

Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo

M.J. Escalona⁽¹⁾, N. Koch⁽²⁾

⁽¹⁾Universidad de Sevilla. Lenguajes y Sistemas Informáticos. España
escalona@lsi.us.es

⁽²⁾Universidad de Munich y F.A.S.T. GmbH, Munich, Alemania
koehn@informatik.uni-muenchen.de
koch@fast.de

Resumen

En los últimos años ha ganado reconocimiento la importancia de la ingeniería de requisitos y los riesgos en que se incurren si ésta es realizada en forma incompleta o incorrecta. Actividades propias de esta área, como la captura de requisitos del usuario, la especificación de requisitos o la validación de los mismos, son algunas de las consideradas más críticas en el desarrollo y la producción del software. Estas actividades tienen por objetivo la determinación de las necesidades del sistema, la adquisición por parte del equipo de desarrollo de la información necesaria para desarrollar un producto de calidad y en definitiva, la comunicación inicial entre el grupo de expertos y los clientes y usuarios. La ingeniería de requisitos es un tema complejo y crucial para el éxito de todo el proyecto de ingeniería del software. Las aplicaciones en el entorno de la web no son una excepción. Sin embargo, las propuestas metodológicas para el desarrollo de aplicaciones web contemplan el tema de ingeniería de requisitos con distinto grado de profundidad y proponen el uso de diferentes técnicas. Este trabajo ofrece una visión general y un estudio comparativo del tratamiento de requisitos en las actuales metodologías para la web.

Palabras clave: Sistemas web, metodologías para la web, ingeniería de requisitos, survey.

1. Introducción

El avance de Internet y las comunicaciones de los últimos años ha provocado un interés creciente por el desarrollo de propuestas metodológicas que ofrezcan un marco de referencia adecuado cuando se desarrolla un sistema de información web. Así, en los últimos años son varios los grupos que han trabajado y han propuesto metodologías que ofrecen procesos, modelos y técnicas adecuadas para trabajar con este tipo de sistemas [17][30][8]. Sin embargo, si se analizan las diferentes propuestas, la gran mayoría de ellas enfocan su trabajo en la etapa de diseño del ciclo de vida, dando menor importancia a la ingeniería de requisitos, al testeo y a la gestión de calidad.

El tratamiento de requisitos es el proceso mediante el cual se especifican y validan los servicios que debe proporcionar el sistema así como las restricciones sobre las que se deberá operar. Consiste en un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en una variedad de formatos y probando la exactitud del conocimiento adquirido [10]. La importancia de esta fase es esencial puesto que los errores más comunes y más costosos de reparar, así como los que más tiempo consumen se deben a una inadecuada ingeniería de requisitos.

Este trabajo presenta el estado del arte y un estudio comparativo de las actuales propuestas que existen en el entorno de la web y que utilizan diferentes técnicas y modelos para la elicitación y el análisis de requisitos¹. Para ello, en el apartado 2 se presentan una visión general de las técnicas clásicas utilizadas en el mundo de la ingeniería de requisitos para capturar, definir y verificar los requisitos de un sistema. En el apartado 3, se presentan aquellas metodologías del ámbito de la web, que a nuestro saber, incluyen el tratamiento de los requisitos con distinto grado de detalle. En el apartado 4, se clasifican y comparan las propuestas presentadas en el apartado 3 y por último, en el apartado 5 se presentan las conclusiones y trabajos futuros.

2. La Ingeniería de Requisitos

En el proceso de desarrollo de un sistema, sea o no para la web, el equipo de desarrollo se enfrenta al problema de la identificación de requisitos. La definición de las necesidades del sistema es un proceso complejo pues en él hay que identificar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y de los clientes. Para realizar este proceso, no existe una única técnica estandarizada y estructurada que ofrezca un marco de desarrollo que garantice la calidad del resultado. Existe en cambio un conjunto de técnicas, cuyo uso proponen las diferentes metodologías para el desarrollo de aplicaciones web. Se debe tener en cuenta que la selección de las técnicas

¹ Este trabajo ