en esa área. Además, el sistema es utilizado también por los investigadores del Laboratorio de Análisis de Imágenes Médicas de la Universidad Rey Juan Carlos (en adelante, LAIM)

El objetivo del sistema es ofrecer a los investigadores en neurociencias: una base de datos de imágenes médicas disponible a través de la Web, sobre la que sea posible realizar todo tipo de consultas; y, procedimientos normalizados que permitan a dichos investigadores realizar el procesamiento y análisis de las imágenes almacenadas, y cuyos resultados sean también almacenados en la misma base de datos, de modo que puedan ser utilizados en estudios futuros. Como puede verse en la Figura 4.2, donde se muestra el escenario de GesIMED, el LAIM ofrece a los investigadores en neurociencias tres servicios concretos:

- *Servicio de almacenamiento y consulta de imágenes médicas:* el LAIM-URJC cuenta con una base de datos de imágenes médicas y resultados de procesamiento de tales imágenes. Los investigadores en neurociencias pueden realizar diferentes consultas sobre las imágenes; pudiendo consultar por estudio, por modalidad de obtención de las imágenes, por patología, etc. Así, por ejemplo, se podría consultar imágenes de cerebros que han sido realizadas para estudios del lenguaje. Tales imágenes o estudios podrían ser consultadas con propósitos de diagnóstico, de investigación o incluso de docencia.

- *Servicio de procesamiento de imágenes médicas:* los investigadores en neurociencias pueden realizar a través de este servicio dos tipos de procesos sobre imágenes médicas: análisis y segmentación. El primero, consiste en la medición de parámetros físicos y fisiológicos en las imágenes médicas; el segundo, consiste en la segmentación de imágenes, por ejemplo, la extracción de una parte de un hueso de una imagen radiográfica. Para la realización de este servicio, los investigadores en neurociencias pueden proporcionar sus propias imágenes o bien, acceder a las imágenes almacenadas en la base de datos.
- *Servicio de visualizaciones de imágenes médicas:* este servicio ofrece a los investigadores en neurociencias la posibilidad de realizar procesos de visualización sobre las imágenes médicas. Los procesos de visualización consisten en: realizar reconstrucciones de imágenes, por ejemplo, creación de imágenes 3D a partir de varias imágenes 2D; y funciones de visualización multimodalidad, por ejemplo, componer en una imagen, una imagen anatómica y una imagen funcional. Para la realización de este servicio, investigadores en neurociencias pueden también proporcionar sus propias imágenes o bien, acceder a las imágenes almacenadas en la base de datos.

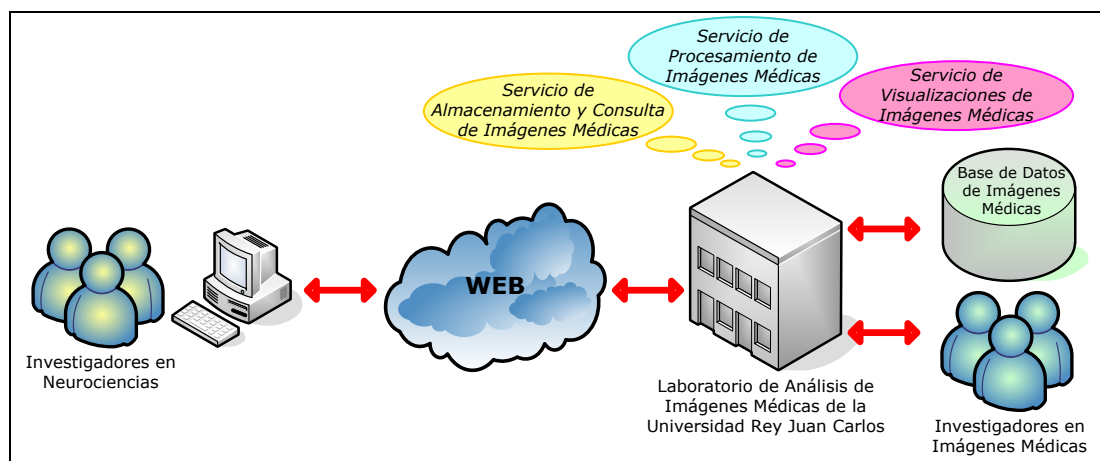


Figura 4.2. Escenario del SIW GesIMED

Para acceder a cada uno de estos servicios, los investigadores en neurociencias pagan una cierta cantidad de dinero establecido por el LAIM. Además, normalmente, son los investigadores en neurociencias quienes proporcionan al LAIM las imágenes sobre las que desean realizar los procesamientos o visualizaciones. Para obtener las imágenes médicas, los investigadores en neurociencias solicitan las mismas a diversos centros médicos, que se encargan de

la adquisición de imágenes médicas en sus diversas modalidades (tomografía por emisión de positrones, tomografía axial computada, resonancia magnética funcional, etc.).

4.3.2 CIM: Modelado de Negocio

La Figura 4.3 muestra el **modelo de valor** para el caso del SIW GesIMED. En dicho modelo se representa el “LAIM” como un actor, y tanto los “centros médicos” como los “investigadores en neurociencias” como segmentos de mercado. En este caso el investigador en neurociencias es un consumidor final del negocio. Además, en el modelo se identifican como actividades de valor los servicios que ofrece el LAIM a sus clientes, puesto que son actividades que realiza el LAIM y por las que espera obtener una rentabilidad económica. Las actividades de valor son por tanto: el “servicio de procesamiento de imágenes médicas (SPim)”, el “servicio de almacenamiento y consulta de imágenes médicas (SACim)” y “el servicio de visualizaciones de imágenes médicas (SVim)”. Los objetos de valor intercambiados entre los actores en este caso son: las “imágenes”, el “resultado del procesamiento de imágenes”, el “acceso y consulta de imágenes”, el “resultado de las visualizaciones de imágenes”, y el “dinero”. Tales objetos de valor se muestran en la Figura 4.3, en textos de color verde, unidos a la línea que indica la dirección del intercambio.

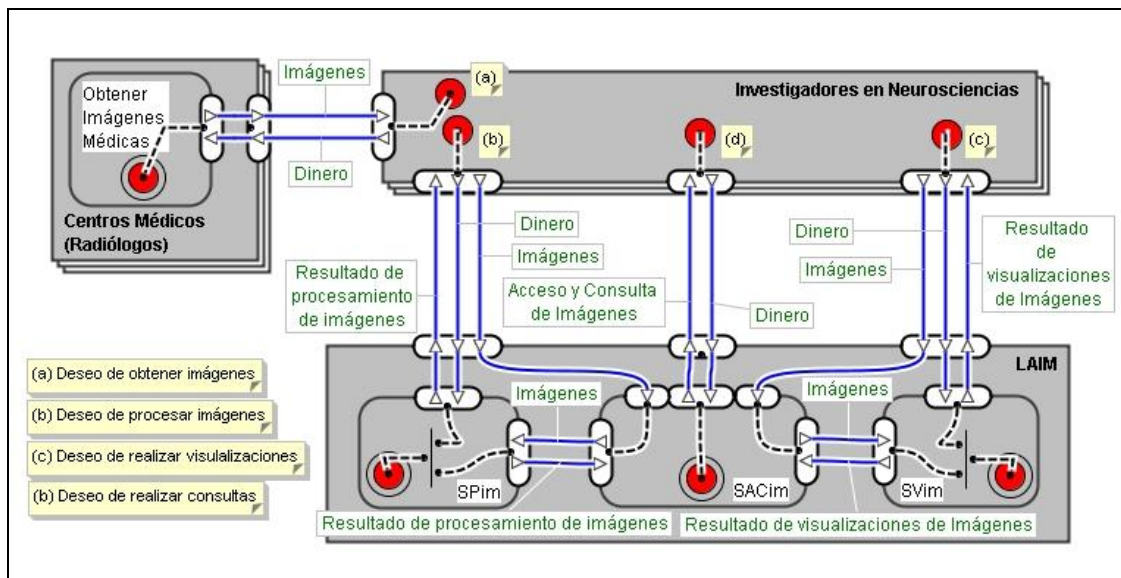


Figura 4.3. Modelo de valor para el caso GesIMED

Además, en la Figura 4.3 se muestran los caminos de dependencia. En la figura se señalan como (a), (b), (c) y (d), las distintas necesidades de los investigadores en neurociencias y que representan los estímulos de inicio de los

distintos caminos de dependencia. En primer lugar, los investigadores necesitan obtener las imágenes que les proporcionan los centros médicos, y lo hacen a cambio del pago de los mismos (camino (a) en la Figura 4.3). Una vez que las imágenes han sido adquiridas, los investigadores en neurociencias solicitan al LAIM el procesamiento o la realización de visualizaciones sobre dichas imágenes. Así, cuando se solicita un procesamiento (camino (b)), los investigadores proporcionan las imágenes y realizan el pago del procesamiento, obteniendo a cambio el resultado del mismo. Las imágenes que proporciona el investigador son recibidas por el SACim, de donde el SPim las obtiene a cambio del resultado del procesamiento de las mismas. Un camino de dependencia similar ocurre cuando un investigador en neurociencias desea obtener las visualizaciones (camino (c)). Finalmente, los investigadores en neurociencias pueden realizar todo tipo de consultas a la base de datos, el SACim es el encargado de atender esta necesidad y recibe a cambio el pago del servicio (camino (d)).

La Figura 4.4 muestra el **modelo de proceso de negocio** que se ha realizado para el caso del SIW GesIMED. En el modelo se muestra el conjunto de actividades que debe llevar a cabo cada uno de los investigadores en neurociencias para realizar sus investigaciones. Así, los investigadores deben primero obtener las imágenes, solicitando la adquisición de las mismas a diversos centros médicos que se encargados de esta actividad, o bien accediendo a bases de datos de imágenes médicas especializadas que proveen a los investigadores diversos tipos de imágenes médicas. Una vez adquirida las imágenes, los investigadores solicitan a diversos centros, como el LAIM, el procesamiento de las mismas.

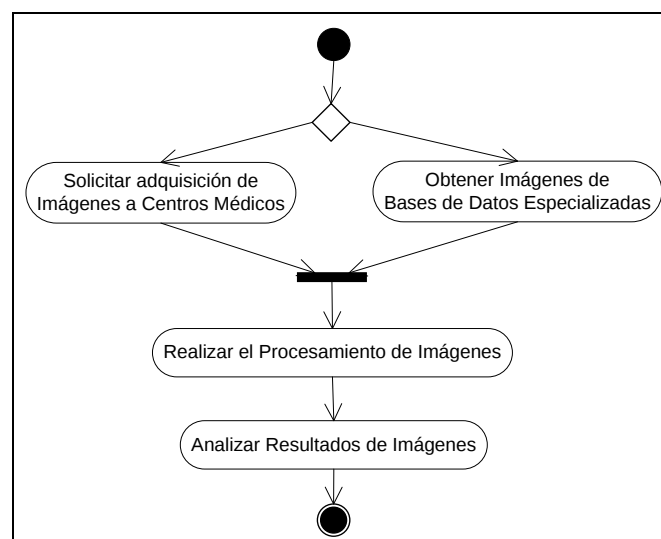


Figura 4.4. Modelo de proceso de negocio para el caso GesIMED

La **lista de los servicios de negocio**, que se muestra en la Tabla 4.1 provee una descripción de todos los servicios que forman parte de un negocio y que se ofrecen para satisfacer las necesidades del consumidor final del negocio de GesIMED. Como se ha dicho anteriormente en esta misma sección, el investigador en neurociencias es un consumidor final del negocio del sistema GesIMED. Los servicios de negocio que requiere un investigador en neurociencias y que se muestran en la Tabla 4.1 se han obtenido tanto del modelo de valor como del modelo del proceso de negocio, siguiendo las guías definidas en la sección 3.6 del capítulo 3: a partir del modelo de valor, identificando las necesidades o estímulos iniciales de los camino de dependencia (ver Figura 4.3); y a partir del modelo proceso de negocio, identificando aquellas actividades del proceso que permiten satisfacer una o varias necesidades de consumidores finales del negocio (ver Figura 4.4).

Tabla 4.1. Lista de servicios de negocio para el caso GesIMED

Consumidor Final	Servicios de Negocio
Investigador en Neurociencias	Servicio de obtención de imágenes médicas
	Servicio de procesamiento de imágenes médicas
	Servicio de visualización de imágenes médicas
	Servicio de consultas de imágenes médicas
	Servicios de análisis de imágenes médicas

Como puede verse en la Figura 4.3 y en la Figura 4.4, tales modelos no describen las características que tendrá el SIW que se desea implementar, sino que permiten describir la situación o el entorno en el que se utilizará el sistema, estableciendo, además, un vocabulario compartido que será utilizado en el resto de los modelos que se utilizarán para describir el sistema.

4.3.3 PIM: Modelado del Comportamiento

Como ya se ha explicado en el apartado 3.6.1.4, donde se describen las tareas necesarias para la generación del modelo de casos de uso, para comenzar con el modelado del sistema, será necesario identificar, en primer lugar, el alcance o límites del SIW que se pretende desarrollar. Esto implica determinar: *cuáles de los servicios del negocio identificados en la lista de servicios del negocio, serán soportados por el SIW, y, cuáles de los consumidores finales identificados en la lista de servicios del negocio, serán consumidores finales o usuarios de los servicios*

SIW. Respondiendo a estas preguntas es posible identificar los elementos de modelado que forman parte del modelo de casos de uso.

La Figura 4.5 muestra el **modelo de casos de uso** para el SIW GesIMED. En el se representa como un actor al “investigador en neurociencias”, que es el consumidor del final de los servicios que se implementarán en el SIW GesIMED. Además, en el modelo se representan como casos de uso, los distintos servicios de negocio que ofrecerá el SIW GesIMED a los investigadores en neurociencias: “realizar procesamiento de imágenes”, “realizar visualizaciones de imágenes” y realizar consultas”. Tales servicios de negocio se han identificado teniendo en cuenta aquellos servicios de negocio, de la lista de servicios del negocio (Tabla 4.1), que serán provistos por el SIW GesIMED.

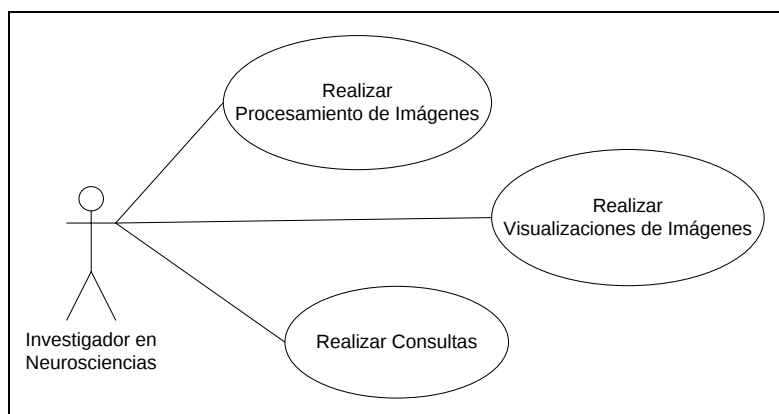


Figura 4.5. Modelo de casos de uso del SIW GesIMED

El siguiente modelo, el modelo de casos de uso extendido, se utilizará para el modelado de las funcionalidades requeridas por el sistema para llevar a cabo los servicios de negocio. En este caso se ha optado, para una mayor claridad, por presentar diferentes diagramas, uno por cada servicio de negocio identificado en la Figura 4.5. Así, la Figura 4.6 muestra el **modelo de casos de uso extendido** para el servicio de negocio “realizar procesamiento de imágenes”, la Figura 4.7 muestra el mismo modelo para el servicio de negocio “realizar visualizaciones de imágenes”, y la Figura 4.8 muestra el mismo modelo para el servicio de negocio “realizar consultas”.

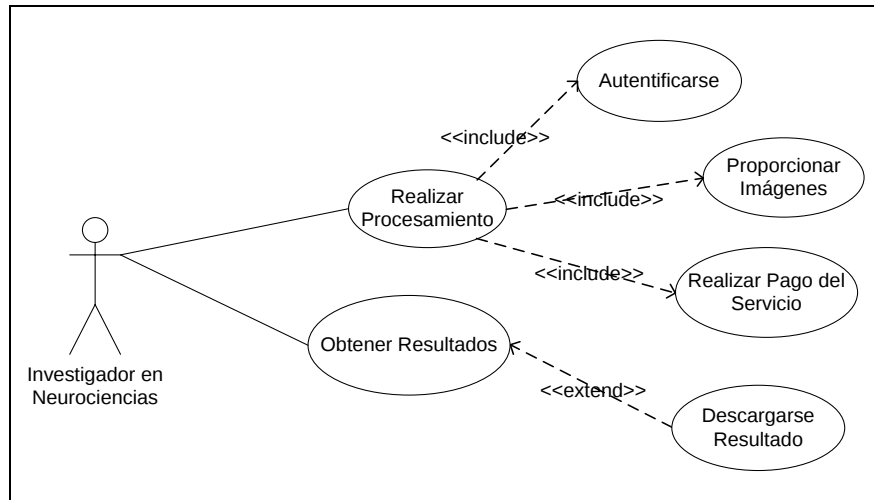


Figura 4.6. Modelo de casos de uso extendido para el servicio de negocio "realizar procesamiento de imágenes"

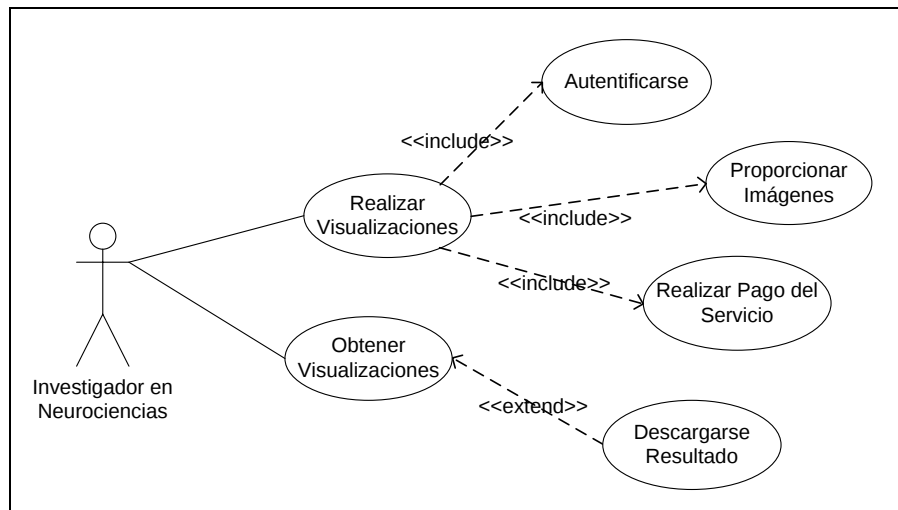


Figura 4.7. Modelo de casos de uso extendido para el servicio de negocio "realizar visualizaciones de imágenes"

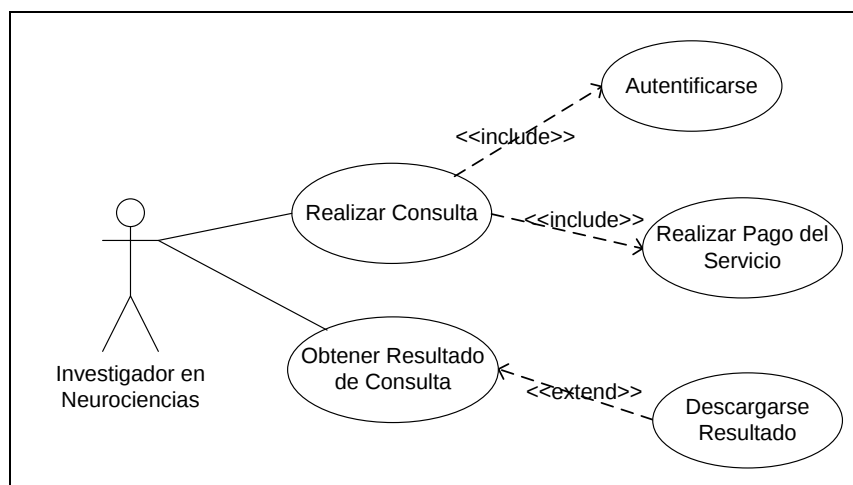


Figura 4.8. Modelo de casos de uso extendido para el servicio de negocio "realizar consultas"

Una vez identificadas las distintas funcionalidades que son requeridas por el sistema para llevar a cabo los servicios de negocio, el **modelo de proceso de servicio** se utiliza para representar el flujo de trabajo necesario para la realización de un servicio de negocio en el sistema. En este caso se representan diferentes diagramas, uno por cada uno de los servicios de negocio, los cuales se muestran en las siguientes figuras: la Figura 4.9, para el servicio de negocio “realizar procesamiento de imágenes”; la Figura 4.10, para el servicio de negocio “realizar visualizaciones de imágenes”; y la Figura 4.11 para el servicio de negocio “realizar consultas”. Las actividades de dichas figuras se han obtenido a partir de los casos de uso básicos representados en el modelo de casos de uso extendido, y los flujos se ha identificado siguiendo las reglas de transformación descritas en la Tabla 3.9.

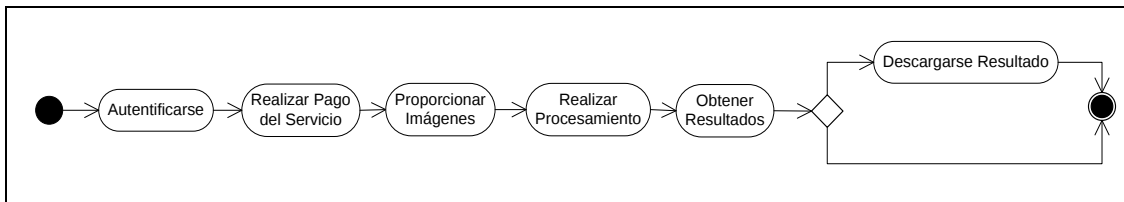


Figura 4.9. Modelo de proceso de servicio para el servicio de negocio “realizar procesamiento de imágenes”

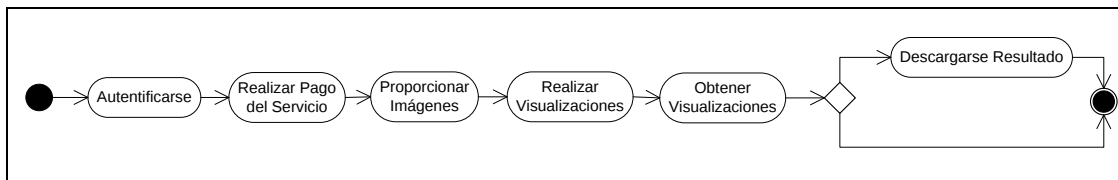


Figura 4.10. Modelo de proceso de servicio para el servicio de negocio “realizar visualizaciones de imágenes”

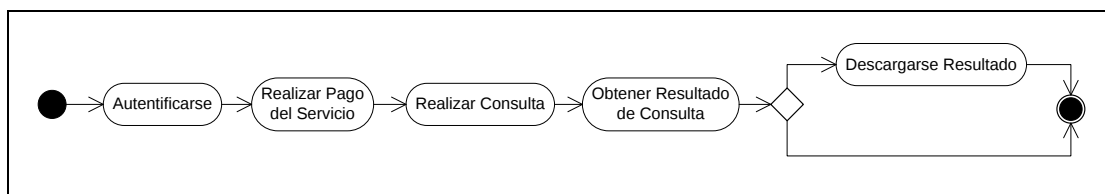


Figura 4.11. Modelo de proceso de servicio para el servicio de negocio “realizar consultas”

Las siguientes figuras: Figura 4.12, Figura 4.13 y Figura 4.14 muestran los diferentes **modelos de composición de servicio**, en los que se representan los procesos de composición de las distintas acciones que son necesarias para llevar a cabo cada servicio del negocio, indicando, además, cuáles son los miembros del negocio que realizan cada acción. Los elementos de este modelo se obtienen en base a las guías descritas en la sección 3.6. Así, por ejemplo, en la Figura 4.14, que

describe la composición para la realización del servicio “Realizar consulta”, se han identificado dos colaboradores del negocio: el LAIM, que es quien implementa las operaciones básicas del SIW GesIMED; y el SACim, que se ocupa del procesamiento de las consultas y gestión de las imágenes médicas. Respecto de las acciones, éstas se obtienen a partir de las actividades de servicio, identificando el conjunto de acciones que son necesarias para la realización de cada una ellas. Así, por ejemplo, para la actividad “realizar pago del servicio”, se identifican dos acciones que son “validar la tarjeta de crédito” y “registrar pago”. Una vez identificadas todas las acciones, éstas se han distribuido en las distintas particiones del diagrama de actividad teniendo en cuenta qué colaborador del negocio realiza cada acción.

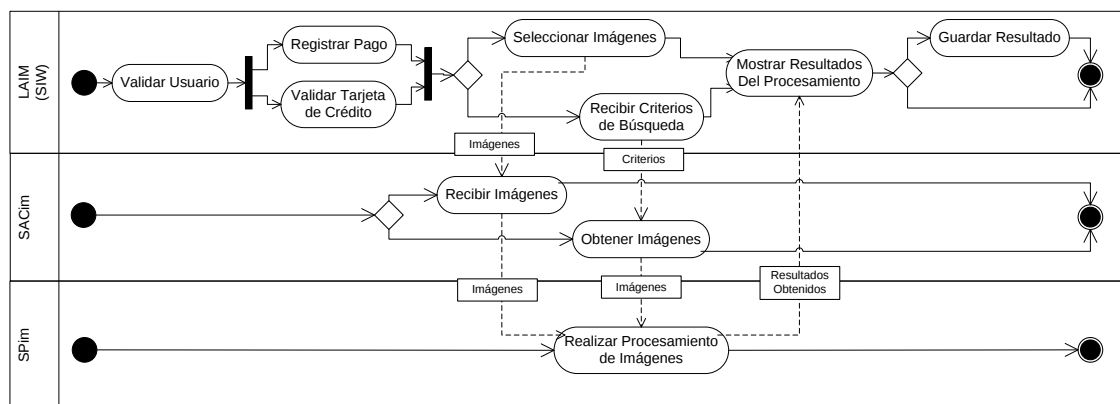


Figura 4.12. Modelo de composición de servicio para el servicio de negocio “realizar procesamiento de imágenes”

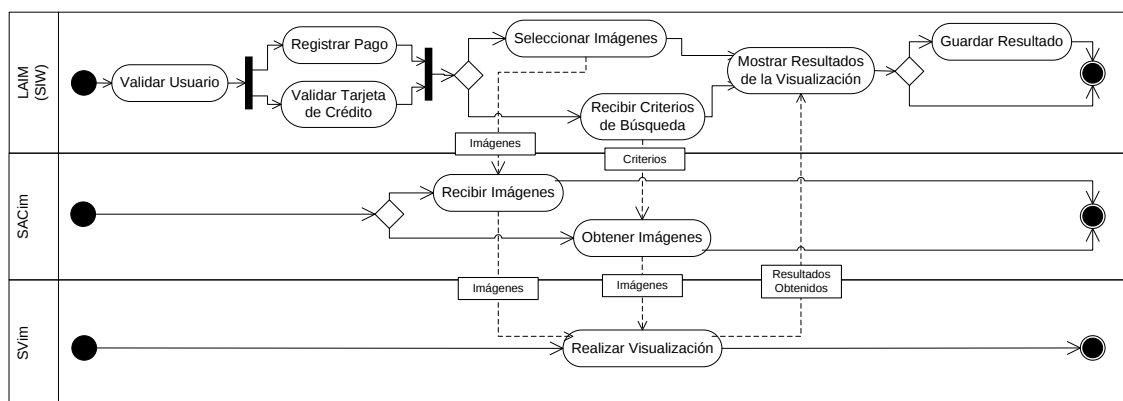


Figura 4.13. Modelo de composición de servicio para el servicio de negocio “realizar visualizaciones de imágenes”

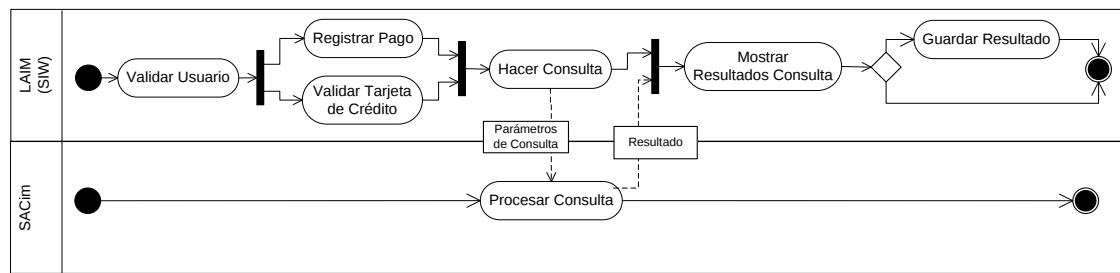


Figura 4.14. Modelo de composición de servicio para el servicio de negocio "realizar consultas"

4.3.4 PSM: Modelado del Comportamiento

Los modelos del nivel PSM propuestos por SOD-M se utilizan para combinar las especificaciones contenidas en los modelos PIM, con detalles concretos de las plataformas basadas en servicios Web.

El primer modelo que se propone obtener es el modelo de composición de servicio extendido. Este modelo permite representar detalles específicos para facilitar la implementación de la composición de servicio en una plataforma basadas en servicios Web. La Figura 4.15, la Figura 4.16 y la Figura 4.17 muestran el **modelo de composición de servicio extendido** para los servicios de negocios implementados en el SIW GesIMED. Para la representación de este modelo ha sido necesario identificar las acciones que se implementarán como servicios Web. Así, por ejemplo, la acción "validar tarjeta de crédito" se ha representado como un servicio Web, puesto que es ya un servicio Web estándar proporcionado por varias organizaciones. También las acciones que realizan los colaboradores del negocio: SACim, SPim y SVim, han sido representadas como servicios Web, puesto que de este modo se facilitaba la comunicación entre el servidor central de la aplicación y los distintos servicios que éstos proporcionan.

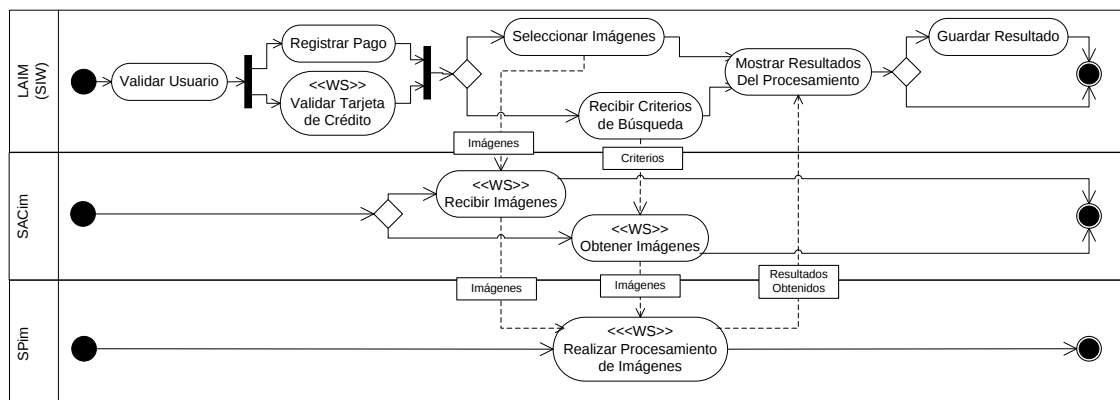


Figura 4.15. Modelo de composición de servicio extendido para el servicio de negocio "realizar procesamiento de imágenes"

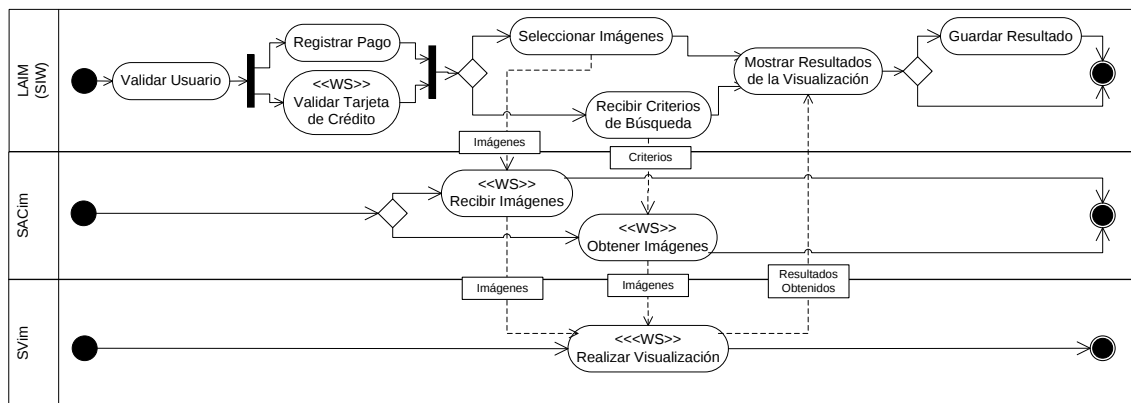


Figura 4.16. Modelo de composición de servicio extendido para el servicio de negocio "realizar visualizaciones de imágenes"

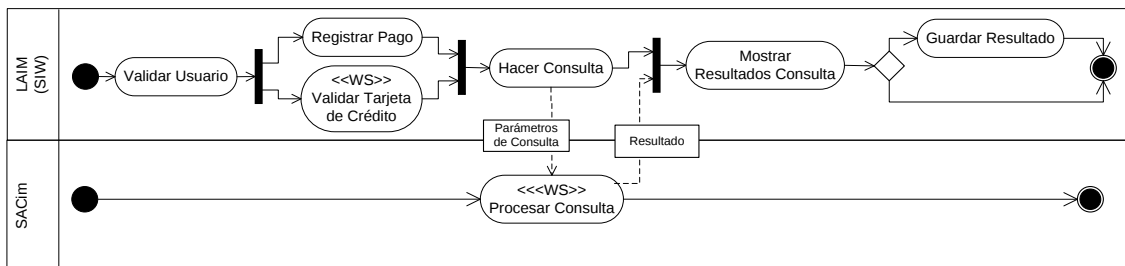


Figura 4.17. Modelo de composición de servicio extendido para el servicio de negocio "realizar consultas"

La Figura 4.18 muestra, a modo de ejemplo, el **modelo de interfaz de servicio Web** para uno de los servicios Web utilizados en el desarrollo de SIW GesIMED, el servicio Web para la validación tarjetas de créditos (que se encuentra disponible en: <http://webservices.imagination.com/validate/Validate.jws?wsdl>). Como puede verse en la Figura 4.18, este modelo permite describir la interfaz del servicio Web en base a los conceptos del estándar WSDL. A partir de dicho modelo es posible la generación automática del código WSDL correspondiente.

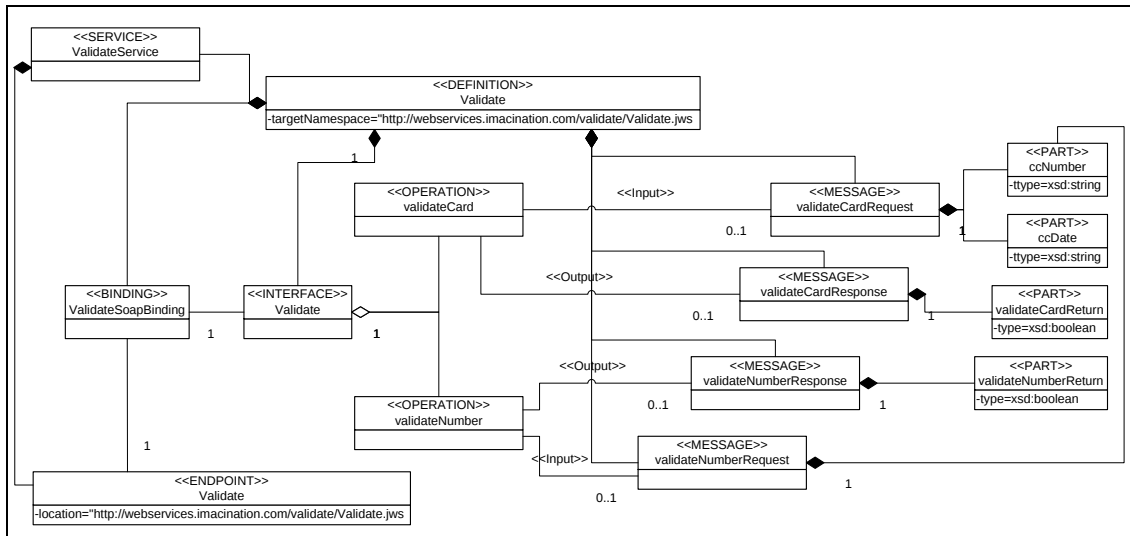


Figura 4.18. Modelo de interfaz de servicio Web para el servicio Web "validar tarjeta de crédito"

4.3.5 Implementación

La arquitectura del SIW GesIMED, que se muestra en la Figura 4.19, se diseñó en base a las características de la tecnología .NET (Microsoft, 2005) y J2EE (Sun Microsystems, 2004). El SIW GesIMED se ha estructurado en tres capas: *presentación, comportamiento y persistencia*.

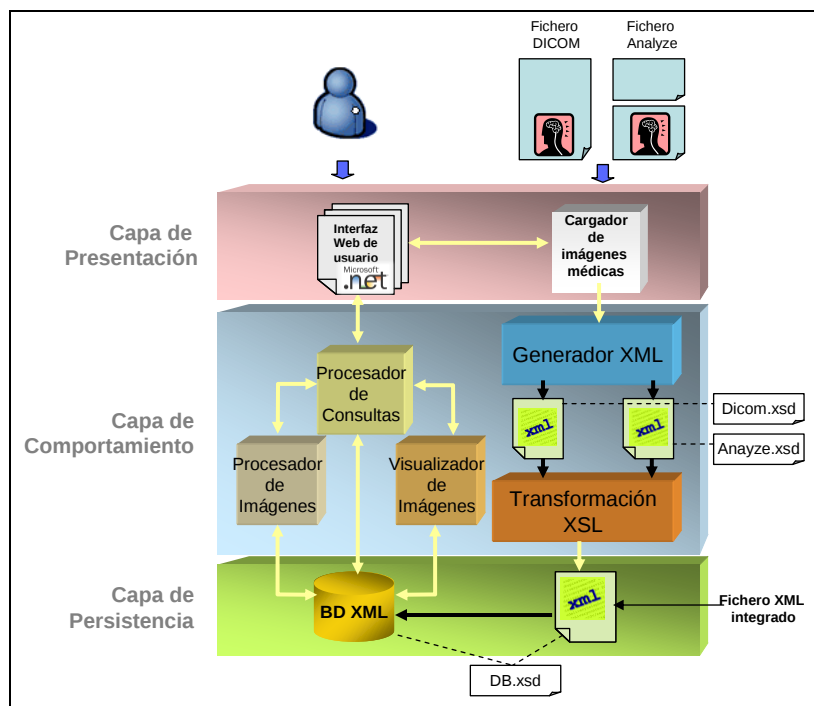


Figura 4.19. Arquitectura el SIW GesIMED

Capa de presentación

Esta capa ofrece dos funcionalidades: por un lado, funciona como interfaz Web de usuario y de herramienta para cargar ficheros de imágenes. El interfaz Web (Figura 4.20) de usuario se implementó utilizando el entorno de trabajo de Microsoft .NET, ASP.Net y C# como lenguaje de programación. Para la carga de imágenes, se utilizaron diferentes clases de la API de Java.

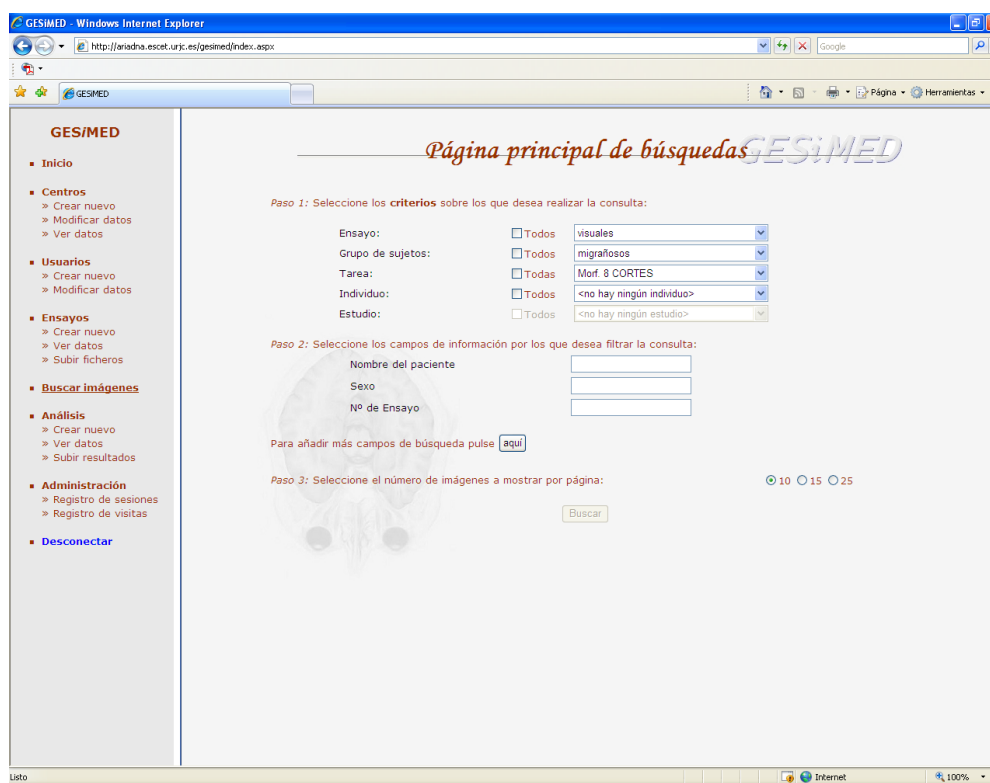


Figura 4.20. Interfaz del SIW GesIMED

Capa de comportamiento

El comportamiento del SIW se ha desarrollado a partir de los modelos descritos previamente. Además, como paso inicial se ha desarrollado el esquema XML (XML Schema) de bases de datos de acuerdo al método definido por MIDAS para el desarrollo de bases de datos para la Web (Marcos *et al.*, 2002). Así, los XML Schemas se han modelado siguiendo los principios indicados en (Vela *et al.*, 2004).

El *Generador de XML* fue desarrollado en Java y su funcionalidad principal consiste en transformar los ficheros DICOM (ACR-NEMA, 2003; Oosterwijk, 2003) o Analyze (Mayo Clinic, 2004) en documentos XML. Tanto DICOM y Analyze, son uno de los estándares más aceptados para el intercambio de información de imágenes médicas. Sin embargo, éstos solo tienen en cuenta la comunicación entre el equipamiento de imágenes médicas y otras herramientas software específicas como

PACS (*Picture Archiving and Communication Systems*), por lo que la transformación de información DICOM y Analyze en XML mejora el intercambio e integración de la información médica.

El *módulo de transformación XSL*, que se desarrolló utilizando XSLT (*Extensible Stylesheet Lang Transformation*) (W3C, 1999b), toma como entradas los ficheros XML intermedios generados por el Generador XML y los transforma en un documento XML diferente adaptado a las particularidades del XML Schema de la base de datos. Así, si el esquema de la base de datos se modifica o si el sistema de gestión de la base de datos cambia, sólo debe ser modificado o sustituido el módulo de Transformación XSL sin que afecte al resto de componentes.

Ambos módulos, tanto el Generador XML como el de Transformación XSL, se encargan así de la funcionalidad asociada al almacenamiento y gestión de los diferentes ficheros proporcionados por los investigadores en la base de datos del LAIM.

El *Procesador de Consultas* es el módulo responsable de construir las consultas de los investigadores en neurociencias para que puedan ser ejecutadas en el sistema gestor de la base de datos, y de mostrar los resultados de las consultas a través del interfaz Web. Para hacer las consultas, el usuario rellena los datos en los formularios Web y después, este módulo transforma los datos en expresiones XPath de consulta (W3C, 1999a).

El módulo *Procesador de Imágenes* permite realizar el procesamiento de las imágenes médicas. Los procedimientos software se almacenan en la base de datos. Éstos, pueden ser seleccionados mediante una consulta y lanzados automáticamente para procesar un conjunto de imágenes. Los resultados y la información de los procedimientos utilizados para obtener tales resultados se almacenan conjuntamente en la base de datos para que puedan ser accedidos y consultados tanto por los miembros del LAIM como los investigadores en neurociencia.

El *Visualizador de Imágenes* se encarga de llevar a cabo las visualizaciones, utilizando procedimientos similares a los utilizados por el procesador de imágenes.

Capa de Persistencia

Para la definición del aspecto de contenido de SIW se ha seguido el método de desarrollo del dicho aspecto propuesto por la metodología MIDAS (Vela *et al.*, 2004).

Para la implementación de la BD, se ha utilizado una herramienta comercial basada en la extensión XML DB de Oracle debido a su fiabilidad en cuanto a realización de consultas, actualizaciones optimizadas y validaciones exhaustivas.

CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se presentan las conclusiones de este trabajo de Tesis Doctoral y las futuras líneas de investigación. En primer lugar, en la sección 5.1, se analiza la consecución de los objetivos planteados al inicio de este trabajo, y que han sido presentados en el capítulo 1. Posteriormente, en la sección 5.2, se indican las principales aportaciones de esta Tesis Doctoral. En la sección 5.3, se contrastan los resultados obtenidos a través de la presentación de las diferentes publicaciones realizadas. Finalmente, en la sección 5.4 se exponen las líneas de investigación que permanecen abiertas.

5.1 Análisis de la Consecución de Objetivos

En la sección 1.2 de esta memoria de Tesis Doctoral, se planteó el objetivo principal de este trabajo de investigación:

"La especificación de un método basado en MDA, que facilite el desarrollo orientado a servicios de SIW, partiendo del modelado del negocio de alto nivel y permitiendo obtener los modelos específicos para plataformas basadas en servicios Web".

Para la consecución de este objetivo, se plantearon un conjunto de objetivos parciales. A continuación, se pasa a analizar el resultado obtenido para cada uno de ellos:

- **Objetivo 1:** *análisis de las distintas metodologías existentes para el desarrollo de SIW, prestando especial atención a sus propuestas para el modelado del comportamiento del sistema, determinando sus aportaciones así como sus limitaciones, con el fin de abordar problemas no resueltos.*

Se ha realizado un estudio sobre las principales metodologías para el desarrollo de SIW, poniendo especial énfasis en los modelos que se proponen para la especificación de la funcionalidad o comportamiento del sistema y analizando otros aspectos generales de las mismas como paradigma de desarrollo utilizado, su orientación, utilización de UML y su adaptación a la propuesta de MDA.

A través de este estudio se ha comprobado, como aspecto más importante, la ausencia de propuestas que utilicen una orientación a servicios para el desarrollo de SIW. Además, si bien han detectado metodologías que han extendido sus modelos

para dar soporte a las nuevas propuestas tecnológicas para la Web, tales como servicios Web o automatización de procesos de negocio, en general, se observa que todas ellas continúan manteniendo sus procesos y enfoques tradicionales para el desarrollo de SIW, realizando a veces pequeñas modificaciones que sólo afectan al hipertexto de la aplicación Web. Este estudio se ha presentado en la sección 2.2 de esta memoria.

- **Objetivo 2:** *análisis de las principales propuestas relacionados con el desarrollo orientado a servicios, tales como propuestas para el desarrollo de servicios Web, para la composición de los mismos o para el desarrollo de procesos de negocios basados en servicios. Dichas propuestas se analizan con el fin de determinar sus aportaciones, he incorporarlas si fuera posible en el método que se va a definir.*

Como parte de este trabajo de investigación, se han analizado propuestas que, aunque no se ocupan del desarrollo completo de SIW, sí han realizado importantes aportaciones en el modelado y desarrollo de ciertos aspectos relacionados con el desarrollo orientado a servicios. Así, se han estudiado las diversas técnicas y métodos propuestos para el desarrollo de servicios Web, para el desarrollo de la composición de servicios Web y propuestas para el desarrollo de procesos de negocios que utilizan servicios Web.

En el análisis de tales propuestas, que se presentó en la sección 2.3, se ha comprobado que, aunque sí se han realizado importantes aportaciones en el modelado y desarrollo de ciertos aspectos relacionados con la orientación a servicios en el desarrollo de sistemas, en la actualidad, no existen propuestas metodológicas que aborden el desarrollo completo de sistemas, con este enfoque.

- **Objetivo 3:** *definición de un método para el modelado del comportamiento de SIW utilizando una aproximación orientado a servicios que incluya:*
 - *Modelos, tanto independientes de computación como independientes y específicos de plataforma, ya sea mediante la definición de los mismos o bien mediante la incorporación en el método de modelos ya existentes.*
 - *Transformaciones entre modelos, que permitan obtener modelos específicos para la tecnología de servicios Web a partir del modelado de negocios de alto nivel.*

En esta Tesis Doctoral se ha propuesto una aproximación metodológica basada en MDA para el desarrollo orientado a servicios de SIW, llamada SOD-M. Dicha propuesta, presentada en el capítulo 3, se centra en el desarrollo del aspecto del comportamiento de los SIW y define guías para la integración con los demás aspectos de la arquitectura de MIDAS, que provee un marco basado en MDA para el desarrollo completo de SIW.

Como parte del método, se han propuesto diversos modelos que se corresponden con los diferentes niveles de abstracción de MDA. Para ello, se han definido dos perspectivas de análisis: la perspectiva del negocio, que coincide con los modelos de nivel CIM, y que se centra en el análisis de las características del negocio en el que se desenvolverá el sistema que se va a construir; y la perspectiva del sistema comportamiento, que coincide con los modelos de nivel PIM y PSM de la arquitectura de MDA, y que se centra en el análisis de las funcionalidades y procesos necesarios para el desarrollo del comportamiento del sistema que se va a construir. Además, se ha definido un perfil UML que incluye todos los elementos necesarios el modelado del comportamiento de un SIW, a nivel PIM y PSM, desde una aproximación orientada a servicios.

Se ha definido, además, el proceso de desarrollo que se extiende desde el modelado del negocio de alto nivel hasta los modelos específicos para la plataforma de servicios Web. Dicho proceso se ha especificado, describiendo el conjunto de tareas necesarias para la generación de cada uno de los modelos, así como el conjunto de transformaciones entre modelos de nivel CIM, PIM y PSM.

- **Objetivo 4:** *integración del método propuesto para el desarrollo orientado a servicios del comportamiento de SIW en el marco de la arquitectura de MIDAS, teniendo en cuenta los demás aspectos contemplados en dicha arquitectura para el desarrollo de SIW.*

Además de la definición del método SOD-M, que se centra en el aspecto del comportamiento de un SIW, se han presentado guías para la integración de los modelos propuestos con los demás modelos definidos en el marco de MIDAS; en concreto, se han especificado reglas de transformación horizontales entre los modelos de SOD-M y los modelos del método para el desarrollo del hipertexto de MIDAS.

Esta integración, que se ha descrito en la sección 3.7, permite completar el proceso de desarrollo del SIW. Además, debido a que también se mantiene el

enfoque orientado a servicios en el desarrollo del hipertexto de los SIW, es posible el desarrollar sistemas más intuitivos y fáciles de navegar por parte de los usuarios, puesto que se propone construir el modelo de navegación del SIW en base a los servicios que son requeridos por los usuarios así como en los procesos necesarios para llevar a cabo dichos servicios.

- **Objetivo 5:** *validación del método propuesto mediante su aplicación a diferentes casos de estudio.*

La definición y validación del método propuesto en esta Tesis Doctoral, que ha sido presentada en el capítulo 4, se ha ido realizando mediante un proceso iterativo e incremental basado en el método de investigación en acción. Para ello se han desarrollado diferentes casos de estudio de laboratorio y reales, pertenecientes a diversos dominios de aplicación.

En concreto, se desarrollaron dos casos de estudio de laboratorio, uno de ellos realizado específicamente para determinar las necesidades en el modelado de servicios Web y otro, el SIW WebConference, para detectar necesidades básicas del desarrollo orientado a servicios de SIW, como por ejemplo, la necesidad de definir modelos específicos que permitan construir el SIW en base a los servicios que el usuario espera del mismo.

Posteriormente, se han llevado a cabo cuatro casos de estudio reales, los cuales han permitido detectar nuevas necesidades a la vez que se iban validando las primeras versiones del método propuesto. Dos de los casos de estudio desarrollados, los SIW GesIMED y PARBUD, han sido especialmente útiles para detectar nuevas necesidades en el modelado de negocio de alto nivel. Los restantes casos de estudio reales, los SIW ATENTO e ISLA, han sido desarrollados aplicando el método completo, permitiendo de este modo la validación de la propuesta de forma completa; ambos casos fueron, además, desarrollados por personas ajenas a la universidad, lo que ha permitido detectar y corregir algunas inconsistencias en los modelos que se habían propuesto.

5.2 Principales Aportaciones

El trabajo realizado ha supuesto un conjunto de aportaciones relativas al tema de investigación abordado en esta Tesis Doctoral, la definición de una aproximación metodológica basada en MDA para el desarrollo orientado a servicios de SIW, que se exponen a continuación:

Un nuevo enfoque orientado a servicios para el desarrollo de sistemas

Como aportación principal de esta Tesis Doctoral se ha definido un nuevo enfoque orientado a servicios para el desarrollo de sistemas, que este caso concreto se ha aplicado al caso de los SIW. En concreto se han propuesto guías para la construcción de aplicaciones Web en base a servicios, es decir, funcionalidades simples o complejas que son ofrecidas por una entidad y que permite satisfacer las necesidades de otra entidad que las solicita. Este nuevo enfoque metodológico permite a los desarrolladores de software aprovechar al máximo los beneficios del paradigma de computación orientado a servicios, favoreciendo el desarrollo de servicios y procesos de composición de servicios significativos.

Como parte de la propuesta realizada, se ha definido un nuevo conjunto de conceptos para el desarrollo de SIW desde una aproximación orientada a servicios; algunos de ellos, necesarios para la descripción de los elementos propios del negocio, los cuales se representan en diversos modelos de negocios propuestos a nivel CIM de la arquitectura de MDA, y otros, necesarios para el modelado de las funcionalidades del sistema, que se representan en los modelos definidos para el nivel PIM y PSM de la arquitectura de MDA.

Además, se ha definido de manera completa el conjunto de tareas necesarias para la generación de cada uno de los modelos propuestos a nivel CIM, PIM y PSM, así como las transformaciones entre estos. De esta manera, se ha permitido alcanzar el principal objetivo de este trabajo de Tesis Doctoral, que es la definición de un método que permita obtener modelos específicos para plataformas basadas en servicios Web a partir de los modelos de negocio de alto nivel.

Integración del método propuesto en el marco de la arquitectura MDA para el desarrollo de SIW

El método propuesto en esta Tesis Doctoral se centra en el aspecto del comportamiento de los SIW. Sin embargo, como parte de esta trabajo de Tesis Doctoral se presentan las guías para la integración de los demás aspectos contemplados por la arquitectura de MIDAS para el desarrollo de SIW. En concreto, se definen las reglas de transformación necesarias para la obtención de los modelos del aspecto del hipertexto, por ser este el eje vertebrador dentro de la arquitectura de MIDAS, a partir de los modelos del comportamiento propuestos por SOD-M. Así, todo el proceso de modelado y desarrollo del SWI se centra en los servicios que serán ofrecidos a los usuarios de la Web.

Este enfoque permite el desarrollo de aplicaciones basadas en el comportamiento (servicios) que los usuarios esperan de la aplicación Web, lo que a su vez favorece el desarrollo de SIW más intuitivos y fáciles de navegar por parte de los usuarios.

Perfil UML para el desarrollo del comportamiento de SIW

El método definido, propone la utilización de UML para el modelado completo del sistema a nivel PIM y PSM. Por ello, como parte de este trabajo de Tesis Doctoral se ha definido un perfil UML, llamado perfil UML SOD-M, que agrupa todos los elementos para el modelado a nivel PIM y PSM, del aspecto del comportamiento de un SIW desde un enfoque orientado a servicios.

Además, se ha definido el metamodelo asociado a cada uno de los nuevos modelos UML propuestos, tanto a nivel PIM como PSM, en los que se incluyen los nuevos elementos de modelado propuestos.

5.3 Contrastación de Resultados

Algunas de las contribuciones realizadas como parte de este trabajo de investigación han sido contrastadas en diferentes foros tanto nacionales como internacionales. A continuación se describen las publicaciones realizadas, agrupadas por tipo de publicación:

Capítulos de Libro:

- B. Vela, P. Cáceres, V. de Castro, E. Marcos. MIDAS: Una Aproximación Dirigida por Modelos para el Desarrollo Ágil de Sistemas de Información Web. *Ingeniería de la Web y patrones de diseño*. Paloma Díaz, Susana Montero e Ignacio Aedo (Eds.). Pearson Education, Junio de 2005.
- V. de Castro, C. Acuña, E. Marcos. Servicios Web. *Factorías de Software: Experiencias, tecnologías y organización*. Mario Piattini y Javier Garzás (eds.). Ra-Ma. Pendiente de publicación.

Artículos en Revistas Internacionales:

- P. Cáceres, J. Ríos, V. de Castro, D. Ríos Insua, E. Marcos. Improving usability in e-democracy systems: systematic development of navigation in an e-participatory budget system. ***International Journal of Technology, Policy and Management***. Special Issue of Towards

Electronic Democracy: Internet-based Complex Decision Support.
Aceptado para publicación.

- V. De Castro, E. Marcos, M. López Sanz. A Model Driven Method for Service Composition Modeling: A Case Study. ***International Journal of Web Engineering and Technology***, Vol. 2(4), Eds: Inderscience Enterprise Ltd, ISSN: 1476-1289, pp. 335-353, Julio 2006.
- J.A. Hernández, C. Acuña, V. de Castro, E. Marcos, M. López, N. Malpica. A WEB-PACS for Multi-center Clinical Trials. ***IEEE Transactions on Information Technology in Biomedicine***, Vol, 11 (1), ISSN: 1089-7771, pp. 87-93, 2006. **(Factor de impacto JCR: 1.575)**
- J.M. Vara, V. De Castro, E. Marcos. WSDL automatic generation from UML models in a MDA framework. *International Journal of Web Services Practices*, Vol. 1, Ed.: Web Services Research Foundation, Sang Yon Hang (Eds.), ISSN: 1738-6535, pp. 1-12, Septiembre 2005.
- C. Acuña, E. Marcos, V. de Castro, J.A. Hernández. Managing Medical Images. *ERCIM News*, Vol. 58, Ed.: ERCIM EEIG, ISSN: 0926-4981, pp. 54-55, 2004.

Artículos en Revistas Iberoamericanas:

- E. Marcos, V. de Castro, B. Vela. Una extensión de UML para el modelado de Servicios Web. *Revista Digital Científica y Tecnológica E-Gnosis*, Vol. 22(2), Ed.: Universidad de Guadalajara, Jalisco, (Mexico), ISSN: 1665-5745, Disponible en: <http://www.e-gnosis.com.mx/>, 2004.
- V. de Castro, E. Marcos, B. Vela. Representing WSDL with Extended UML. *Revista Colombiana de Computación*, Vol. 5(1), Ed.: Universidad Autónoma de Bucaramanga (Colombia), ISSN: 1657-2831, pp. 7-21, 2004.
- P. Cáceres, E. Marcos, V. de Castro. Modelado Navegacional desde una Perspectiva Orientada a Servicios de Usuario. ***IEEE América Latina***, Vol. 3(1), J. Hernández, E. Pimentel y A. Toval (Eds.), ISSN: 1548-0992. Disponible en: <http://www.ewh.ieee.org/reg/9/etrans/index.html>, 2004.

Artículos en Congresos Internacionales:

- G. López, V. de Castro, E. Marcos. *Implementation of Business Process requiring User Interaction*. Actas de On the Move to Meaningfull Internet

- Systems 2006: OMT 2006 Workshops: En International Workshop on Agents, Web Services and Ontologies Merging (AWeSOMe'06). R. Meersman, Z. Tari, P. Herrero et al. (Eds.), Springer Verlag, LNCS 4277, pp. 107-115, 2006.
- G. López, V. de Castro, E. Marcos. *Service Oriented Method to Model Web Information Systems: A Case Study*. Actas de IADIS International Conference WWW/INTERNET 2006, Vol II, P. Isaías, M. Baptista Nunes, I.J. Martínez (Eds), 2006. IADIS Press, pp. 172-176, 2006.
 - V. de Castro, M. López Sanz, E. Marcos. *Business Process Development based on Web Services: a Web Information System for Medical Image Management and Processing*. **Actas de IEEE International Conference on Web Services (ICWS 2006)**. IEEE Computer Society, F. Leymann, L.J. Zhang (Eds.), pp. 807-814, 2006. **(Ratio de aceptación: 33%)**
 - F. Lucas, F. Molina, A. Toval, P. Cáceres, V. de Castro, E. Marcos. *Precise WIS Development*. **Actas de 6th International Conference on Web Engineering (ICWE 2006)**. ACM Pres, D. Wolber, N. Calder, C. Brooks, A. Ginige (Eds.), pp.71-77, 2006.
 - P. Cáceres, V. de Castro, J.M. Vara, E. Marcos. *Model Transformations for Hypertext Modeling on Web Information Systems*. **Actas de 21st Annual ACM Symposium on Applied Computing (SAC 2006)** - Track on Model Transformation, Ed.: The Association for Computing Machinery, Inc. pp. 1232-1239, 2006. **(Ratio de aceptación: 32%)**
 - J.M. Vara, V. de Castro, E. Marcos. *WSDL automatic generation from UML models in a MDA framework*. Actas de International Conference on Next Generation Web Services Practices (NWeSP 2005), A. Abraham, S.Yong Han, D. Hung-Chang Du, M. Paprzycki (Eds.), IEEE Computer Society Press, pp.319-324, 2005.
 - V. de Castro, E. Marcos, P. Cáceres. *A User Services Oriented Method to model Web Information Systems*. **Actas de 5th International Conference on Web Information Systems Engineering (WISE 2004)**, X. Zhou, S. Su, M. Papazoglou, M. Orlowska, K. Jeffery (Eds.), Springer Verlag, LNCS 3306, pp. 41-52, 2004. **(Ratio de aceptación: 22%)**
 - P. Cáceres, V. de Castro, E. Marcos. *Navigation Modeling from a User Services Oriented Approach*. **Actas de The 3rd Biennial International**

Conference in Advanced in Information Systems (ADVIS 2004), T. Yakhno (Ed.), Springer Verlag, LNCS 3271, pp. 150-160, 2004. (Ratio de aceptación: 30%)

- C. Acuña, E. Marcos, V. de Castro, J.A. Hernández. *A Web Information System for Medical Image Management*. Actas de 5th International Symposium on Biological and Medical Data Analysis, J.M. Barreiro, F. Martin-Sanchez, V. Maojo, F. Sanz (Eds.), Springer Verlag, LNCS 3337, pp. 49-59, 2004. **(Ratio de aceptación: 32%)**
- P. Cáceres, E. Marcos, V. de Castro. *Integrating agile and model-driven practices in a methodological framework for the Web Information Systems development*. Actas de The 6th International Conference on Enterprise Information Systems (ICEIS 2004), Vol. 3, I. Seruca, J. Filipe, S. Hammoudi, J. Cordeiro (Eds.), pp.523-526, 2004.
- E. Marcos, P. Cáceres, V. de Castro. *Navigation Modeling with Services*. Actas de The IADIS International Conference (WWW/Internet 2004), Vol. 1, P. Isaias, N. Karmakar (Eds.), pp. 723-730, 2004.
- E. Marcos, P. Cáceres, V. de Castro. *An approach for the Navigation Model Construction from the Use Case Model*. Actas de The 16th Conference on Advanced Information Systems Engineering. CAISE'04 FORUM, J. Grabis, A. Persson, J. Stirna (Eds.), pp. 83-92, 2004.
- E. Marcos, V. de Castro, B. Vela. *Representing Web Services with UML: A Case Study*. **Actas de 1º International Conference on Services Oriented Computing (ICSOC 2003)**, M. Orlowska, S. Weerawarana, M. Papazoglou, J. Yang (Eds.), Springer Verlag, LNCS-2910, pp. 15-27, 2003. **(Ratio de aceptación: 21%)**

Artículos en Congresos Iberoamericanos:

- V. de Castro, E. Marcos, M. López Sanz. *Service Composition Modeling: A Case Study*. Actas de 7th Mexican International Conference on Computer Sciences. IEEE Computer Society, S. Rajsbaum (Ed.), pp. 101-108, 2006. (Ratio de aceptación: 26%)
- V. De Castro, M. Genero, E. Marcos, M. Piattini. *Construyendo Sistemas de Información Web Más Fáciles de Navegar: Un Estudio Experimental*. Actas de 7th Argentine Symposium on Software Engineering (ASSE 2006), CD ASSE 2006, pp.141-154, 2006.

- V. de Castro, M. López Sanz, E. Marcos. *Modelado de Procesos de Negocios Basados en Servicios Web*. Actas del 9º Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software (IDEAS 2006), J. Castro, L. Cernuzzi, S. Gordillo (Eds.), pp. 497-500, 2006.
- J.M. Vara, V. de Castro, P. Cáceres y E. Marcos. *Arquitectura de MIDAS-CASE: una herramienta para el desarrollo de SIW basada en MDA*. Actas de IV Jornadas Iberoamericanas en Ingeniería del Software e Ingeniería del Conocimiento (JIISIC 2004), O. Dieste, A. Moreno (Eds.), pp.83-92, 2004.
- P. Cáceres, V. de Castro, E. Marcos, C. Acuña. *Del Modelo de Casos de Uso al Modelo de Navegación: Un enfoque Orientado a Servicios*. Actas del VII Workshop Iberoamericano de Ingeniería de Requisitos y Desarrollo de Ambientes de Software (IDEAS 2004), L. Olsina, E. Cuadros-Vargas, A. Vasconcelos (Eds.), pp. 124-135, 2004.
- E. Marcos, V. de Castro, B. Vela. *Modelado de Servicios Web con UML: Un Caso de Estudio*. Actas de Workshop on Advances in Database and Information Retrieval. Avances en Ciencias de la Computación, J. H. Sossa Azuela, E. Pérez Cortés (Eds.), pp. 23-28, 2003.

Artículos en Congresos Nacionales:

- G. López, V. de Castro, E. Marcos. *Modelado de Sistemas de Información Web Orientados a Servicios: Un Caso de Estudio*. Actas de las II Jornadas Científico-Técnicas en Servicios Web (JSWEB 2006). M. Lama, F. Curbera, J.C. del Arco, E. Sánchez (Eds.), pp. 38-43, 2006.
- M. López Sanz, V. de Castro, J.L. Bosque, E. Marcos. *Estudio comparativo del desarrollo con Servicios Web y Servicios Grid en un marco basado en MDA: aplicación a un caso de estudio*. Actas de las II Jornadas Científico-Técnicas en Servicios Web (JSWEB 2006). M. Lama, F. Curbera, J.C. del Arco, E. Sánchez (Eds.), pp. 23-29, 2006.
- G. López, M. López Sanz, V. de Castro, E. Marcos. *Diseño de procesos de negocio que requieren la intervención del usuario: un estudio sobre las alternativas de implementación*. En Actas del Taller de Desarrollo y Mantenimiento Ágil de Aplicaciones Basadas en Servicios Web, en conjunción con las XI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2006), Sitges, Barcelona, pp. 39-49, 2006.

- V. de Castro, M. López Sanz. *Modelado de Composición de Servicios: Un Caso de Estudio*. Actas de las I Jornadas Científicos-Técnicas en Servicios Web (JSWEB 2005), E. Marcos, J. M. Alonso, V. de Castro, J. C. del Arco (Eds.), pp. 149-156, 2005.
- J.M. Vara, V. de Castro. *Generación Automática de WSDL a partir de Modelos UML*. Actas de I Jornadas Científico-Técnicas en Servicios Web (JSWEB 2005), E. Marcos, J. M. Alonso, V. de Castro, J. C. del Arco (Eds.), pp. 157-164, 2005.
- P. Cáceres, E. Marcos, V. de Castro. *Modelado Navegacional desde una Perspectiva orientada a Servicios de Usuario*. Actas de las IX Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2004). J. Hernández, E. Pimentel (Eds.), pp. 111-122, 2004.
- V. de Castro, E. Marcos, B. Vela. *Una extensión para la representación de WSDL*. Actas del Taller de Nuevas Tecnologías de la Información: Componentes y Servicios Web, en conjunción con las VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2003), P.J. Clemente, J.M. Corchado, R. Corchuelo, J. Pavón, D. Sevilla (Eds.), pp. 45-54, 2003.
- P. Cáceres, E. Marcos, V. de Castro. *Marco metodológico para el desarrollo de sistemas de información Web basado en MDA*. Actas de Taller de Ingeniería del Software Orientada a la Web (WebE'03), en conjunción con las VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2003), 2003.

5.4 Líneas de Investigación Abiertas

Como continuación de este trabajo de Tesis Doctoral, existen diversas líneas de investigación que quedan abiertas y en las que es posible continuar trabajando. Algunas de ellas, están más directamente relacionadas con este trabajo de Tesis Doctoral y son el resultado de cuestiones que han ido surgiendo durante la realización del mismo. Otras, sin embargo, son líneas más generales y proceden de problemas que se conocían pero que, sin embargo, no son objeto de esta Tesis Doctoral; tales líneas están siendo abordadas por otros miembros del grupo de investigación Kybele y, en algunos casos, en colaboración con miembros de otras universidades:

- En primer lugar, una de las principales líneas de investigación abiertas está relacionada con el desarrollo orientado a servicios en sí mismo. Como

ya se ha dicho antes, el paradigma de computación orientada a servicios, como todo nuevo paradigma, proporciona nuevas facilidades y nuevos retos de investigación. Esta Tesis Doctoral se ha desarrollado en esta línea, pero es evidente que aún queda mucho por hacer. Surgen así, nuevas líneas de investigación relacionadas con otros **aspectos importantes del desarrollo orientado a servicios** tales como aspectos de diseño de arquitecturas orientadas a servicios, políticas de descubrimiento de servicios, cuestiones relacionadas con la calidad de servicios, etc.

- Aunque en este trabajo de Tesis Doctoral se ha propuesto un conjunto de transformaciones asociado al método, que ha sido especificado en lenguaje natural y representado luego mediante gramáticas de grafos, aún queda mucho trabajo por hacer en este sentido. Sería conveniente realizar un análisis detallado de las diferentes propuestas existentes y de las líneas abiertas para la **especificación de mappings**, así como completar la formalización de las reglas propuestas en esta Tesis Doctoral. Además, en este sentido, se puede decir que el principal reto de investigación se centra en el estudio de las **transformaciones entre modelos de nivel CIM y los demás modelos propuestos en la arquitectura de MDA**. La formalización de los metamodelos y mappings de MIDAS está siendo realizada en el marco de la tesis de D. Juan Manuel Vara Mesa.
- Hasta el momento, y tal como se describió en esta memoria de Tesis Doctoral en MIDAS se ha considerado como PSM las plataformas basadas en servicios Web (WSDL, BPEL, .NET). Sin embargo, quedan abiertas otra líneas de trabajo como ser la **definición de modelos específicos adaptados a tecnologías concretas** o bien, a la **adaptación de la arquitectura MDA de MIDAS a otras plataformas específicas**. En concreto, en este tema se ha comenzado a trabajar, en el marco de la Tesis Doctoral de D. Marcos López Sanz y en colaboración con el grupo de Arquitecturas de Altas Prestaciones de la URJC, en la adaptación de MIDAS a plataformas Grid.
- La tendencia actual al desarrollo dirigido por modelos (MDD - Model Driven Development) sin duda pasa por la **generación de una herramienta MDA**, que automatice en la medida de lo posible la generación de código a partir de modelos previamente definidos. En esta línea se está trabajando en el desarrollo de una herramienta modular y fácilmente escalable, de

modo que pueda ser extendida a medida que se decida incluir nuevos modelos en la arquitectura de MIDAS.

- Además, a pesar de que el método propuesto ha sido validado mediante su aplicación a diversos casos de estudios reales, una continuación directa de este trabajo es la **validación formal** de los modelos y métodos propuestos en el marco de esta tesis. En este sentido, en la actualidad se está colaborando con el grupo de Ingeniería del Software de la Universidad de Murcia (dirigido por el Dr. Ambrosio Toval) en la formalización de diversos metamodelos de la metodología MIDAS, incluyendo los definidos en SOD-M
- Otra línea de investigación abierta, hace referencia a la adaptación de la metodología MIDAS al desarrollo de SIW concretos. En este sentido, y en el marco de la tesis de D. Cesar J. Acuña, se está trabajando en la adaptación de MIDAS al **desarrollo de aplicaciones de integración de portales Web**, mediante la utilización de servicios Web semánticos.
- Por último, se piensa continuar trabajando en **nuevos casos de estudio** aplicando el método propuesto en esta Tesis Doctoral, para detectar así futuras mejoras o extensiones al método propuesto. En concreto, y como primer paso, se continuará el trabajo iniciado durante la estancia realizada en la Universidad Tecnológica Nacional en Argentina, sobre el desarrollo de un SIW para la gestión de historias clínicas únicas y automatizadas que se implantará en hospitales públicos.

BIBLIOGRAFÍA

- ACR-NEMA, 2003. The DICOM Standard, Disponible en: <http://medical.nema.org/>
- Acuña *et al.*, 2004. A Web Information System for Medical Image Management. Acuña, C., Marcos, E., de Castro, V., Hernández, J.A. En actas de 5th International Symposium on Biological and Medical Data Analysis (ISBMDA 2006). Springer Verlag, LNCS 3337, pp. 49-59.
- Acuña *et al.*, 2004b. Managing Medical Images. Acuña, C., Marcos, E., de Castro, V., Hernández, J.A. ERCIM News, Vol. 58, Ed.: ERCIM EEIG, ISSN: 0926-4981, pp. 54-55.
- Acuña y Marcos, 2006. Modeling semantic web services: a case study. Acuña, C., Marcos, E. Proceedings of the 6th International Conference on Web Engineering (ICWE 2006). ACM Pres, Eds: D. Wolber, N. Calder, C. Brooks, A. Ginige, pp. 32-39.
- Andrews *et al.*, 2003. Business Process Execution Language for Web Services Version 1.1 Specification. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S. Disponible en: <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>.
- Atzeni *et al.*, 1998. Design and Maintenance of Data-Intensive Web Sites. Advances in Database Technology. Atzeni, P., Mecca, G., Merialdo, P. Actas de EDBT'98, Ed.: Sheck, Saltor, Ramos, Alonso. Springer Verlag, LNCS 1377, pp. 436-450.
- Avison *et al.*, 1999. Action Research. Avison, D., Lan, F., Myers, M. y Nielsen, A. Communications of the ACM, 42(1), pp. 94-97.
- Baïna *et al.*, 2004. Model Driven Web Services Development. Baïna, K., Benatallah, B., Casati, F., Toumani, F. Actas de 16^o International Conference on Advanced Information Systems Engineering (CAISE'2004). Ed.: A. Persson y J. Stirna, Springer Verlag, LNCS 3084, pp. 290-306.
- Baresi *et al.*, 2000. From Web Sites to Web Applications: New Issues for Conceptual Modeling. Baresi, L., Garzotto, F., Paolini, P. Actas de 2^o International Workshop on the World Wide Web and Conceptual Modeling (WCM2000), pp. 89-100.
- Baresi *et al.*, 2001. Extending UML for Modeling Web Applications. Baresi, L., Garzotto, F., Paolini, P. Proceedings of the 34th International Conference on System Sciences.
- Bauer y Müller, 2004. MDA Applied: From Sequence Diagrams to Web Service Choreography. Bauer, B., Müller, P. Actas de 4th International Conference on Web Engineering (ICWE 2004). Ed.: N. Koch, P. Fraternali, M. Wirsing. Springer Verlag, LNCS 3140, pp. 132-136.
- Baumeister *et al.*, 1999. Towards a UML Extension for Hypermedia Design. Baumeister H., Koch N., Mandel L. Actas de International Conference on the Unified Modeling Language (UML'1999). The Unified Modelling Language - Beyond the Standard, Ed.: R. France y B. Rumpe, Springer Verlag, LNCS 1723, pp. 614-629.

Bibliografía

- Beck, 2001. *Extreme Programming Explained: Embrace Change*. Beck, K. Editorial Addison-Wesley.
- Bichler y Lin, 2006. *Service-Oriented Computing*. Bichler, M., Lin, KJ. *IEEE Computer* 39(3), pp. 99-101.
- Bonifati *et al.*, 2000. *Building Multi-device, Content-Centric Applications Usign WebML and the W3I3 Tool Suite*. Bonifati, A., Stefano, C., Fraternali, P., Maurino, A. *Actas de ER 2000 Workshops on Conceptual Modeling Approaches for E-Business and The World Wide Web and Conceptual Modeling*, Ed.: S. W. Liddle, H. C. Mayr, B. Thalheim, Springer Verlag, LNCS 1921, pp. 64-75.
- Booch *et al.*, 1999. *The Unified Modelling Language*. Booch, G., Rumbaugh, J., Jacobson I. Addison Wesley.
- BPMI, 2004. *Business Process Modeling Notation (BPMN) Version 1.0*. BPMI Notation Working Group. Disponible en: <http://www.bpmn.org/>
- Brambilla *et al.*, 2003a. *Specification and Design of Workflow-Driven Hypertexts*. Brambilla, M., Ceri, S., Comai, S., Fraternali, P., Manolescu, I. *Journal of Web Engineering* 1(2), pp. 163-182.
- Brambilla *et al.*, 2003b. *Model-driven Development of Web Services and Hypertext Applications*. Brambilla, M., Ceri, S., Comai, S., Fraternali, P., Manolescu, I. *Actas de The 7th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI2003)*.
- Brambilla *et al.*, 2006. *Process Modeling in Web Applications*. Brambilla, M., Ceri, S., Fraternali, P., Manolescu, I. *ACM Transaction on Software Engineering and Methodology* 15(4), pp. 360-409.
- Brambilla, 2003. *Extending Hypertext Conceptual Models with Process-Oriented Primitives*. Brambilla, M. *Actas del ER 2003*, Ed.: Song et al. Springer Verlag, LNCS 2813, pp. 246-262.
- Bunge, 1976. *La Investigación Científica*. Bunge, M. Ariel S.A. Barcelona.
- Cáceres *et al.*, 2003. *A MDA-Based Approach for Web Information System Development*. Cáceres, P., Marcos, E., Vela, B. *Actas de Workshop in Software Model Engineering*. Disponible en: <http://www.metamodel.com/wisme-2003/>
- Cáceres *et al.*, 2004. *Navigation Modeling from a User Services Oriented Approach*. Cáceres, P., De Castro, V., Marcos, E. *Actas de The 3rd Biennial International Conference in Advanced in Information Systems (ADVIS 2004)*. Ed.: Yakhno, T. Springer Verlag, LNCS 3271, pp. 150-160.
- Cáceres *et al.*, 2004b. *Modelado Navegacional desde una Perspectiva Orientada a Servicios de Usuario*. Cáceres, P., Marcos, E., de Castro, V. *IEEE América Latina*, Vol. 3(1), J. Hernández, E. Pimentel y A. Toval (Eds.), ISSN: 1548-0992. Disponible en: <http://www.ewh.ieee.org/reg/9/etrans/index.html>, 2004.

- Cáceres *et al.*, 2006a. Model Transformations for Hypertext Modeling on Web Information Systems. Cáceres, P., de Castro, V., Vara, J.M., Marcos, E. Actas de 21st Annual ACM Symposium on Applied Computing (SAC 2006) - Track on Model Transformation, Ed.: The Association for Computing Machinery, Inc. pp. 1232-1239, 2006.
- Cáceres *et al.*, 2006b. Improving usability in e-democracy systems: systematic development of navigation in an e-participatory budget system. Cáceres, P., Ríos, J., de Castro, V., Ríos Insua, D., Marcos, E. International Journal of Technology, Policy and Management. Special Issue of Towards Electronic Democracy: Internet-based Complex Decision Support. Aceptado para publicación.
- Castano *et al.*, 2000. A General Methodological Framework for the Development of Web-Based Information Systems. Castano, S., Palopoli, L., Torlone, R., Actas de ER 2000 Workshops on Conceptual Modeling Approaches for E-Business and The World Wide Web and Conceptual Modeling. Ed.: S. W. Liddle, H. C. Mayr, B. Thalheim. Springer Verlag, LNCS 1921, pp. 128-139.
- Ceri *et al.*, 2000. Web Modeling Language (WebML): a modeling language for designing Web sites. Ceri, S., Fraternali, P, Bongio, A. Computer Networks 33(1-6), pp. 137-157.
- Conallen, 2000. Building Web Applications with UML. Conallen, J. Editorial Addison Wesley.
- Cornford y Smithson, 1996. Project Research in Information Systems: A Student's Guide Cornford, T., Smithson, S. Macmillan Press.
- Curbera *et al.*, 2002. Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. Curbera F., Duftler M., Khalaf R., Nagy W., Mukhi N. y Weerawarana S. Internet Computing, IEEE, 6(2), pp. 86-93.
- Curbera *et al.*, 2003. The next step in Web services. Curbera, F., Khalaf, R., Mukhi, N., Tai, S., Weerawarana, S. Communications of ACM 46(10), pp. 29-34.
- Curbera, 2005. Servicios Web: estándares basados en proceso distribuido. Curbera, F., Publicación Novática / Upgrade nº 173, pp. 37- 41.
- Czarnecki y Helsen, 2003. Classification of model transformation approaches. Czarnecki, K., Helsen, S. En actas de Second Workshop on Generative Techniques in the context of Model Driven Architecture. Ed.: J. Bettin, G. van Emde Boas, A. Agrawal, E. Willink, J. Bezivin, October 2003.
- De Castro *et al.*, 2004. A User Services Oriented Method to model Web Information Systems. De Castro, V., Marcos, E., Cáceres, P. Actas de 5th International Conference on Web Information Systems Engineering (WISE 2004). Ed.: Zhou, X., Su, S., Papazoglou, M., Orlowska, M., Jeffery, K., Springer Verlag, LNCS 3306, pp. 41-52.
- De Castro *et al.*, 2006a. A Model Driven Method for Service Composition Modeling: A Case Study. De Castro, V., Marcos, E., López Sanz, M. International Journal of Web Engineering and Technology, Vol. 2(4), Eds: Inderscience Enterprise Ltd, pp. 335-353.
- De Castro *et al.*, 2006b. Construyendo Sistemas de Información Web Más Fáciles de Navegar: Un Estudio Experimental. De Castro, V., Genero, M., Marcos, E., Piattini, E.

Bibliografía

- Actas de 7th Argentine Symposium on Software Engineering (ASSE 2006), CD ASSE 2006, pp. 141-154.
- De Troyer y Leune, 1998. WSDM: A User Centered Design Method for Web Sites. De Troyer, O., Leune, C.J. Computer Networks and ISDN Systems 30(1-7), pp.85-94.
- De Troyer y Casteleyn, 2003. Modeling Complex Processes for Web Applications using WSDM. De Troyer, O., Casteleyn, S. En actas de 3rd International Workshop on Web Oriented Software Technology (IWWOST 2003), pp 1-12.
- Díaz *et al.*, 2001. Ariadne, a Development Method for Hypermedia. Díaz, P., Aedo, I. y Montero, S. Actas de DEXA 2001, pp. 764-774.
- Díaz *et al.*, 2005. Modelling Hypermedia and Web Applications: The Ariadne Development Method. Díaz, P., Montero, S., Aedo, I. Information Systems 30(8), pp. 649-673.
- Foster, 2005. Globus Toolkit Version 4: Software for Service-Oriented Systems. Foster, I. En actas de International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13.
- France y Bieman, 2001. Multi-view Software Evolution: A UML based Framework for Evolving Object-Oriented Software. France, R., Bieman, J. Actas de International Conference on Software Maintenance (ICSM 2001), IEEE Computer Society, pp. 386 - 396.
- Fraternali y Paolini, 1998. A Conceptual Model and a Tool Environment for Developing more Scalable, Dynamic and Customizable Web Applications. Fraternali, P., Paolini, P. Actas de EDBT'98, Ed.: Sheck, Saltor, Ramos, Alonso, Springer Verlag, LNCS 1377, pp. 21-435.
- Fraternali y Paolini, 2000. Model- Driven Development of Web Applications: The Autoweb System. Fraternali, P., Paolini, P. ACM Transactions on Information Systems, 28(4)
- Fraternali, 1999. Tools and approaches for developing data-intensive Web applications: a survey. Fraternali, P. ACM Computing Surveys 31(3), pp 227-263.
- French y Bell, 1996. Desarrollo organizacional (quinta edición). French, W.L., Bell, C.H. Jr. Editorial Prentice-Hall, Naucalpán de Juárez, México.
- Garzotto *et al.*, 1993. HDM- a Model-Based Approach to Hypertext Application Design. Garzotto, F., Paolini, P., Schwabe, D. ACM Transaction On Database Systems, 11(1), pp. 1-26.
- Ginige y Murugesan, 2001a. Web Engineering: An Introduction. Ginige A., Murugesan S. IEEE MultiMedia, 8 (1), pp. 14 -18.
- Ginige y Murugesan, 2001b. The essence of web engineering - managing the diversity and complexity of web application development. Ginige A., Murugesan S. IEEE MultiMedia, 8 (2), pp. 22 -25.
- Gómez *et al.*, 2000. Extending a conceptual modeling approach to web application design. Gómez, J., Cachero, C., Pastor O. En actas de Conference on Advanced Information Systems Engineering (CAISE' 00). Springer- Verlag, LNCS 1789, pp. 79-93.

- Gómez *et al.*, 2001. Conceptual Modeling of Device-Independent Web Applications. Gómez J., Cachero C., Pastor O. IEEE Multimedia, 8 (2), pp. 26-39.
- Gordijn y Akkermans, 2003. Value based requirements engineering: exploring innovative e-commerce idea. Gordijn, J., Akkermans, J.M. Requirements Engineering Journal 8(2). Springer Verlag, pp. 114 -134.
- Hernández *et al.*, 2006. A WEB-PACS for Multi-center Clinical Trials. Hernández, J.A., Acuña, C., de Castro, V., Marcos, E., López, M., Malpica, N. IEEE Transactions on Information Technology in Biomedicina, 11 (1), pp. 87-93.
- Harmon, 2004. The OMG's Model Driven Architecture and BPM. Harmon, P. Newsletter of Business Process Trends. Disponible en: <http://www.bptrends.com/publications.cfm>.
- Hennicker y Koch, 2000. A UML-based Methodology for Hypermedia Design. Hennicker, R., Koch, N. Actas de International Conference on the Unified Modeling Language (UML'2000), Springer Verlag, LNCS 1939, pp.410-424.
- IBM, 2001. Web Services Flow Language (WSFL 1.0). Disponible en: <http://www-306.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>
- IBM, 2005a. Web Services Coordination (WS-Coordination) 1.0. Disponible en: <http://www-106.ibm.com/developerworks/library/ws-coor>
- IBM, 2005b. Web Services Transaction (WS-Transaction) 1.0. Disponible en: <http://www-106.ibm.com/developerworks/library/ws-transpec>
- Isakowitz *et al.*, 1995. RMM: A Methodology for Structured Hypermedia Design. Isakowitz, T., Stohr, E.A.S., Balasubramanian, P. Communications of the ACM, 38(8), pp. 34-43.
- Isakowitz *et al.*, 1998. The Extended RMM Methodology for Web Publishing. Isakowitz, T., Kamis, A. y Koufaris, M. Working Paper IS-98-18, Center for Research on Information System. Disponible en: <http://rmm-java.stern.nyu.edu/rmm/>
- Jacobs, 2005. Enterprise software as service. Jacobs, D. ACM Queue 3(6), pp. 36-42.
- Jacobson *et al.*, 2001. El Proceso Unificado de Desarrollo de Software. Jacobson, I., Booch, G. y Rumbaugh J. Addison Wesley, 2001.
- Jouault y Kurtev, 2005. Transforming Models with ATL. Jouault, F., Kurtev, I. Actas de Model Transformations in Practice. Workshop realizado en conjunción con MoDELS 2005. Jamaica, 2005.
- Juristo y Moreno, 2001. Basics of Software Engineering Experimentation. Juristo, N., Moreno, A. Ed. Springer, 2001.
- Knapp *et al.*, 2004. Modeling Business Processes in Web Applications with ArgoUWE. Knapp, A., Koch, N., Zhang, G., Hassler H. Actas de International Conference on the Unified Modeling Language (UML'2004), Springer Verlag, LNCS 3273, pp. 69-83.
- Koch *et al.*, 2000. Extending UML to Model Navigation and Presentation in Web Applications. Koch, N., Baumeister, H. y Mandel, L. Actas de Modeling Web Applications, Workshop of the UML'2000, Ed.: Geri Winters y Jason Winters.

Bibliografía

- Koch *et al.*, 2004. Integration of Business Processes in Web Application Models. Koch, N., Kraus, A., Cachero, C. y Melia, S. Journal of Web Engineering 3(1), pp 22-49.
- Kock *et al.*, 1997. Can Action Research be Made More Rigorous in a Positivist Sense? The Contribution of an Iterative Approach. Kock, N., McQueen, R., Scott, J. Journal of Systems and Information Technology, 1 (1), pp. 1-24. Disponible en: http://carbon.cudenver.edu/~mryder/itc/act_res.html
- López *et al.*, 2006a. Service Oriented Method to Model Web Information Systems: a Case Study. López, G., De Castro, V., Marcos, E. Actas de IADIS International Conference WWW/INTERNET 2006, Vol II, Ed.: P. Isaías, M. Baptista Nunes, I.J. Martínez, 2006. IADIS Press, pp. 172-176, 2006.
- López *et al.*, 2006b. Implementation of Business Process requiring User Interaction. López, G., de Castro, V., Marcos, E. Actas de On the Move to Meaningfull Internet Systems 2006: OMT 2006 Workshops: En International Workshop on Agents, Web Services and Ontologies Merging (AWeSOMe'06). Eds.: R. Meersman, Z. Tari, P. Herrero *et al.*, Springer Verlag, LNCS 4277, pp. 107-115
- Lowe y Hall, 1999. Hypermedia and the Web: An Engineering Approach. Lowe, D., Hall, W. Editorial Wiley y Sons.
- Mayo Clinic, 2004. Analyze Software. Disponible en: <http://www.mayo.edu/bir/Software/Analyze/Analyze.html>
- Manolescu *et al.*, 2005. Model-Driven Design and Deployment of Service-Enabled Web Applications. Manolescu, I., Brambilla, M., Ceri, S., Comai, S., Fraternali, P. Transaction on Internet Technology 5 (2), pp. 439-479.
- Mantell, 2005. From UML to BPEL Model Driven Architecture in a Web services world. Mantell, K. Disponible en: <http://www-128.ibm.com/developerworks/webservices/library/ws-uml2bpel/>
- Marcos y Marcos, 1998. An Aristotelian Approach to the Methodological Research: a Method for Data Models Construction. Marcos, E. y Marcos, A. En Information Systems - The Next Generation. Ed.: L. Brooks y C. Kimble. Editorial: Mc Graw-Hill, pp. 532-543.
- Marcos *et al.*, 2002. MIDAS/BD: a Methodological Framework for Web Database Design. Marcos, E., Vela, B., Cavero, J.M. Conceptual Modeling for New Information Systems Technologies. Ed.: H. Arisawa, Y. Kambayashi, V. Kumar, H. C. Mayr, I. Hunt. Springer Verlag, LNCS-2465, pp. 227-238.
- Marcos *et al.*, 2003. Representing Web Services with UML: A Case Study. Marcos, E., De Castro, V., Vela, B. The First International Conference on Service Oriented Computing (ICSOC'03). Ed.: M.E. Orlowska, S. Weerawarana, M.P. Papazoglou, J. Yang. Springer Verlag, LNCS 2910, pp. 15-27.
- Marcos *et al.*, 2004. An approach for the Navigation Model Construction from the Use Case Model. Marcos, E, Cáceres, P., de Castro, V. Actas de The 16th Conference on

- Advanced Information Systems Engineering (CAISE'04 FORUM). Ed.: Grabis, J., Persson, A., Stirna, J., pp. 83-92.
- Marcos *et al.*, 2006. Integrating Software Architecture into a MDA Framework. Marcos, E., Acuña, C., Cuesta, C. Software Architecture, 3rd European Workshop, EWSA 2006, Springer Verlag, LNCS 4344, pp. 27-143.
- McTaggart, 1991. Principles of Participatory Action Research. McTaggart, R. Adult Education Quarterly 41(3).
- Mecca *et al.*, 1999. The ARANEUS guide to Web-site development. Mecca, G., Merialdo, P., Atzeni, P., Crescenzi, V., March. Disponible en: <http://poincare.inf.uniroma3.it/>
- Meliá *et al.*, 2005. MDA Transformations Applied to Web Application Development. Meliá, S., Kraus, A., Koch, N. En Internacional Conference of Web Engineering (ICWE 2005). Springer-Verlag, LNCS 3579, pp. 465-471.
- Microsoft, 2001. XLANG. Web Services for Business Process Design. Disponible en: http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm
- Microsoft, 2005. Microsoft .NET. Disponible en: <http://www.microsoft.com/net/>
- Miller y Mukerji, 2001. Model Driven Architecture. Ed.: Miller, J., Mukerji, J. Document number ormsc/2001-07-01. Disponible en: <http://www.omg.com/mda>
- Moreno *et al.*, 2006. A UML 2.0 Profile for WebML Modeling. Moreno, N., Fraternalli, P., Vallecillo, A. Actas de Workshop on Model Driven Web Engineering (MDWE 2006), realizado en conjunción con ICWE 2006, Palo Alto (California). Disponible en: <http://www.lcc.uma.es/~av/mdwe2006/Agenda.html>.
- Muller *et al.*, 2005. Platform Independent Web Application Modeling and Development with Netsilon. Muller, P., Studer, P., Fondement, F. y Bezivin, J. Software and Systems Modeling 4(4), Springer Verlag, pp. 424 -442.
- Muñoz, 2004. Una Gramática de Grafos para la Transformación de Relaciones de Asociación desde Modelos de Análisis hacia Modelos de Diseño. Muñoz, J. Technical Report, DSIC-UPV 2004.
- Nielsen, 1999. Designing Web Usability: The Practice of Simplicity. Nielsen, J. Editorial New Riders Press.
- OMG, 2001. MDA Guide V1.0.1. Miller, J., Mukerji, J. (eds.). Document N° omg/2003-06-01. Disponible en: <http://www.omg.org/cgi-bin/doc?omg/03-06-01>
- OMG, 2003a. OMG Unified Modeling Language Specification. Version 1.5. Disponible en: <http://www.omg.org/technology/documents/formal/uml.htm>
- OMG, 2003b. UML 2.0 OCL Specification. OMG Final adopted Specification. Disponible en: <http://www.omg.org/docs/ptc/03-10-14.pdf>
- OMG, 2003c. UML Superstructure 2.0. OMG Adopted Specification ptc/03-08-02. Disponible en: <http://www.uml.org/>
- OMG, 2005. Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification Final Adopted Specification ptc/05-11-01. Disponible en: <http://www.omg.org/>

Bibliografía

- Oosterwijk, 2003. The DICOM Standard, Overview and Characteristics. Oosterwijk H. Disponible en: http://www.ringholm.de/docs/02010_en.htm
- OSGA, 2003. Open Grid Services Architecture. Disponible en: <http://forge.gridforum.org/projects/ogsa-wg/document/draft-ggf-ogsa-spec/en/23>
- Ouyang *et al.*, 2006. Translating Standard Process Models to BPEL. Ouyang, Ch., Dumas, M., Breutel, S., H. M. ter Hofstede, A. Actas de 18th International Conference on Advanced Information Systems Engineering (CAISE'06). Ed.: E. Dubois, K. Pohl. Springer Verlag, LNCS 4001, pp. 417-432.
- Overmyer, 2000. What's Different about Requirements Engineering for Web Sites? Overmyer, S. P. Requirements Engineering 5(1), pp. 62-65.
- Papazoglou y Georgakopoulos, 2003. Serviced-Oriented Computing. Papazoglou, M.P. y Georgakopoulos, D. Communications of ACM 46(10), pp. 25-28.
- Papazoglou *et al.*, 2006. Service-Oriented Computing. Research Roadmap. Papazoglou, M. Traverso, P., Dustdar, S., Leymann, F. Disponible en: ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/st-ds/services-research-roadmap_en.pdf
- Parikh y Pradhan, 2003. Modeling Augmented Web Services. Parikh, A., Pradhan, R. Java Skyline Magazine for Java Developers, Mayo de 2003. Disponible en: <http://www.javaskyline.com/WebServicesModeling.html>
- Pastor *et al.*, 2001. An Object-Oriented Approach to Automate Web Applications Development. Pastor, O., Abrahão, S., Fons, J. En actas de internacional Conference of Electronic Commerce and Web Technologies (EC-Web2001). Springer Verlag, LNCS 2115, pp. 16-28.
- Paterno *et al.*, 1997. ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. Paternò, F., Mancini, C., Meniconi, S. Actas de Human-Computer Interaction (INTERACT '97), Chapman & Hall, pp. 362-366.
- Pressman, 2005. Software Engineering. A Practitioner's Approach. R. Pressman. McGraw-Hill International Edition. 6th Edition.
- Queralt *et al.*, 2006. Un motor de transformación de modelos con soporte para el lenguaje QVT Relations. Queralt, P., Hoyos, L., Boronat, A., Carsí, J.A, Ramos, I. Actas de Taller de Desarrollo de Software Dirigido por Modelos (DSDM'06), realizado en conjunction con JISBD'06. Barcelona (España), Octubre, 2006. Disponible en: <http://www.dsic.upv.es/workshops/dsdm06/programa.php>
- QVT Partners, 2003. Initial Submission for MOF 2.0 Query/View/Transformations RFP, QVTPartners, Disponible en: <http://qvtp.org/downloads/1.1/qvtpartners1.1.pdf>
- Schmid y Donnerhak, 2005. OOHMDA – An MDA Approach for OOHDM. Hans Schmid, H.A., Donnerhak, O. En actas de Internacional Conference of Web Engineering (ICWE 2005). Springer-Verlag, LNCS 3579, pp. 569-574.
- Schmid y Rossi, 2004. Modeling and Designing Processes in E-Commerce Applications. Schmid, H.A., Rossi, G. IEEE Internet Computing 8(1), pp. 19-27.

- Schwabe *et al.*, 2001. Engineering Web Applications for Reuse. Schwabe, D. Esmeraldo, L., Rossi, G., Lyardet, F. IEEE MultiMedia 8(1), pp. 20-31.
- Schwabe y Rossi, 1995. The Object-Oriented Hypermedia Design Model. Schwabe, D., Rossi, G. Communications of ACM 38(8), pp. 45-46.
- Schwabe y Rossi, 1998. An object oriented approach to web-based applications design. Schwabe, D., Rossi, G. Theory and practice of object system 4(4), pp. 207-225.
- Sendall y Kozaczynski, 2003. Model Transformation—the Heart and Soul of Model-Driven Software Development. Sendall, S., Kozaczynski, W. IEEE Software archive 20(5), pp. 42-45.
- ModelWare Project, 2006. SmartQVT: An open source model transformation tool implementing the MOF 2.0 QVT-Operational language. Disponible en: <http://smartqvt.elibel.tm.fr/>
- Sommerville, 2000. Software Engineering. Ian Sommerville. Addison Wesley. 6ª Edición.
- Sun Microsystems, 2004. Java 2 Platform Enterprise Edition (J2EE). Disponible en: <http://java.sun.com/j2ee/>
- Susman y Evered, 1978. An Assesment of the Scientific Merits of Action Research. Susman, G, Evered, R. Administrative Science Quarterly 23(4), pp. 582-603.
- Taentzer, 2000. AGG: A Tool environment for Algebraic Graph Transformation. Taentzer, G. Applications of Graph Transformations with Industrial Relevance. Nagl, M., Schürr, A., Münch, M. (Eds). Springer Verlag, LNCS 1779, pp. 481-488.
- Tongrungrrojana y Lowe, 2004. WIED: A Web Modelling Language for Modelling Architectural-Level Information Flows. Tongrungrrojana, R., Lowe, D. Journal of Digital Information 5(2).
- Tratt, 2005. Model transformations and tool integration. Tratt, L. Software and Systems Modeling 4(2), Mayo 2005, pp. 112-122.
- Turk *et al.*, 2002. Model-Driven Approaches to Software Development. Turk, D., France, R., Rumpe, B., Georg, G. En actas de Advances in Object-Oriented Information Systems (OOIS 2002). Springer Verlag, LNCS 2426, pp. 229-230.
- Valderas *et al.*, 2005. From Web Requirements to Navigational Design - A Transformational Approach. Valderas, p., Fons, J., Pelechano. V. En actas de Internacional Conference of Web Engineering (ICWE 2005). Springer Verlag, LNCS 3579, pp. 506-511.
- Vela *et al.*, 2004. A Model Driven Approach for XML Database Development. Vela, B., Acuña, C. y Marcos, E. 23rd. International Conference on Conceptual Modelling (ER 2004). Ed.: P. Atzeni, W. Chu, H. Lu, S. Zhou, T.W Ling. Springer Verlag, LNCS 3288, pp. 780-794.
- Verner, 2004. BPM The Promise and the Challenge. Verner, L. Queue of ACM 2(4), pp. 82-91.
- W3C, 1999a. XML Path Language (XPath) Version 1.0, W3C Recommendation. J. Clark, S. DeRose (Eds.). Disponible en: <http://www.w3.org/TR/xpath>

Bibliografía

- W3C, 1999b. XSL Transformations (XSLT) Version 1.0. W3C Recommendation, J. Clark (Eds.). Disponible en: <http://www.w3.org/TR/xslt>
- W3C, 2002. Web Services Choreography. W3C Working Group. Disponible en: <http://www.w3.org/2002/ws/chor/>
- W3C, 2003a. W3C Glossary and Dictionary Project. Disponible en: <http://www.w3.org/2003/glossary/>
- W3C, 2003b. Web Services Description Language (WSDL) Version 1.2, W3C Working Draft 3 March 2003. Chinnici, R., Gudgin, M., Moreau, J., Weerawarana, S. (Eds). Disponible en: <http://www.w3.org/TR/2003/WD-wsdl12-20030303/>
- W3C, 2004a. Web Services Description Language (WSDL) Version 2.0 Core Language, W3C Working Draft 26 March 2004. Chinnici, R., Gudgin, M., Moreau, J., Schlimmer, J., Weerawarana, S. (Eds). Disponible en: <http://www.w3.org/TR/2004/WD-wsdl20-20040326/>
- W3C, 2004b. XML Schema Working Group. XML Schema Parts 0-2: (Primer, Structures, Datatypes). W3C Recommendation. Disponible en: <http://www.w3.org/TR/xmlschema-0/>, <http://www.w3.org/TR/xmlschema-1/> y <http://www.w3.org/TR/xmlschema-2/>
- Wadsworth, 1998. What is Participatory Action Research?. Wadsworth, Y. Action Research International. Disponible en: <http://www.scu.edu.au/schools/gcm/ar/ari/pywadsworth98.html>
- Yang y Papazoglou, 2004. Service Components for managing the life-cycle of the service composition. Yang, J., Papazoglou, M.P. Information Systems 29, pp. 97-125.
- Yin, 2003. Case Study Research Design and Methods. Yin, R.K. Sage Publications. 3º Edición.

LUGARES DE INTERNET

Business Process Management Initiative. <http://www.bpmi.org/>
ConfMaster. <http://www.confmaster.net/>
GesIMED System. <http://ariadna.escet.urjc.es/gesimed/>
Java 2 Platform Enterprise. <http://java.sun.com/j2ee/>, 2004
Microsoft .NET. <http://www.microsoft.com/net/default.mspx>
Model Driven Architecture. <http://www.omg.org/mda/>
Object Management Group. <http://www.omg.org/>
PARBUD. <http://parbud.escet.urjc.es/>
Unified Modeling Language. <http://www.uml.org/>
VisualWADE. <http://www.visualwade.com/>
WebConference. <http://kybele.escet.urjc.es/webconference/>
WebRatio. <http://www.webratio.com>

APÉNDICE A: SIGLAS

ACM	Association for Computing Machinery
ADM	Ariadna Development Method
ADVs	Abstract Data Views
AECI	Agencia Española de Cooperación Internacional
AGG	Attributed Graph Grammar
API	Application Programming Interface
AS	Actividad de Servicio
ATL	Atlas Transformation Language
AutoWeb	Metodología para el desarrollo de aplicaciones Web
B2B	Bussines to Bussines
BD	Base de Datos
BPD	Business Process Diagram
BPEL	Business Process Execution Language
BPEL4WS	BPEL for Web Services
BPMN	Business Process Modeling Notation
CAM	Comunidad de Madrid
CIM	Computational Independent Model
CUB	Caso de Uso Básico
CUC	Caso de Uso Compuesto
DAWIS	Digital Archives Web Integrated System
DB	Data Base
DGT	Dirección General del Tráfico
DICOM	Digital Imaging and Communications in Medicine
DSL	Domain Specific Language
EDAD	Entorno para el Desarrollo e integración automática de Archivos Digitales
ER	Entidad/Interrelación
ERCIM	European Research Consortium for Informatics and Mathematics
ERMM	Extended - Relationship Management Methodology
FS	Functional Slice
GTEBIM	Grupo de Tecnología Electrónica, Bioingeniería e Imagen Médica

HDM	Hypertext Design Method
HTTP	HyperText Transfer Protocol
ICM	Informática de la Comunidad de Madrid
ICWE	International Conference on Web Engineering
IEEE	Institute of Electrical and Electronics Engineers
ISBN	International Standard Book Number
ISSN	International Standard Serial Number
J2EE	Java Platform Enterprise Edition
JISBD	Jornadas de Ingeniería del Software y Bases de Datos
LAIM	Laboratorio de Análisis de Imágenes Médicas
LHS	Left Hand Side
LNCS	Lectura Notes in Computer Sciences
MCYT	Ministerio de Ciencia y Tecnología
MDA	Model Driven Architecture
MDD	Model Driven Development
MDWSD	Model Driven Web Service Development
MIDAS	Metodología para el Desarrollo de Aplicaciones Web
MIME	Multipurpose Internet Mail Extensions
MSS	Message
.NET	Plataforma de desarrollo de software de Microsoft
OGSA	Open Grid Services Architecture
OMG	Object Management Group
OO	Object Oriented - Orientado a Objeto
OO-H	Object Oriented Hypermedia Method
OOHDM	Object Oriented Hypermedia Design Method
OOWS	Object Oriented Web-Solutions
OP	Operation
OS	Orientado a Servicio
PACS	Picture Archiving and Communication Systems
PD-PSM	Platform Dependent - Platform Specific Model
PIM	Platform Independent Model
PIGE	Proyecto de Investigación GEneral
PPR	Proyecto Producción Responsable

PSM	Platform Specific Model
QVT	Query View Transformation
QVT-P	Query View Transformation - Partners
RHS	Right Hand Side
RMM	Relationship Management Methodology
SACim	Servicio de Almacenamiento y Consulta de imágenes médicas
SDL	Specification and Description Language
SIW	Sistema de Información Web
SN	Servicio de Negocio
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOC	Service Oriented Computing
SOD-M	Service Oriented Development Method
SPim	Servicio de Procesamiento de imágenes médicas
SS	Structural Slice
SVim	Servicio de Visualización de imágenes médicas
TD-PSM	Technology Dependent - Platform Specific Model
TIC	Tecnologías de la Información y de las Comunicaciones
TIN	Tecnologías Informáticas
UDDI	Universal Description, Discovery, and Integration
UID	User Interaction Diagram
UML	Unified Modeling Language
URJC	Universidad Rey Juan Carlos
UT	Universidad de Twente
UTN	Universidad Tecnológica Nacional
UWE	UML Based Web Engineering
W2000	Marco para el diseño de aplicaciones Web
W3C	World Wide Web Consortium
WebML	Web Modeling Language
WIED	Web Information Exchanged Diagram
WIS	Web Information System
WS	Web Service
WSDL	Web Service Description Language
WSDM	Web Site Design Method

Apéndice A: Siglas

WSFL	Web Services Flow Language
WWW	World Wide Web
XMI	XML Metadata Interchange
XML	Extensible Markup Language
XPath	XML Path
XSD	XML Schema Definition
XSL	Extensible Stylesheet Lang
XSLT	Extensible Stylesheet Lang Transformation

APÉNDICE B: CASOS DE ESTUDIO

En este apéndice se incluyen los modelos de los distintos casos de estudio de laboratorios y reales realizados para la validación de MIDAS.

Durante el desarrollo del trabajo de investigación que se presenta en esta Tesis Doctoral se desarrollaron *dos casos de estudio de laboratorio* (un Servicio Web de información de vuelos y el SIW para la gestión de envío de trabajos a conferencias), que han sido diseñados específicamente para investigar necesidades en el desarrollo orientado a servicios del comportamiento de SIW. Además, se han realizado otros *cuatro casos de estudio reales* (GesIMED, PARBUD, ATENTO e ISLA), los cuales han permitido ir detectando nuevas necesidades a la vez que se iban validando las primeras versiones del método propuesto.

En caso de estudio del SIW GesIMED se ha presentado ya de manera detallada en la sección 4.3; en este apéndice se presentan los restantes casos de estudio.

Caso O: Servicio Web de Información de Vuelos

Este caso consistió en el modelado y desarrollo de una interfaz de un servicio Web para proporcionar información acerca de los vuelos de un aeropuerto, y se desarrollo utilizando una versión anterior de estándar WSDL (W3C, 2003b).

La Figura B.1 muestra el modelo de interfaz de servicio Web realizado para el servicio Web objeto del caso de estudio, llamado "ServicioVuelo", y la Figura B.2 muestra el código WSDL correspondiente al servicio Web desarrollado.

Como puede verse en la Figura B.1 el servicio Web define dos operaciones "GetInfoVuelo" y "CheckIn". La operación "GetInfoVuelo" contiene dos mensajes, uno de entrada y otro de salida. El mensaje de entrada "GetInfoVueloInput" contiene dos parámetros o partes "AerolineaNom" y "VueloNum", los cuales utilizan como tipo de dato un tipo base XSD, una cadena de caracteres (string) y un entero (int), respectivamente. El mensaje de salida "GetInfoVueloOutput" contiene un parámetro "InfoVuelo" que utiliza como tipo de dato el elemento "InfoVueloType", este último está asociado a través de un atributo *type*. La operación "CheckIn" contiene solo un mensaje de entrada "CheckInInput". Dicho mensaje contiene un solo parámetro "Cuerpo" que utiliza como tipo de dato el elemento "Ticket", que está asociado a través de un atributo *element*.

El tipo de puerto "ServicioAeropPortType" agrupa las operaciones que serán realizadas por el servicio. El elemento "ServiceAeropBinding" describe el enlace entre dicho tipo de puerto y el protocolo SOAP. El Servicio contiene un solo puerto "ServicioVueloPort", el cual define el punto concreto de acceso a través de una URL, que indica la localización del servicio. El elemento "ServiceAeropBinding", se asocia a un puerto, y describe la manera de interactuar con el servicio Web.

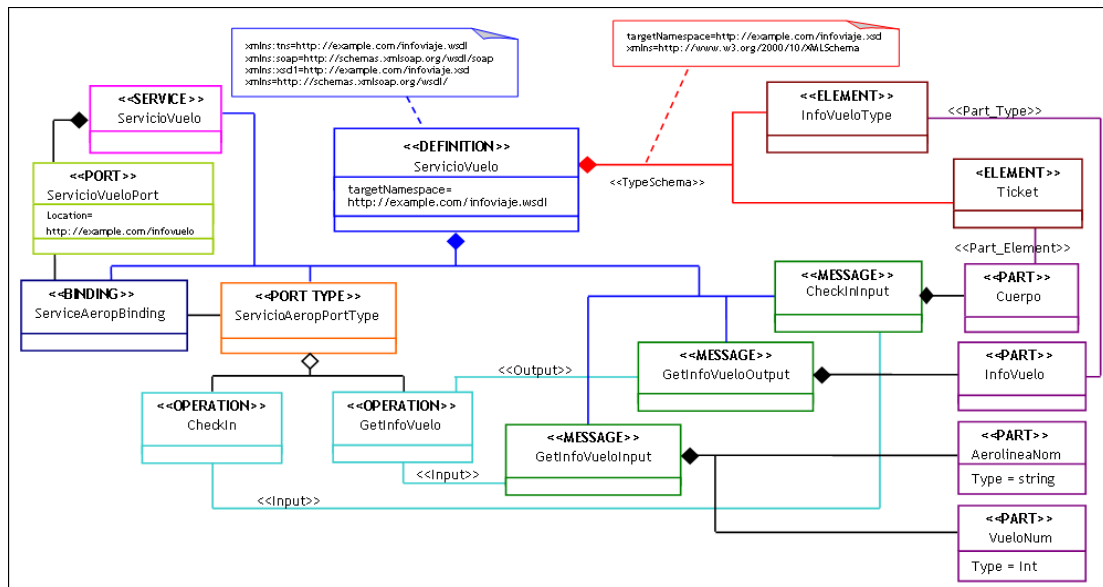


Figura B.1. Modelo de interfaz del servicio Web “ServicioVuelo”



Figura B.2. Representación de la interfaz del servicio Web “ServicioVuelo” en WSDL 1.2

Caso 1: WebConference: Sistema para la Gestión del Envío de Trabajos a Conferencias

Este caso consistió en el desarrollo de un SIW para la gestión y envío de trabajos a conferencias. El SIW WebConference está basado en otro sistema para el envío de conferencias, llamado ConfMaster (<http://www.confmaster.net/>) a partir del cual se han detectado los servicios que se deben ofrecer a los usuarios.

En este apartado se presenta el diseño del SIW, que se ha llevado a cabo aplicando una primera versión de los modelos propuestos a nivel PIM para el modelado del aspecto del comportamiento, los cuales se ha presentado en esta Tesis Doctoral, y realizando, además, la integración con los modelos para el aspecto del hipertexto definidos en MIDAS.

El SIW que se ha realizado ofrece tres servicios específicos a los usuarios, en este caso, los autores:

- *Envío de artículo:* para llevar a cabo este servicio es necesario, indicar los datos del artículo a enviar (título, resumen, autores, etc.), y posteriormente, realizar el envío del trabajo a través de la aplicación.
- *Edición de los datos de autor:* en este caso el autor puede visualizar sus datos y realizar las modificaciones que desee.
- *Visualización de artículos enviados por un autor:* contempla la visualización por parte del autor, de los diferentes artículos enviados a una conferencia, así como la posibilidad de actualización de los datos de éstos, tales como título, resumen, autores, etc.

La Figura B.3 muestra el **modelo de casos de uso** realizado para este caso. En el se representa como un actor al "Autor", que es el consumidor del final de los servicios que se implementarán en el mismo. Además, en el modelo se representan, como casos de uso, los distintos servicios que se ofrecen a los autores: "enviar un artículo", "editar los datos del autor" y "ver los artículos enviados". Para llevar a cabo estas tareas, un autor debe siempre identificarse, indicando nombre de usuario y contraseña. Además, el sistema ofrece la posibilidad de consultar su contraseña en caso de olvido.

La Figura B.4 muestra el **modelo de casos de uso extendido** para este caso, en el se representan las funcionalidades requeridas por el sistema para llevar a cabo los servicios de negocio. Así, por ejemplo, para el servicio de negocio "enviar un artículo", se han identificado las siguientes funcionalidades: "identificarse como autor", para lo que deberá primero "registrarse como autor" u,

opcionalmente, indicar si “olvidó su password”; “registrar los datos del artículo”; “ver los datos del artículo introducido”; y, “enviar el artículo”. Además, como puede verse en la Figura B.4, los casos de uso básicos del modelo se han representado como funcionales (utilizando el estereotipo <<FB>>, de *Functional Basic use case*) o estructurales (utilizando el estereotipo <<SB>>, de *Structural Basic use case*), puesto que es una tarea necesaria para la construcción del modelo de fragmentos extendido.

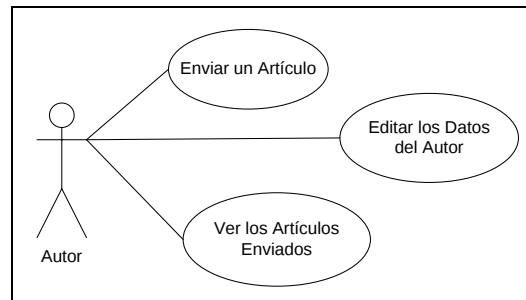


Figura B.3. Modelo de casos de uso del SIW WebConference

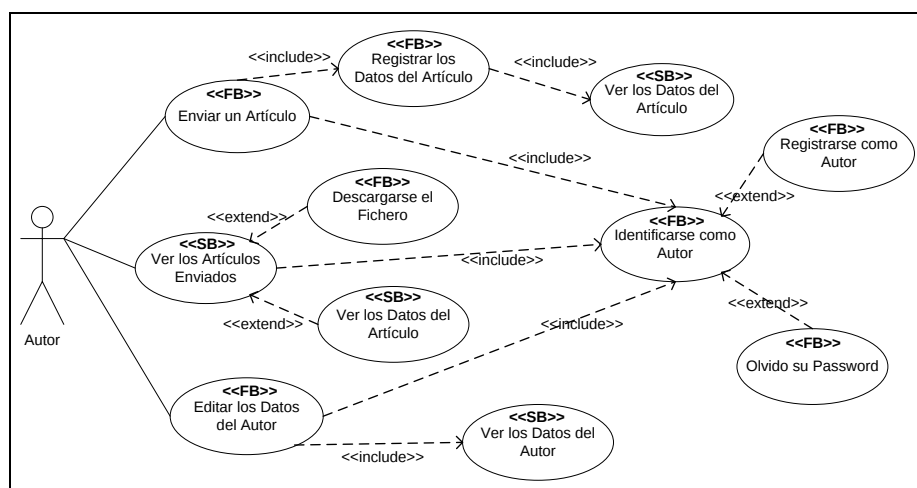


Figura B.4. Modelo de casos de uso extendido del SIW WebConference

La Figura B.5 muestra el **modelo de proceso de servicio** para cada uno de los servicios de negocio identificados para el SIW WebConference. Dicho modelo muestra el flujo de trabajo necesario para la realización de cada servicio de negocio que se implementará en el SIW. Así, por ejemplo, si nos centramos en el servicio de negocio “enviar un artículo”, puede verse que el autor deberá primero identificarse como autor, y luego de ello, deberá registrar los datos del artículo en primer lugar, luego ver los datos del artículo que ha introducido, y finalmente, enviar el artículo.

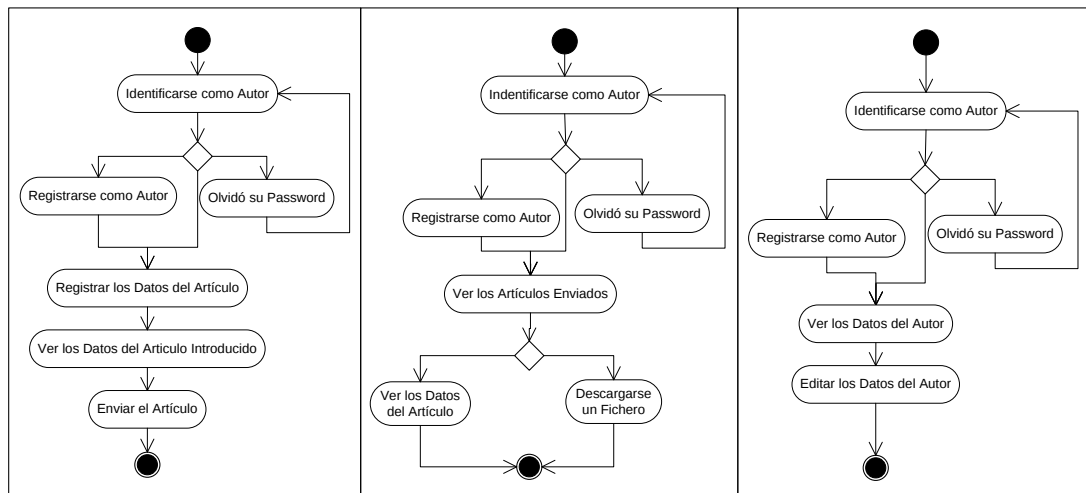


Figura B.5. Modelo de proceso de servicio para los servicios de negocio “enviar un artículo”, “editar los datos del autor” y “ver los artículos enviados”

Las siguientes figuras muestran los modelos realizados para el desarrollo del hipertexto del SIW.

La Figura B.6 muestra en **modelo de fragmentos extendido**, que se ha construido siguiendo las guías descritas en el apartado 3.7.2 de la sección 3.7. En el modelo se han representado: los fragmentos, tanto estructurales como funcionales, obtenidos a partir del modelo de casos de uso extendido (ver Figura B.4); y las rutas, que muestran la secuencia de pasos que el usuario debe seguir para llevar a cabo con éxito cada uno de los servicios, las cuales se han obtenido a partir del modelo de proceso de servicio (ver Figura B.5).

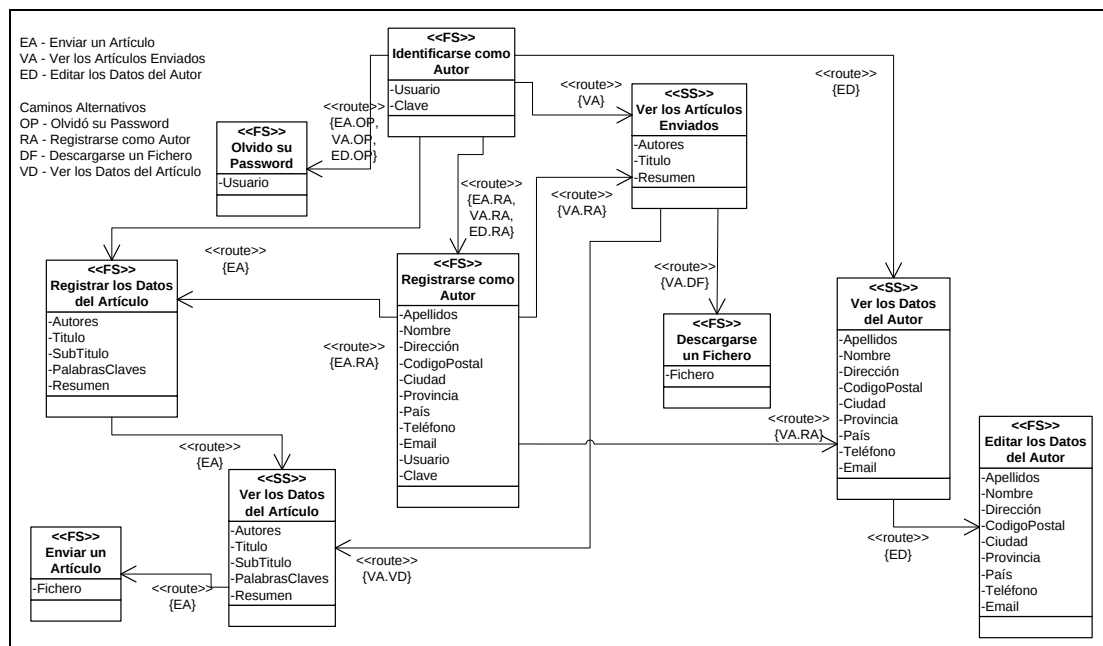


Figura B.6. Modelo de fragmentos extendido para del SIW WebConference

La Figura B.7 muestra el **modelo de navegación extendido** que se obtiene añadiendo las estructuras de navegación (*menu*, *index*, *query* y *guide tours*) al modelo de fragmentos extendido descrito previamente. Como puede verse en la Figura B.7, siguiendo con las guías definidas el método para el modelado del hipertexto de MIDAS, se ha introducido un menú principal en el que se represente una entrada por cada uno de los servicios de negocio identificado en el SIW.

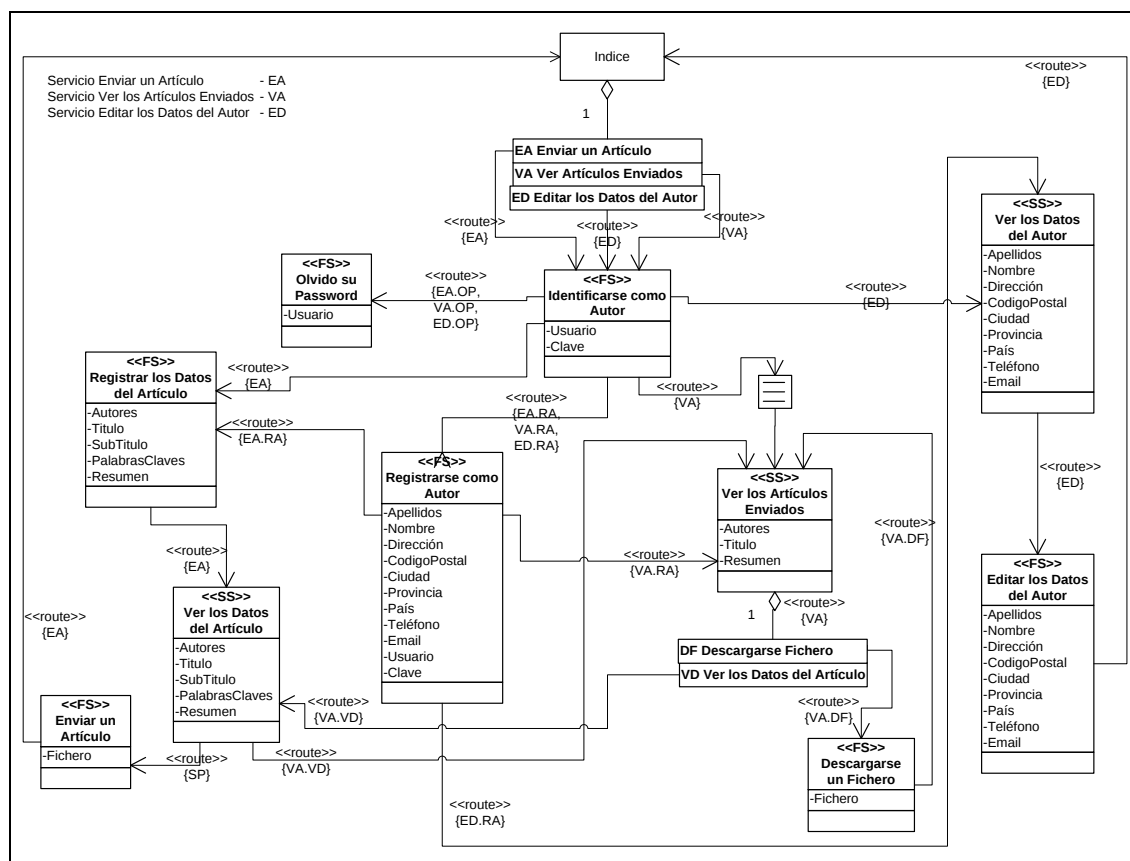


Figura B.7. Modelo de navegación extendido para del SIW WebConference

Caso 3: PARBUD: Sistema para la Gestión de Presupuestos Participativos

Este caso se ha abordado el desarrollo de un SIW para la gestión de presupuestos participativos. La idea de presupuestos participativos consiste en permitir a los ciudadanos participar en la decisión de asignación de un presupuesto. Así por ejemplo, los miembros de un barrio podrían decidir cómo y en qué invertir parte del presupuesto que el ayuntamiento tiene para el desarrollo del barrio.

Este caso se llevó a cabo aplicando las técnicas y modelos que se habían propuesto hasta el momento, con el objetivo de detectar posibles mejoras o nuevas

necesidades de un dominio particular como es el desarrollo de aplicaciones de democracia electrónica.

Para el diseño de esta aplicación se hizo necesaria la definición del modelo de proceso de negocio, puesto que permitía comprender y determinar explícitamente, que pasos o etapas sería necesario seguir para completar el ciclo de un presupuesto participativo.

La Figura B.8 muestra el **modelo de proceso de negocio** desarrollado para este caso. Dicho modelo se ha obtenido identificando las diferentes etapas del negocio que conforman el proceso para la elaboración de un presupuesto participativo. El proceso para la elaboración de un presupuesto participativo incluye las siguientes etapas: (a) preparación, en la que se establecen cuestiones como dinero que se asignará, quienes decidirán, alternativas de asignación, etc.; (b) comunicación de preferencias, en la que los participantes indican sus preferencias de asignación del presupuesto; (c) negociación, que se realiza en caso de no haber una propuesta de asignación que satisfaga a los distintos participantes, y se realiza a través un método de negociación determinado por el mediador; (d) votación, que se realiza en caso de no llegar a un acuerdo a través de la negociación, y en la que cada participante vota por su preferencia; y, finalmente, se incluye una etapa de (e) pos-acuerdo, en la que se analiza en qué medida la asignación de presupuesto acordada satisface al conjunto de participantes.

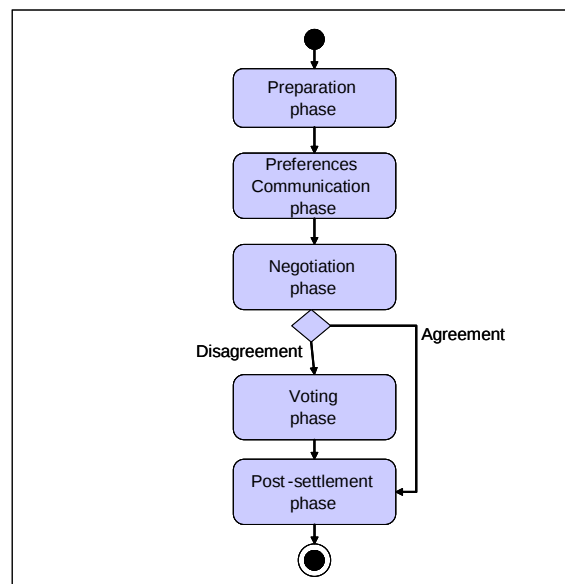


Figura B.8. Modelo de proceso de negocio para la elaboración de presupuestos participativos

En este caso, la representación del modelo de proceso de negocio permite dar a conocer, tanto a los desarrolladores como a los usuarios del SIW, los pasos

que son necesarios llevar a cabo para completar el ciclo de elaboración de un presupuesto participativo.

En el resto de este apartado se muestran algunos de los modelos desarrollados para la primera etapa del diseño de este SIW, la etapa *comunicación de preferencias*. En esta etapa se contemplan dos tipos de usuarios o consumidores de los servicios del SIW: participante y mediador. Los servicios ofrecidos por el SIW a dichos usuarios son:

- *Solicitud de participación en presupuestos*: ofrece a los participantes la posibilidad de participar en la asignación de un determinado presupuesto, debiendo ingresar para ello, sus datos personales y de contacto.
- *Autorización de la participación en presupuestos*: ofrece a los mediadores la posibilidad de gestionar los distintos miembros que participan en una asignación de presupuesto, autorizando o no el acceso a los mismos.
- *Comunicación de preferencias*: ofrece a los participantes la posibilidad de comunicar sus preferencias entre las distintas alternativas de asignación de un presupuesto.
- *Visualizar presupuestos de un participante*: ofrece a los participantes la posibilidad de consultar los presupuestos en los que esté participando.

La Figura B.9 muestra el **modelo de casos de uso** en el que se representan los servicios de negocio contemplados en la etapa de comunicación de preferencias. Como se ha dicho anteriormente, en esta etapa se contemplan dos tipos de usuarios o consumidores de los servicios del SIW, participante y mediador, los cuales se han representado como actores en el modelo. Cada uno de los consumidores se muestra relacionado con los servicios que consume.

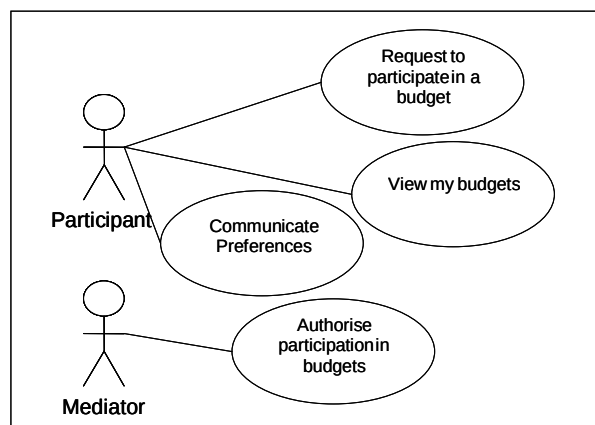


Figura B.9. Modelo de casos de uso para la etapa de “comunicación de preferencias” del SIW PARBUD

La Figura B.10 muestra, de manera parcial, el **modelo de fragmentos extendido**; en el se representa los fragmentos y las rutas de navegación necesarias para llevar a cabo el servicio de negocio “comunicar preferencias”. La Figura B.11 muestra el **modelo de navegación extendido** que se obtiene añadiendo las estructuras de navegación al modelo de fragmentos extendido de la Figura B.10.

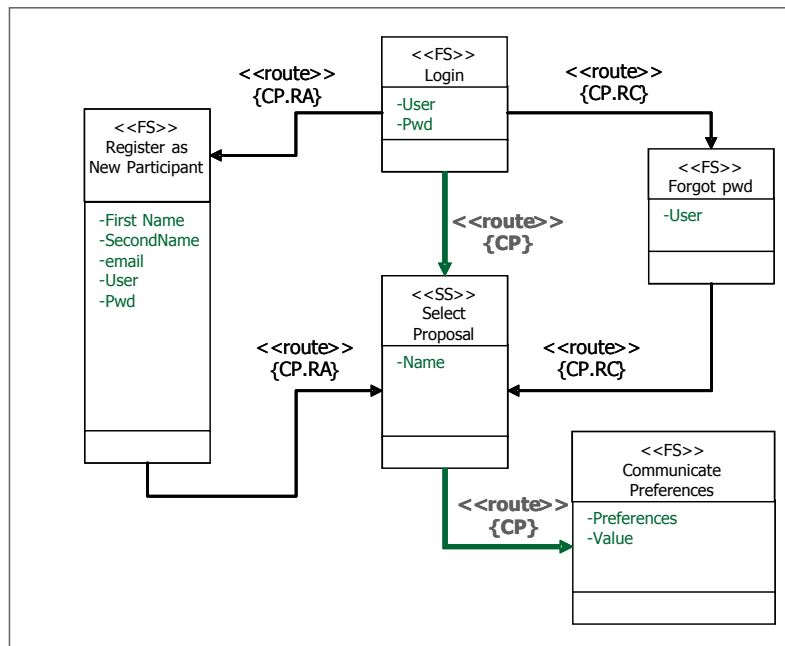


Figura B.10. Modelo de fragmentos extendido para el servicio de negocio “comunicación de preferencias” del SIW PARBUD

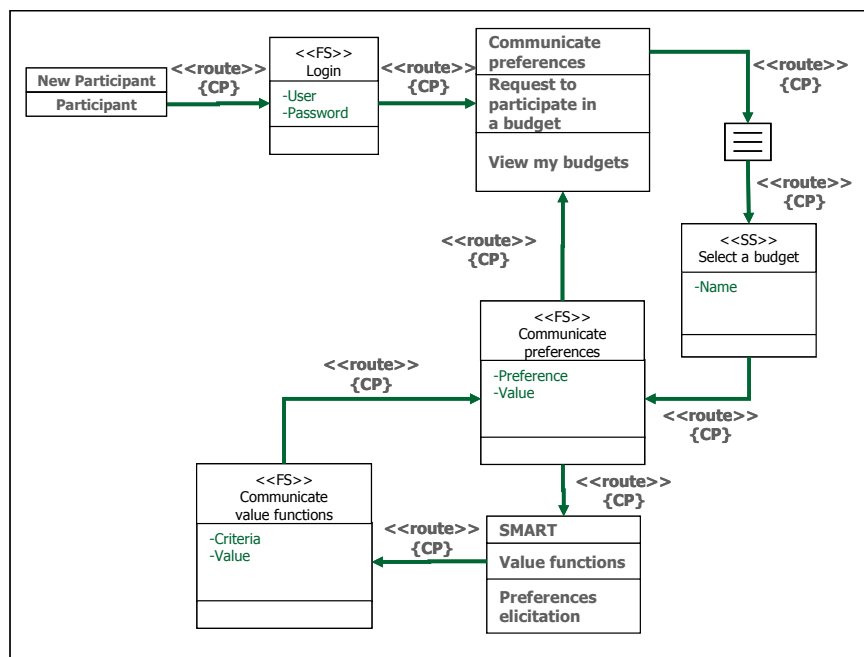


Figura B.11. Modelo de navegación extendido para el servicio de negocio “comunicación de preferencias” del SIW PARDUB

Caso 4: Atento: Sistema de Ventanilla Única para la Administración Pública

En este caso se ha abordado el desarrollo del SIW Atento. Este SIW se ha diseñado utilizando tanto los modelos definidos en SOD-M como los modelos del aspecto del hipertexto. En la actualidad está SIW está siendo implementado.

Atento es un sistema para la gestión de las relaciones del ciudadano con la administración pública que permite acceder, a través de la intranet de la administración, a la información sobre las peticiones, y sus correspondientes respuestas, que se han producido entre un ciudadano y la administración, sea cual sea su canal de entrada (telefónico, presencial, Web, e-mail) o el área de la administración que la haya atendido. Un ejemplo de relación entre el ciudadano y la administración es la solicitud y emisión de un certificado oficial o la consulta sobre los trámites a seguir para obtener una determinada prestación de la administración. Esta información sobre relaciones entre ciudadanos y administración se encuentra almacenada en el "sistema de gestión de las relaciones con el ciudadano" que es externo a Atento y que además de gestionar estas relaciones, puede acceder a información útil durante el proceso de atención y que se encuentra almacenada en otras bases de datos corporativas. Este otro sistema es utilizado por los Agentes de Atención Telefónica. Atento tiene una funcionalidad más limitada que el "sistema de gestión de las relaciones con el ciudadano". La Figura B.12 muestra el esquema de la arquitectura del SIW Atento y sus relaciones con otros sistemas externos.

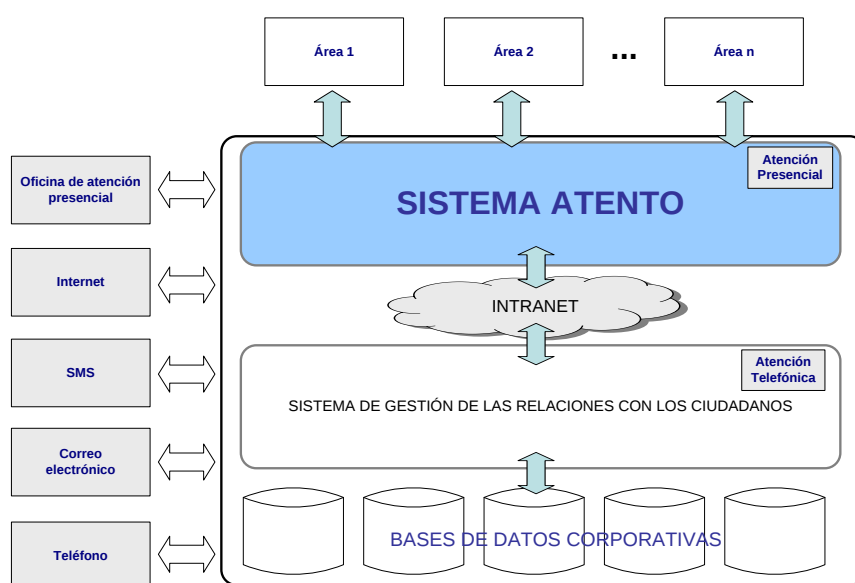


Figura B.12. Arquitectura del sistema Atento

Los usuarios del sistema Atento son los agentes de atención presencial (empleado público) los cuales pueden acceder, a través de la Intranet de la entidad pública, a los siguientes servicios:

- *Editar datos de un ciudadano:* ofrece a los usuarios de Atento la posibilidad de editar cualquiera de los datos de un ciudadano, como por ejemplo un actualizar un cambio de domicilio. Para ello, el usuario deberá en primer lugar buscar al ciudadano en el sistema y, una vez seleccionado podrá modificar los datos de contacto del ciudadano.
- *Grabar interacción de un ciudadano:* para ello, el usuario también deberá buscar al ciudadano en el sistema y después podrá indicar que se inicia la interacción y el sistema se encargará de crear una nueva interacción asociada a ese ciudadano, indicando datos como el código del usuario que lo está atendiendo, el estado inicial, la fecha de apertura, etc. Una interacción de un ciudadano podría ser por ejemplo, una solicitud de un informe de vida laboral.
- *Consultar interacciones de ciudadanos:* ofrece a los usuarios la posibilidad de consultar los datos de una interacción existente. Para ello, el usuario deberá buscar la interacción existente aportando alguno de los datos que la identifican tales como, número de la interacción, nombre del ciudadano al que pertenece, fecha de apertura, etc.
- *Grabar encuesta de un ciudadano:* ofrece la posibilidad de realizar una encuesta de satisfacción al ciudadano para preguntarle sobre el proceso de atención. Para ello, el usuario deberá localizar al ciudadano en la base de datos para después grabar una nueva encuesta asociada a el.

La Figura B.13 muestra el **modelo de casos de uso** realizado para este caso. Como ya se ha comentado, Atento es un SIW orientado al uso por el personal de la administración pública y que permite acceder a la información de un sistema externo denominado Sistema CRM. Por lo tanto, este SIW tiene un solo consumidor de los servicios del mismo, el Agente de Atención Presencial, que será el usuario que interactuará con el sistema Atento. Los servicios de negocio que serán implementados en el SIW, y que se muestran en el modelo de casos de uso como casos de uso, son los cuatro descritos previamente: "editar datos de un ciudadano", "grabar interacción de un ciudadano", "consultar interacciones" y "grabar encuesta de un ciudadano".

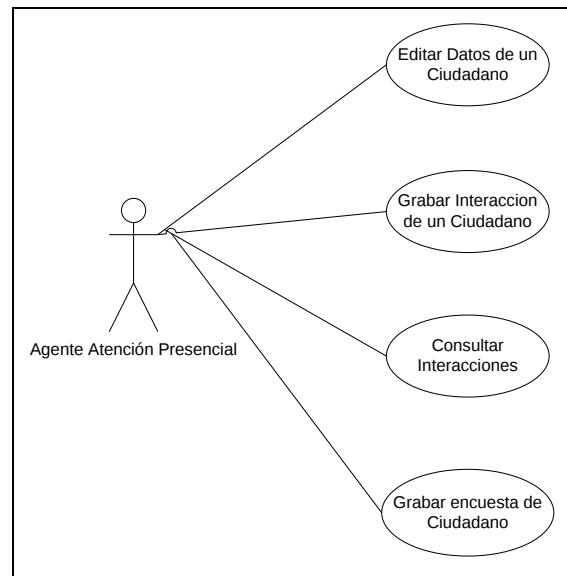


Figura B.13. Modelo de casos de uso del SIW Atento

La Figura B.14 muestra el **modelo de casos de uso extendido** para el servicio de negocio “grabar interacción de un ciudadano”; en el se representan las funcionalidades requeridas por el sistema para llevar a cabo dicho servicio. Así, por ejemplo, para el servicio de negocio “enviar un artículo”, se han identificado las siguientes funcionalidades: “identificarse como autor”, “buscar ciudadanos”, “mostrar campos de búsqueda avanzada de ciudadanos”, “ver resultados de búsqueda de ciudadanos”, “ingresar la información de la interacción” y “ver los resultados de la grabación de la interacción”.

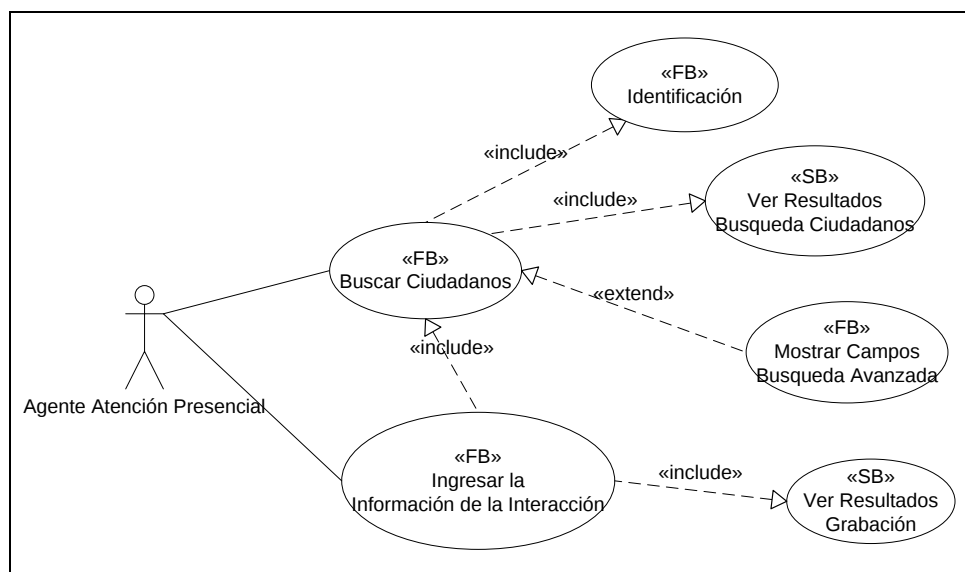


Figura B.14. Modelo de casos de uso extendido para el servicio de negocio “grabar interacción de un ciudadano” del SIW Atento

La Figura B.15 muestra el **modelo de proceso de servicio** para el servicio de negocio “grabar interacción de un ciudadano”, y la Figura B.16 muestra la representación del mismo servicio de negocio en el **modelo de composición de servicio extendido**.

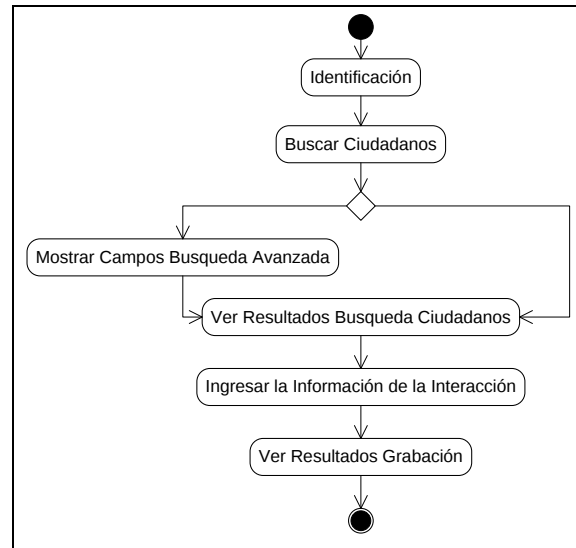


Figura B.15. Modelo de proceso de servicio para el servicio de negocio “grabar interacción de un ciudadano” del SIW Atento

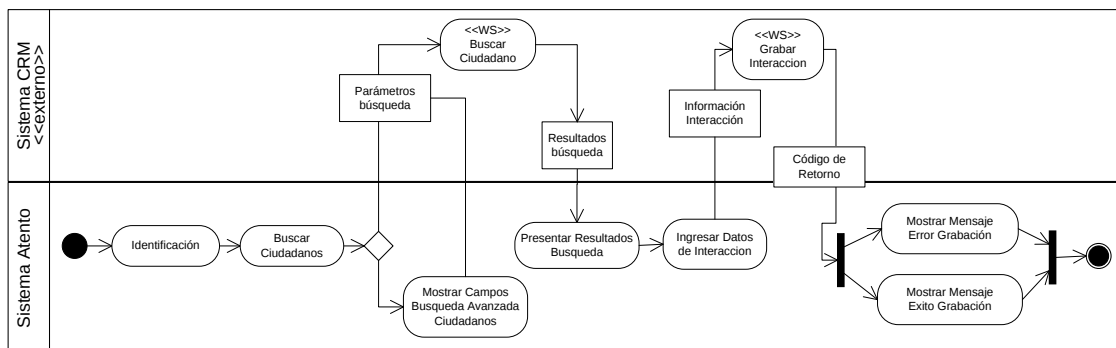


Figura B.16. Modelo de composición de servicio extendido para el servicio de negocio “grabar interacción de un ciudadano” del SIW Atento

Como puede verse en la Figura B.16, la actividad de servicio “ver resultados búsqueda ciudadanos” se descompone en dos acciones: “buscar ciudadano” y “presentar resultados búsqueda”; la actividad de servicio “ingresar la información de la interacción” se descompone en dos acciones: “ingresar datos de interacción” y “grabar interacción”; y la actividad de servicio “ver resultados grabación” se descompone en dos operaciones de actividad: “mostrar mensaje éxito grabación” y “mostrar mensaje error grabación”. El resto de actividades de servicio implica una sola acción. En este modelo, las dos acciones que lleva a cabo el sistema de gestión de relaciones con los ciudadanos (Sistema CRM), que se ha representado

como un colaborador del negocio, se han identificado como acciones que podrán ser implementadas como servicios Web: “buscar ciudadano” y “grabar interacción”.

Las siguientes figuras muestran los modelos realizados para el desarrollo del hipertexto del SIW.

La Figura B.17 muestra, de manera parcial, el **modelo de fragmentos extendido**; en el se representa los fragmentos y las rutas de navegación necesarias para llevar a cabo el servicio de negocio “grabar interacción de un ciudadano”. Dicho modelo se ha construido siguiendo las guías descritas en el apartado 3.7.2 de la sección 3.7.

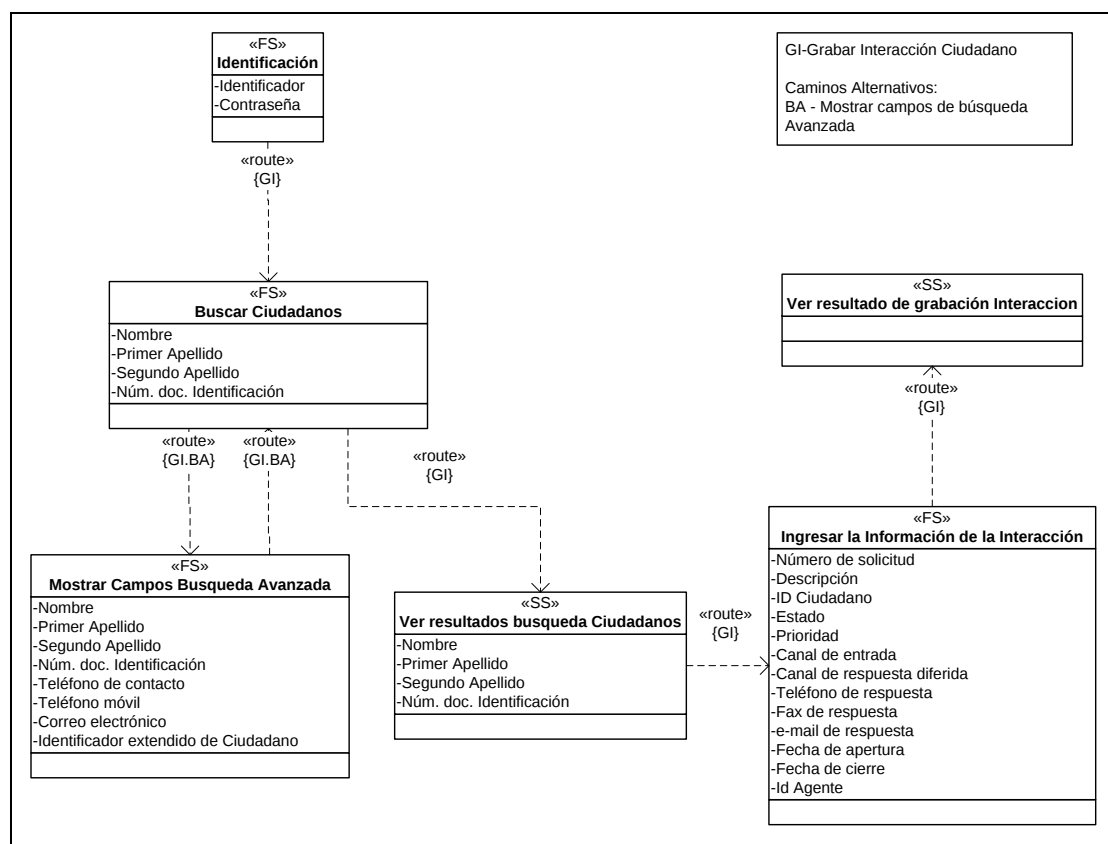


Figura B.17. Modelo de fragmentos extendido parcial para del SIW Atento

La Figura B.18 muestra el **modelo de navegación extendido** que se obtiene añadiendo las estructuras de navegación al modelo de fragmentos extendido descrito previamente.

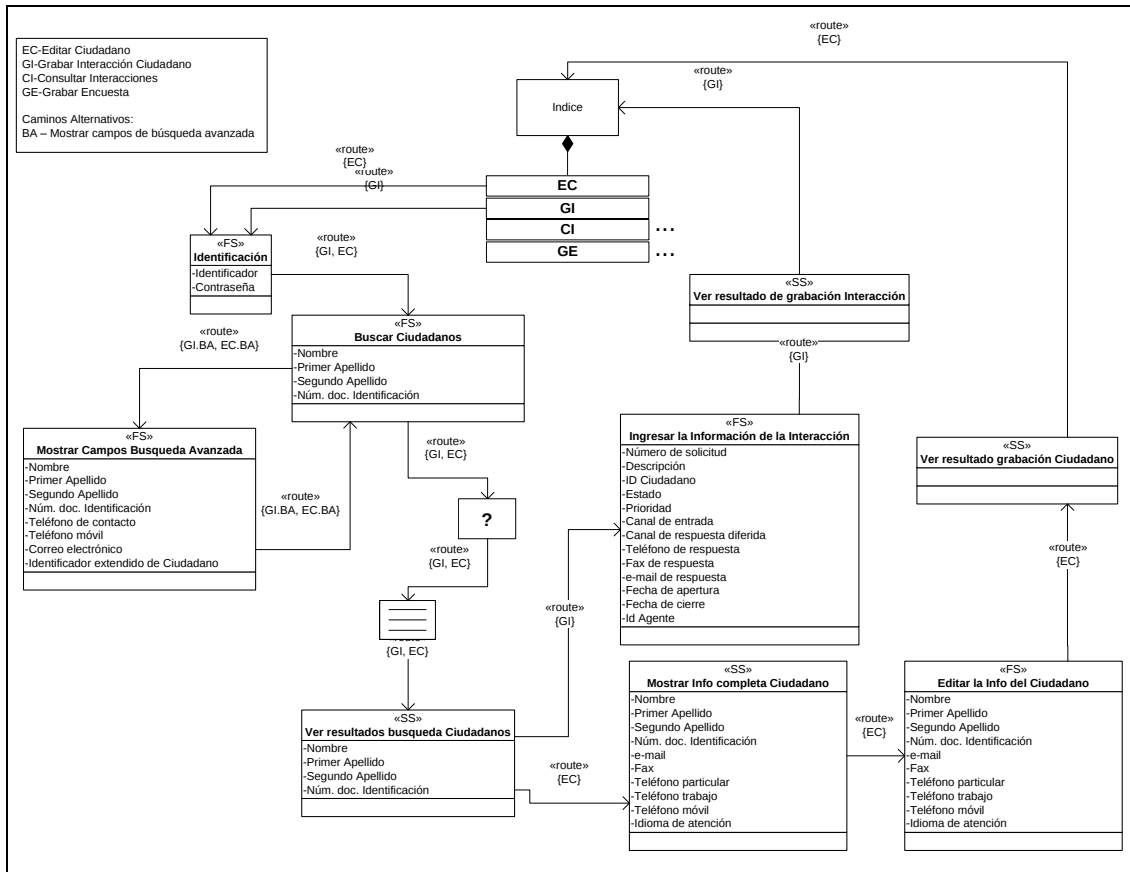


Figura B.18. Modelo de navegación extendido parcial para del SIW Atento

Caso 5: ISLA: Sistema para la Gestión de una Autoescuela

En este apartado se presenta el último caso de estudio realizado. El SIW ISLA contempla todas las actividades de gestión relacionadas con los servicios que proporciona una Autoescuela llamada ISLA. Este SIW ha sido diseñado aplicando el método SOD-M completo y en actualidad está en fase de implementación; un prototipo del mismo está disponible en <http://isla-pfc.selfip.org>.

El negocio de la autoescuela involucra a distintos miembros que interaccionan con la autoescuela: los alumnos, que son los consumidores de los servicios de la autoescuela; el profesor, encargado de la impartición de las clases prácticas y teóricas a los alumnos; la asociación de autoescuelas, que aporta información fundamental para el desarrollo de la autoescuela, tales como nuevas disposiciones; las empresa de distribución de material, encargadas de proporcionar el material necesario para el funcionamiento de la autoescuela; y la Dirección General de Tráfico (DGT), el organismo público que se encarga de conceder los carnés de conducir a los alumnos que lo solicitan, después de realizar un examen y ser declarados aptos.

Los servicios que la autoescuela ofrece a los alumnos son los siguientes:

- *Obtención del carné de conducir:* este es el principal servicio que da la autoescuela, y proporciona a los alumnos la posibilidad de obtener el carné de conducir. Para ello, el alumno debe primero contratar los servicios de la autoescuela, adquirir el material que le proporciona la misma y, posteriormente, apuntarse a clases prácticas y teóricas. Después de haber aprobado los exámenes teórico y práctico en la DGT, el alumno podrá solicitar la expedición del carné y la autoescuela dará por finalizado el contrato con el mismo.
- *Solicitud de clases prácticas de conducción:* proporciona al alumno la posibilidad de apuntarse a clases prácticas de conducción, por ejemplo, si lleva muchos años sin conducir, o bien para una convalidación de carné extranjero. Para ello, el alumno deberá también formalizar un contrato con la autoescuela y, posteriormente, apuntarse a las clases prácticas, seleccionando horas disponibles y realizando el pago de las mismas.
- *Solicitud de compra de material:* proporciona al alumno a posibilidad de adquirir el material que proporciona la autoescuela, tales como libros de teoría, libros de test prácticos, placas de conductor novel (L), etc. Para ello el alumno deberá seleccionar el material que desea adquirir y realizar el pago del mismo.
- *Recuperación del carné de conducir:* ofrece al alumno la posibilidad de asistir a un curso teórico para recuperar el carné de conducir que le ha sido retirado por una autoridad competente, por ejemplo, por haber agotados los puntos del mismo. Para recuperar el carné, el alumno debe apuntarse y asistir a todas las clases de un curso de reciclaje impartido por la autoescuela, por lo que se deberá llevar un control de asistencia.

En la Figura B.19 se muestra el **modelo de valor** realizado para este caso, en el que se representan los distintos miembros del negocio que se han descrito previamente: La autoescuela, los alumnos, profesores, la asociación de autoescuelas, la empresa de materia y la DGT.

Como actividades de valor de la autoescuela se representa: la *obtención del permiso de circulación*, que es la actividad que proporciona a los alumnos la posibilidad de obtener el carné de conducir; la *enseñanza teórica*, que comprende las actividades de impartición de clases teóricas que ofrece la autoescuela; *enseñanza práctica*, actividad que comprende la impartición de clases prácticas, es

decir clases de conducción que imparten los profesores con los coches de autoescuelas; *venta de material*, comprendiendo la actividad de venta tanto de material para la obtención del permiso, por ejemplo libros de test o de teoría, como también cualquier otro material que desee un alumno como placas de conductor novel; *reciclaje*, esta actividad tiene especial relación con los carnés de conducir por puntos, y comprende los cursos a los que acceden aquellas personas que lo pierdan su carné de conducir por diferentes tipos de infracciones. La actividad de valor del profesor es el *conocimiento* que este posee y que es gestionado por la autoescuela para ofrecérselo a los alumnos. Las actividades de valor de la empresa de material son la impresión del material para los alumnos y para la autoescuela en general. Finalmente, de la DGT, se representan como actividades de valor tanto la concesión de permiso de conducir como la realización de exámenes a los alumnos que se lleva a cabo en los centros de la DGT.

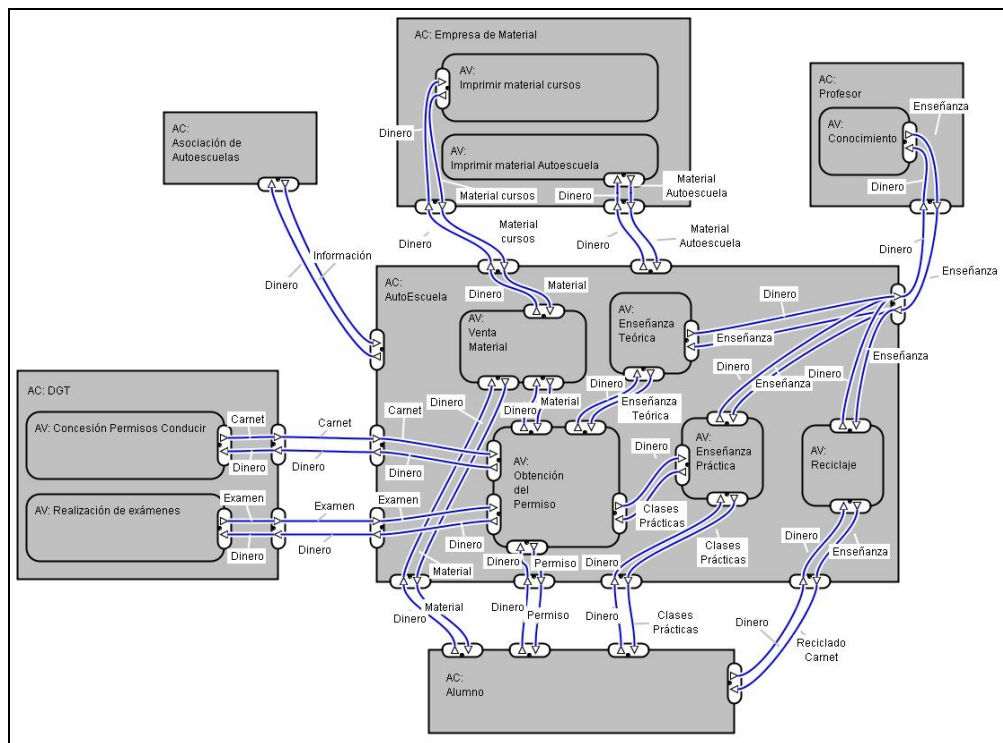


Figura B.19. Modelo de valor para la Autoescuela ISLA

En las siguientes figuras se muestran los distintos caminos de dependencias identificados, que representan los intercambios de objetos de valor que ocurren ante cada una de las necesidades de los consumidores, que en este caso son los alumnos.

La Figura B.20 muestra el camino de dependencia ante la necesidad de un alumno de obtener el carné de conducir; la Figura B.21 muestra el camino de

dependencia ante la necesidad de un alumno de asistir a clases prácticas; la Figura B.22 muestra el camino de dependencia ante la necesidad de un alumno de comprar material; y finalmente, la Figura B.23 muestra el camino de dependencia ante la necesidad de un alumno de realizar un curso de reciclaje.

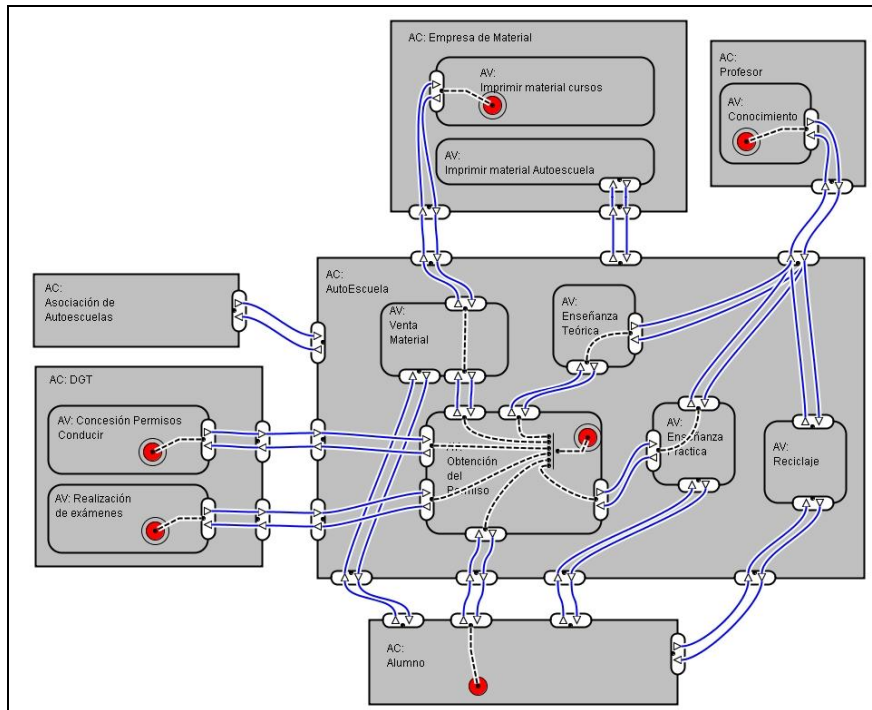


Figura B.20. Modelo de valor: necesidad del alumno de obtener el carné de conducir

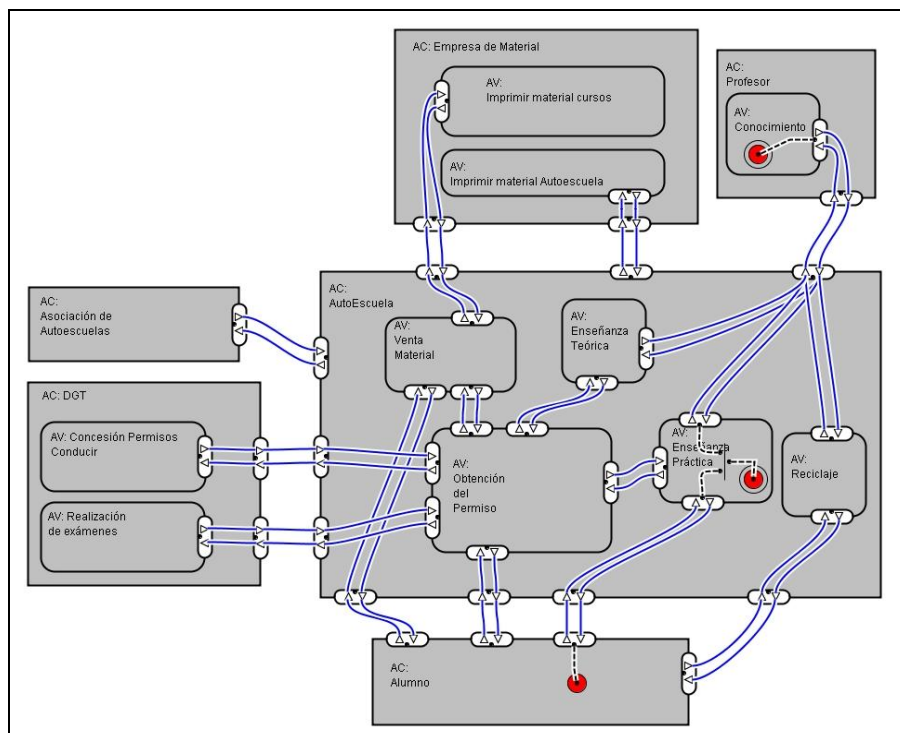


Figura B.21. Modelo de valor: necesidad del alumno de tomar clases prácticas

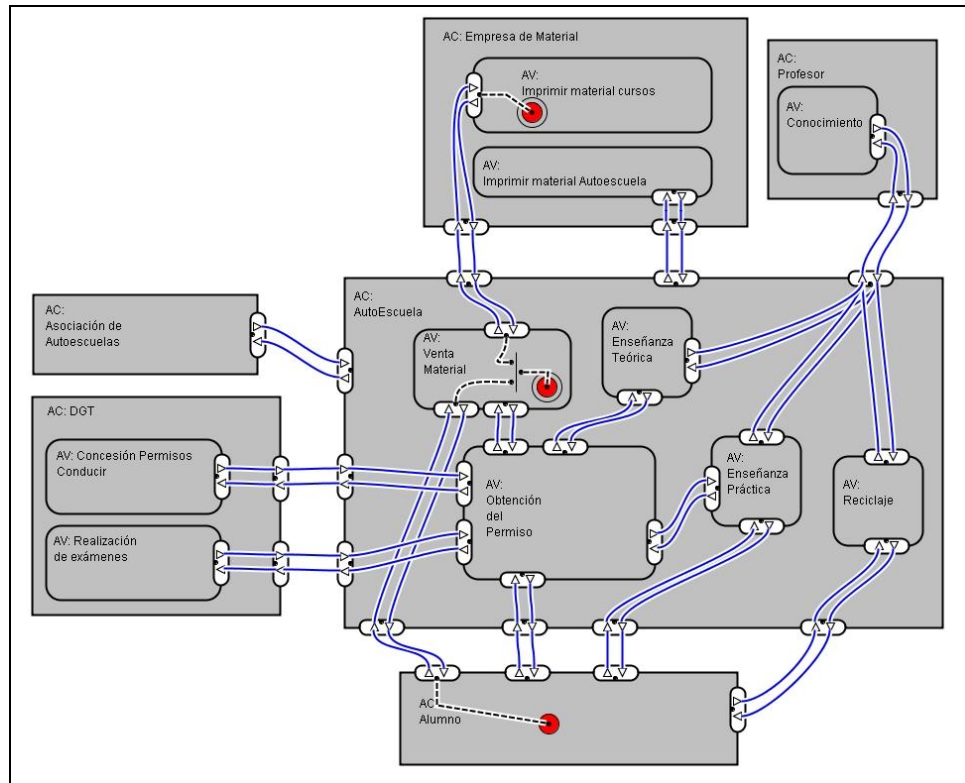


Figura B.22. Modelo de valor: necesidad del alumno de comprar material

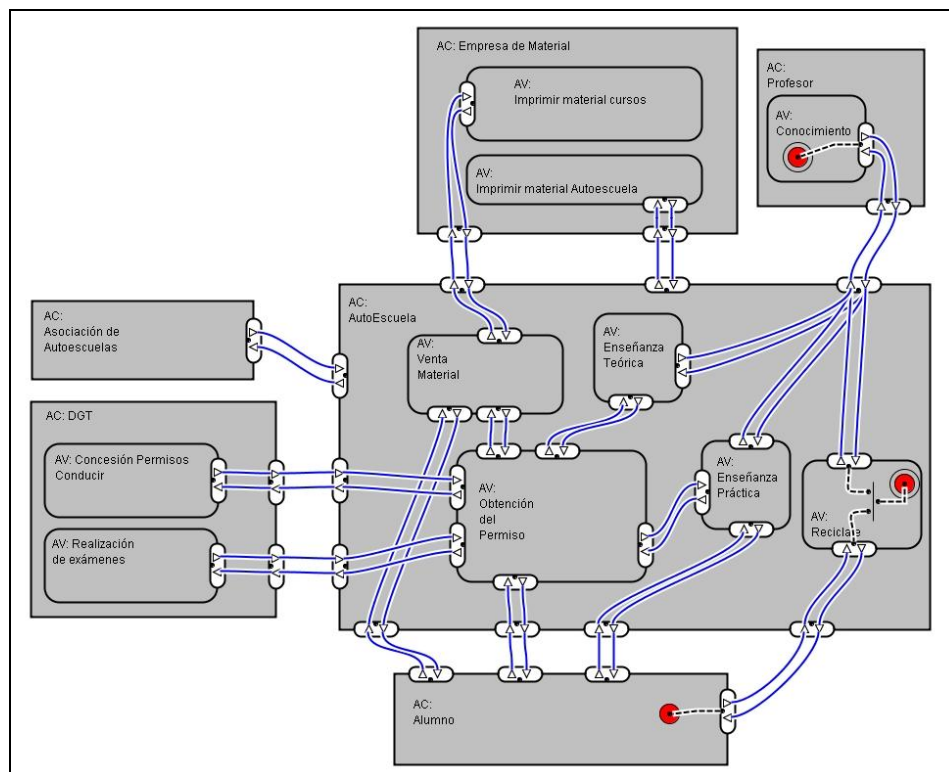


Figura B.23. Modelo de valor: necesidad del alumno de realizar un curso de reciclaje

La **lista de los servicios de negocio**, que se muestra en la Tabla B.1 provee una descripción de todos los servicios que forma parte de un negocio y que se ofrecen para satisfacer las necesidades del consumidor final de la Autoescuela ISLA. En este caso no ha sido necesaria la representación del modelo de procesos de negocio, por lo que los servicios de negocio que requiere un alumno, que se muestran en la Tabla B.1, se han obtenido sólo a partir del modelo de valor, identificando las necesidades o estímulos iniciales de los camino de dependencia.

Tabla B.1. Lista de servicios de negocio para la Autoescuela ISLA

Consumidor Final	Servicios de Negocio
Alumno	Servicio de obtención del carné de conducir
	Servicio de clases prácticas
	Servicio de venta de material
	Servicio de recuperación del carné de conducir

La Figura B.24 muestra el **modelo de casos de uso** en el que se representa como actor al alumno que es el consumidor del final de los servicios de negocio que se implementarán en el SIW. Los servicios de negocio se muestran como casos de uso en el modelo.

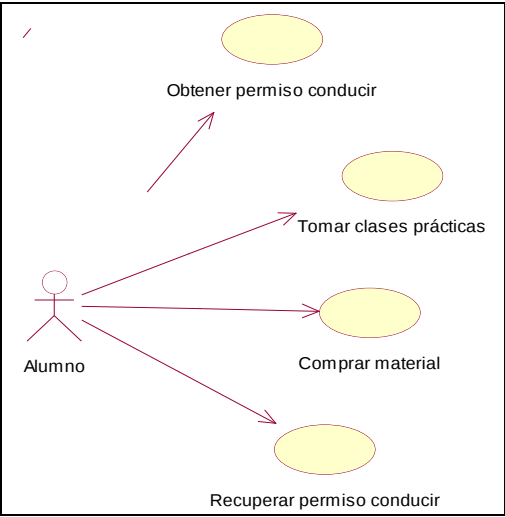


Figura B.24. Modelo de casos de uso del SIW ISLA

Las siguientes figuras muestran los distintos diagramas que se han realizado, uno por cada servicio de negocio identificado en el modelo de casos de uso, para la representación del **modelo de casos de uso extendido**. Así, la Figura B.25 muestra el modelo de casos de uso extendido para el servicio de negocio “obtener

permiso de conducir”, la Figura B.26 muestra el mismo modelo para el servicio de negocio “tomar clases prácticas”, y la Figura B.27 muestra el mismo modelo para el servicio de negocio “comprar material”, y la Figura B.28 muestra el modelo de casos de uso extendido para el servicio de negocio “recuperar permiso de conducir”.

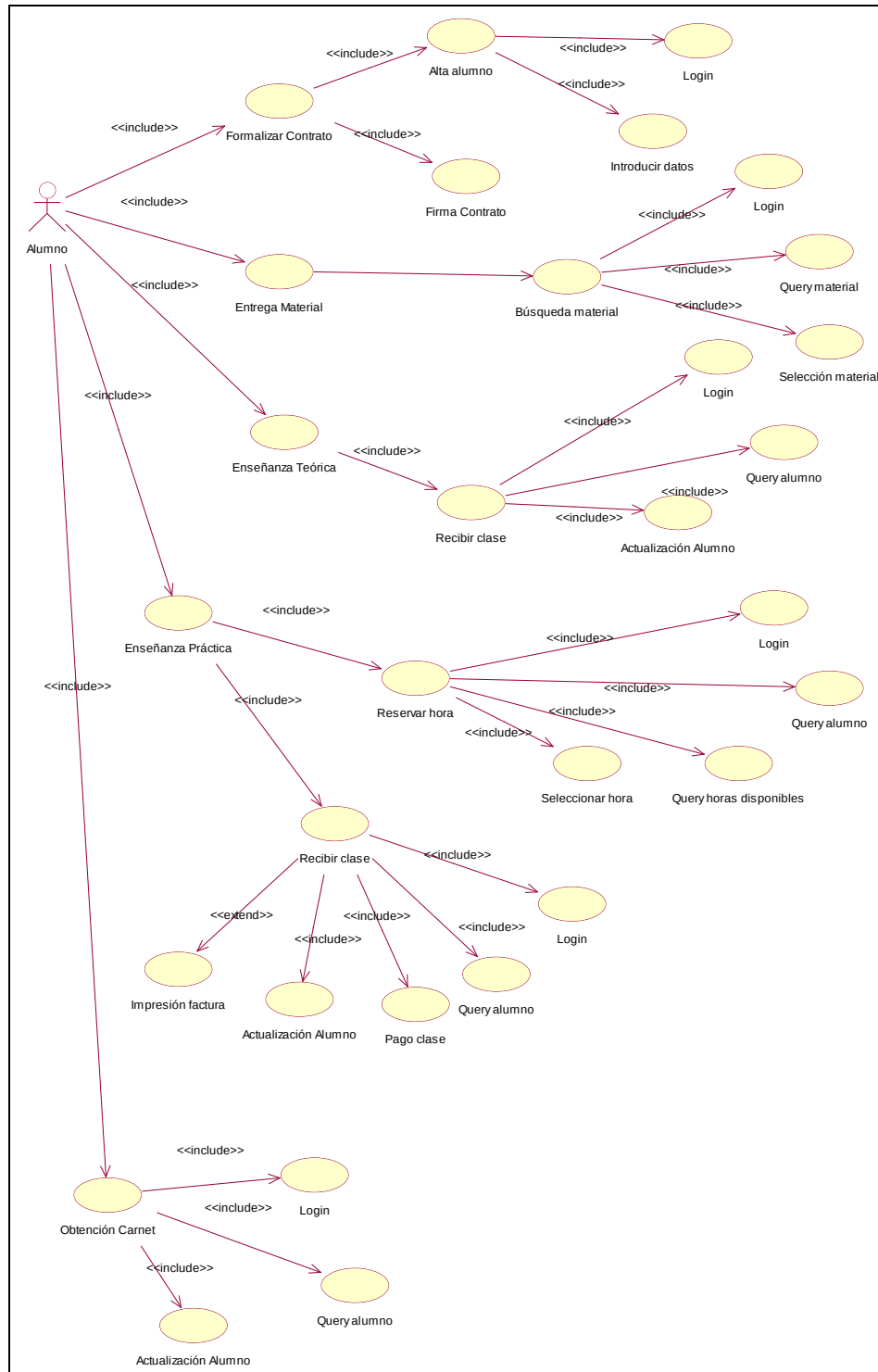


Figura B.25. Modelo de casos de uso extendido para el servicio de negocio “obtener permiso de conducir”

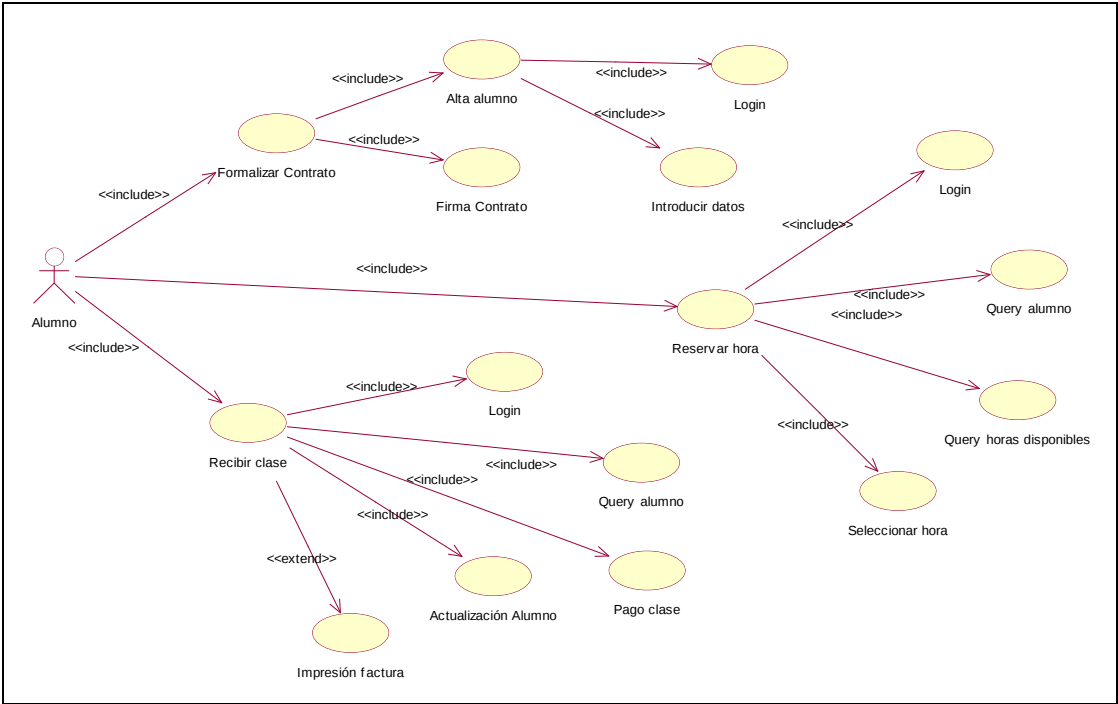


Figura B.26. Modelo de casos de uso extendido para el servicio de negocio “tomar clases prácticas”

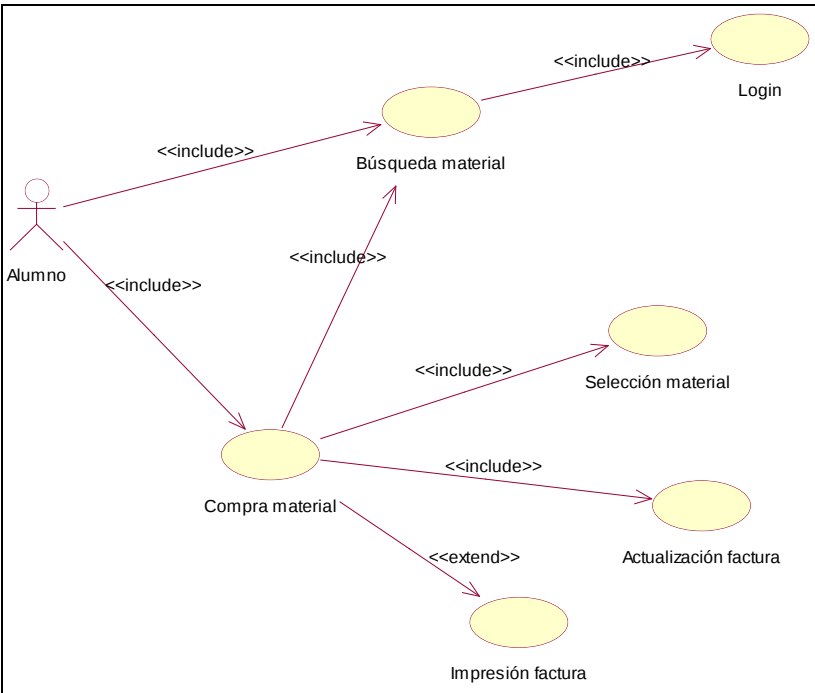


Figura B.27. Modelo de casos de uso extendido para el servicio de negocio “comprar material”

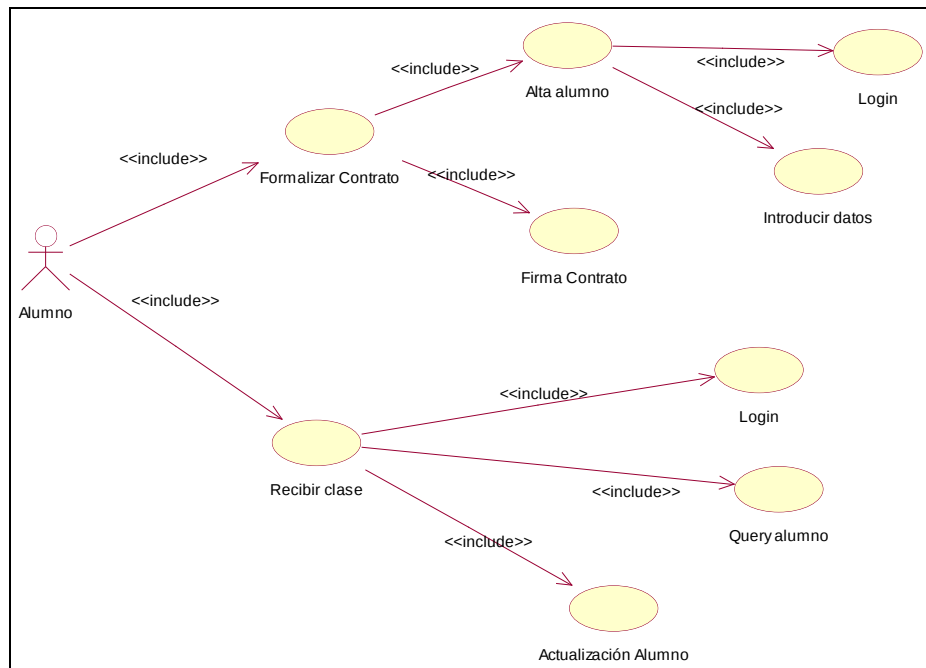


Figura B.28. Modelo de casos de uso extendido para el servicio de negocio “recuperar permiso de conducir”

APÉNDICE C: REGLAS DE TRANSFORMACIÓN

En este apéndice se incluyen el conjunto de reglas de transformación basadas en grafos para las transformaciones completamente automatizables definidas entre los modelos propuestos por SOD-M, las cuales se han presentado en el apartado 3.6.2 de la sección 3.6. En concreto, en este apéndice se especifican las siguientes reglas de transformación: *a) entre modelos de nivel PIM* (reglas 1, 2, 4, 4.1, 4.2, 5, 5.1, 6, 7 y 7.1 de la Tabla 3.9); *b) entre un modelo de nivel PIM y otro de nivel PSM* (reglas 8 y 8.1 de la Tabla 3.9); *c) reglas de transformación entre los modelos de nivel PSM* (reglas 9, 9.1, 9.2 y 9.3 de la Tabla 3.9). Tales reglas de transformación basadas en grafos se describen de acuerdo con las directrices presentadas en el 3.6.2.3 de la sección 3.6.

En la Figura C.1 se muestran las reglas 1 y 2, entre modelos de nivel PIM, de la Tabla 3.9, las reglas 4 y 4.1 se presentan en la Figura C.2, la regla 4.2 se representa en la Figura C.3, las regla 5 en la Figura C.4 y la 5.1 en la Figura C.5, finalmente, la reglas 6, 7 y 7.1 se muestran en la Figura C.6.

En la Figura C.7 se muestran las reglas 8 y 8.1, entre modelos de nivel PIM y PSM, de la Tabla 3.9.

Finalmente, en la Figura C.8 se muestran las reglas 9 y 9.1, entre modelos de nivel PSM, y en la Figura C.9 las reglas 9.2 y 9.3.

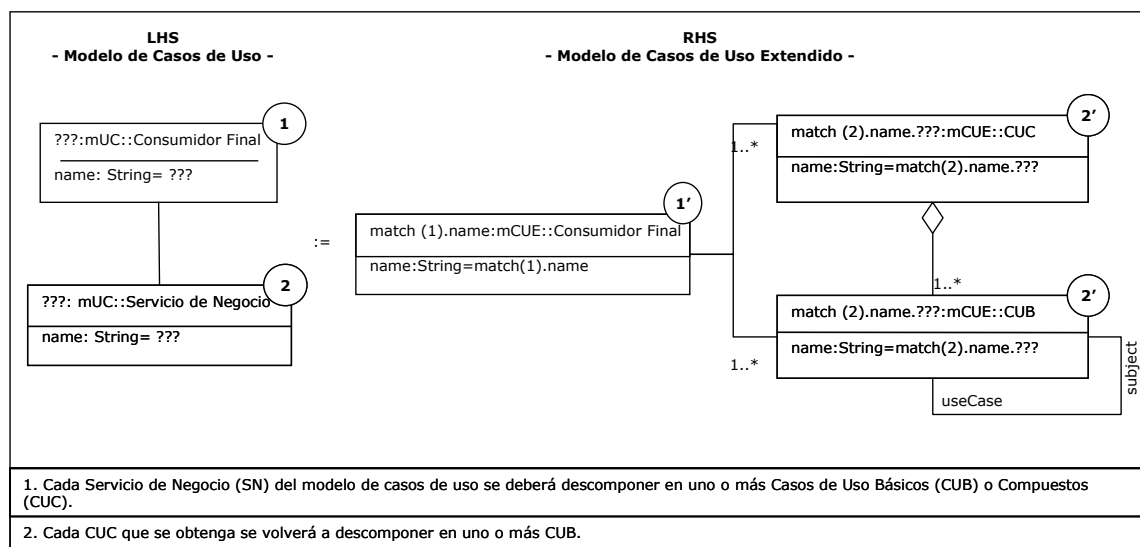


Figura C.1. Transformación de *Servicios de Negocios* y *Consumidor Final* del metamodelo de Casos de Uso a *Casos de Uso Básicos* y *Compuestos* en el metamodelo de Casos de Uso Extendido

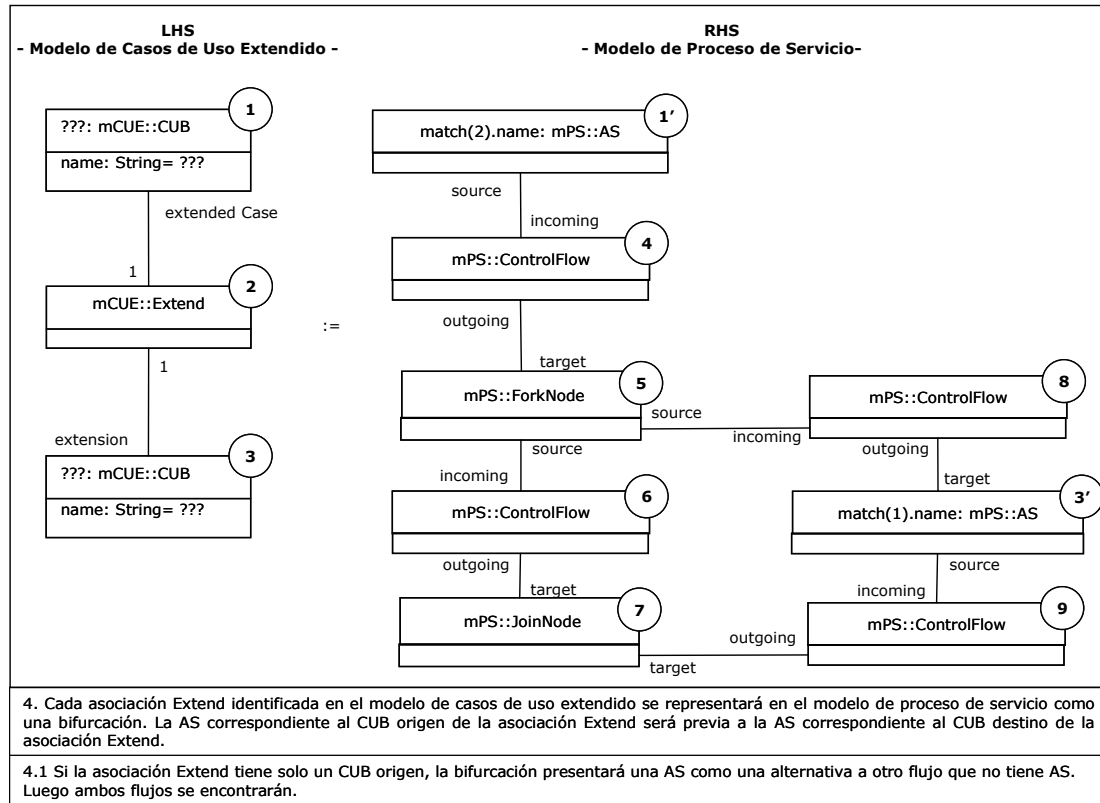


Figura C.2. Transformación de relaciones *extend* del metamodelo de Casos de Uso Extendido al metamodelo de Procesos de Servicio

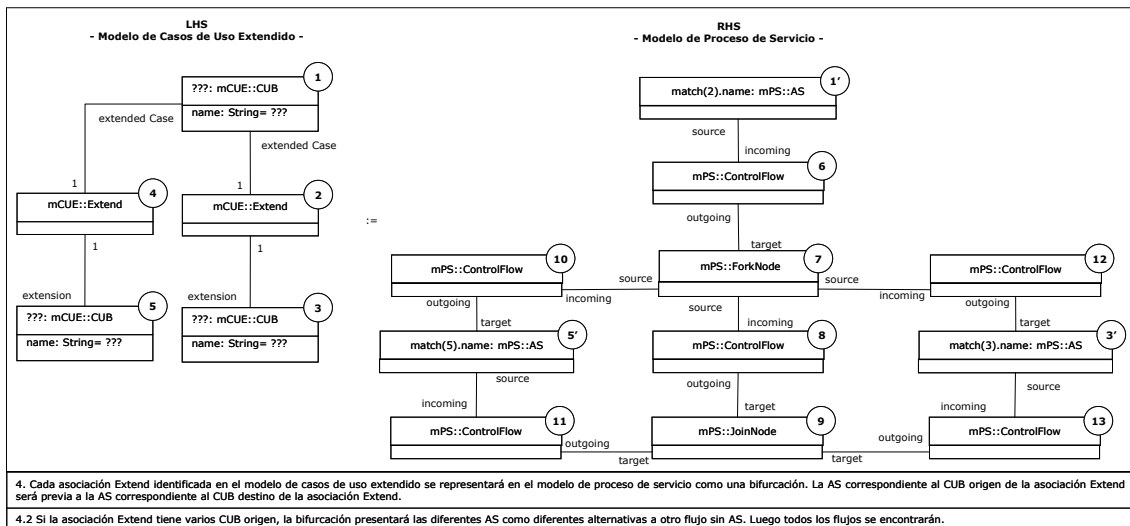


Figura C.3. Transformación de relaciones *extend* (con varios orígenes) del metamodelo de Casos de Uso Extendido al metamodelo de Procesos de Servicio

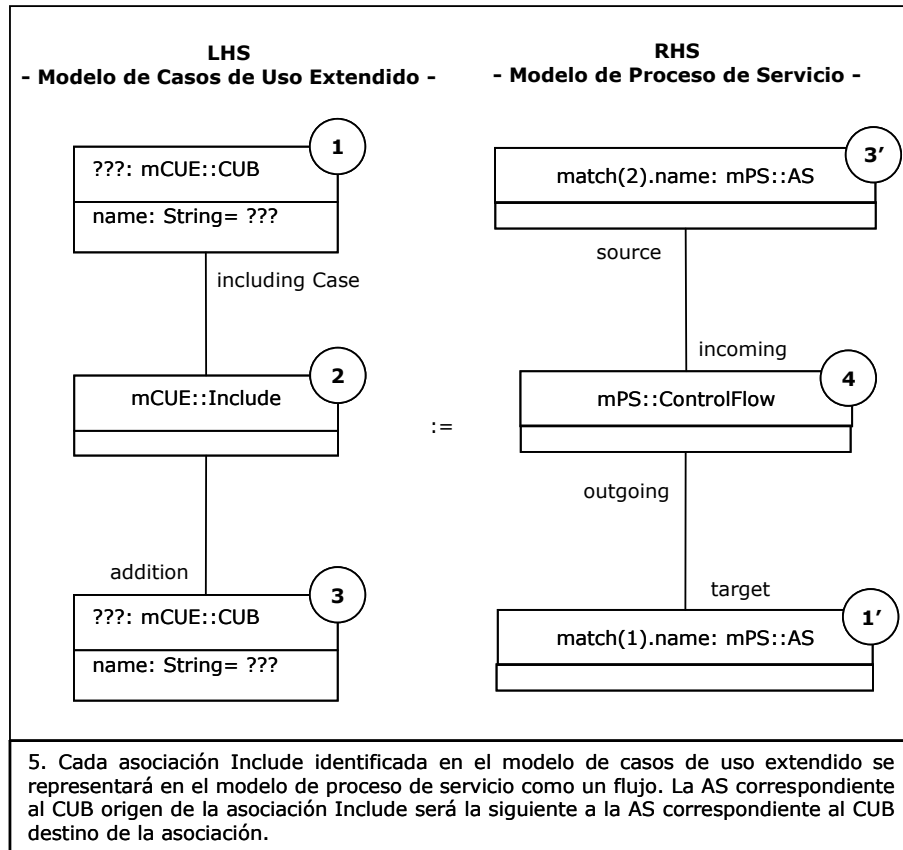


Figura C.4. Transformación de relaciones *include* del metamodelo de Casos de Uso Extendido al metamodelo de Procesos de Servicio

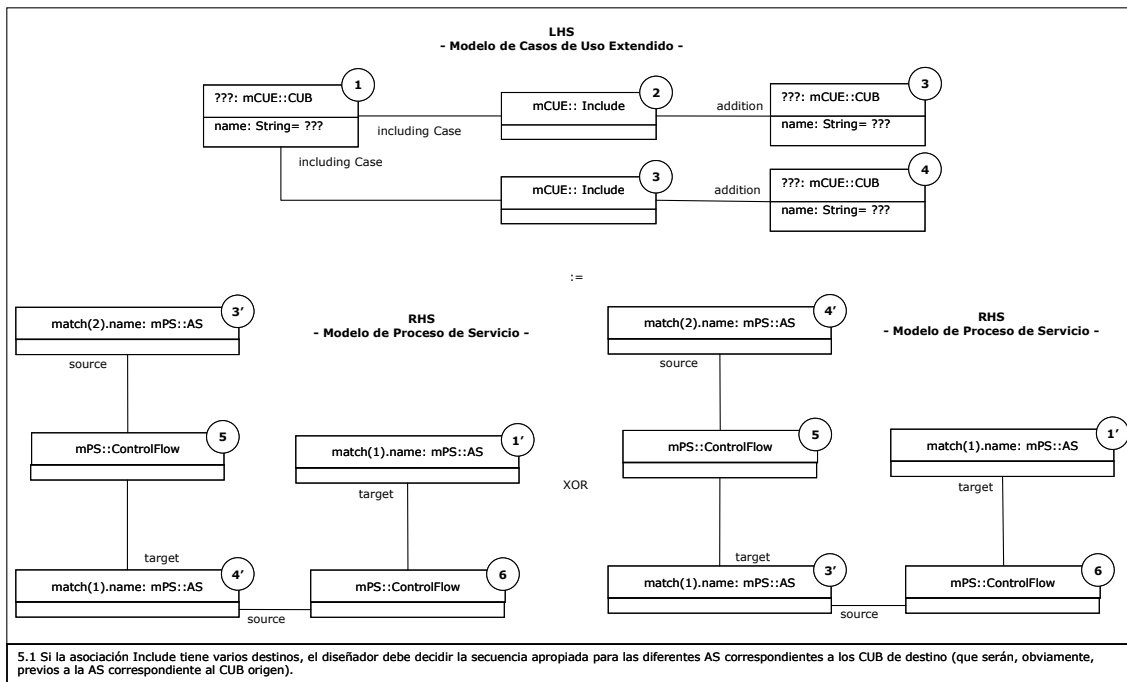


Figura C.5. Transformación de relaciones *include* (con varios destinos) del metamodelo de Casos de Uso Extendido al metamodelo de Procesos de Servicio

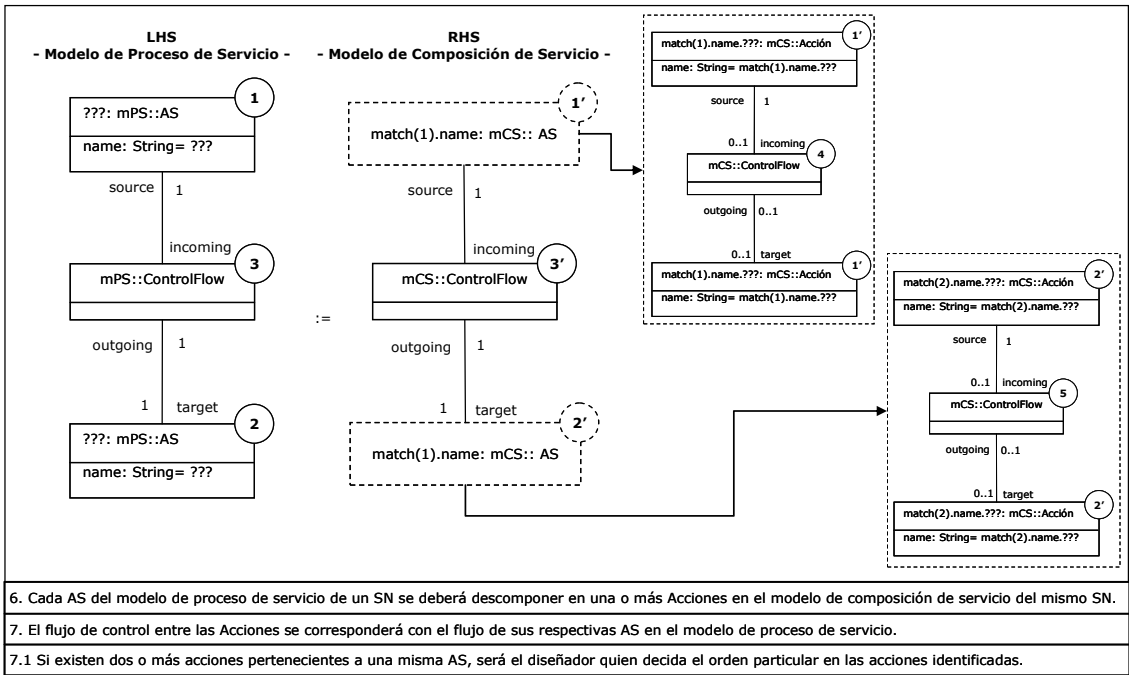


Figura C.6. Transformación de *Actividades de Servicios* del metamodelo de Procesos de Servicio al metamodelo de Composición de Servicio

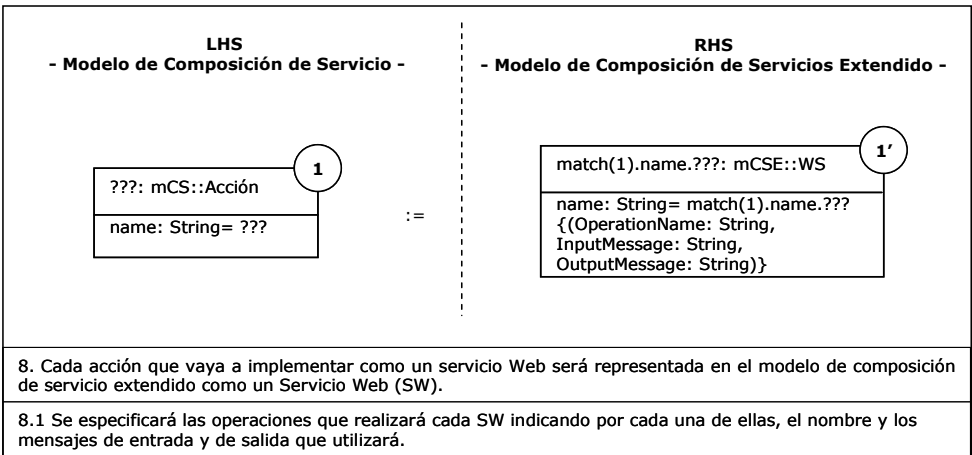


Figura C.7. Transformación de *Acciones* del metamodelo de Composición de Servicio a *Servicios Web* en el metamodelo de Composición de Servicio Extendido

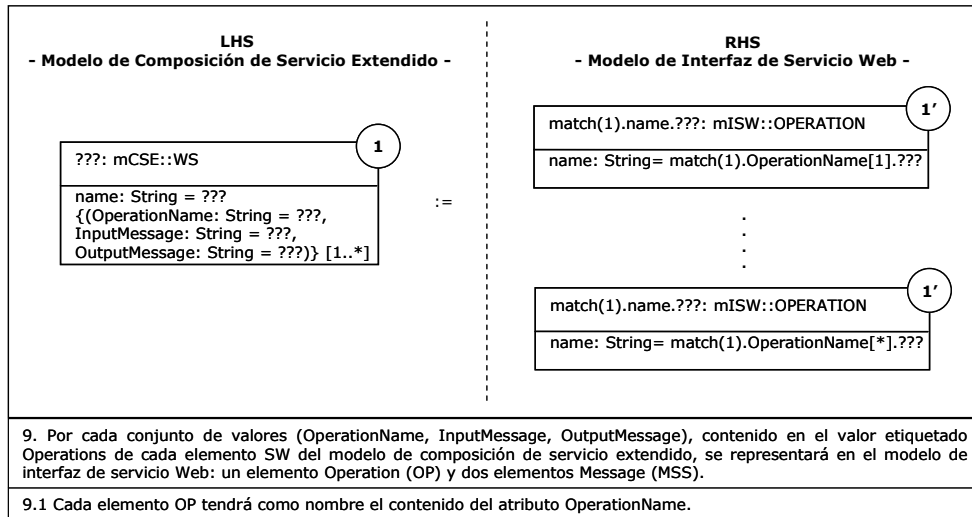


Figura C.8. Transformación de *Servicios Web* del metamodelo de Composición de Servicio Extendido a elementos *Operation* en el metamodelo de Interfaz de Servicio Web

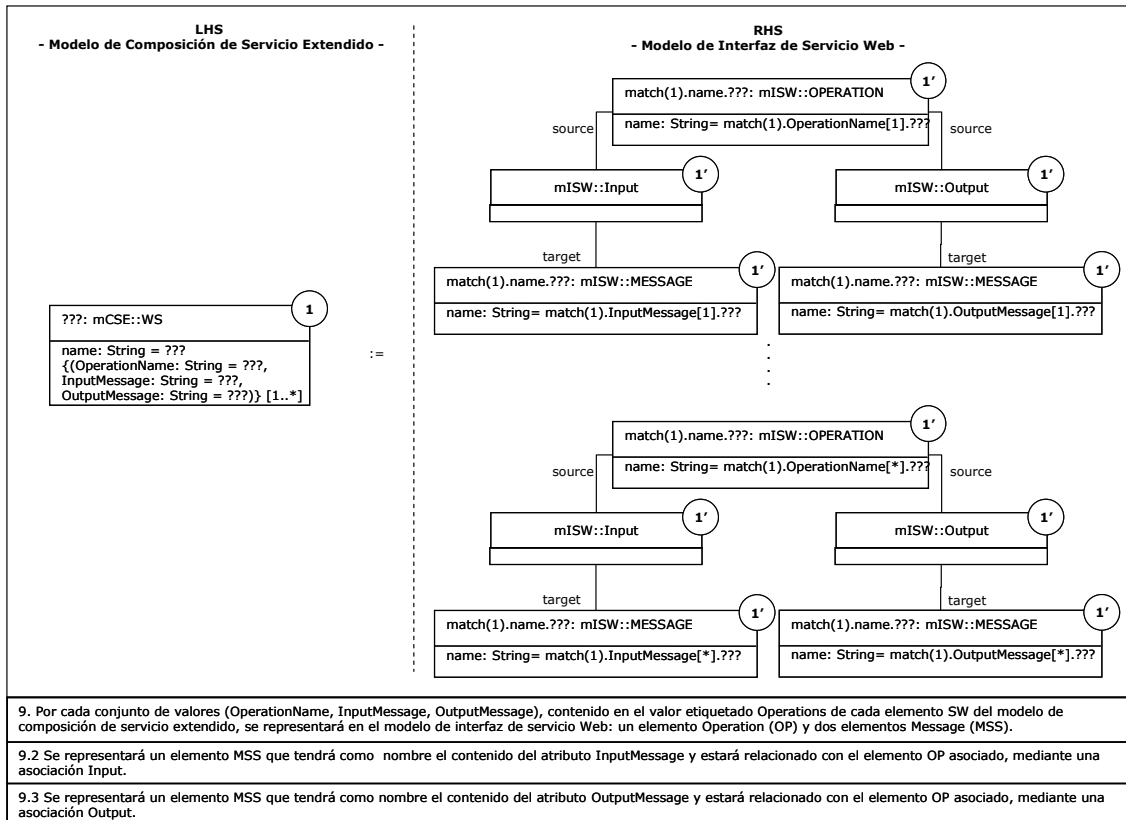


Figura C.9. Transformación de *Servicios Web* del metamodelo de Composición de Servicio Extendido a elementos *Message* en el metamodelo de Interfaz de Servicio Web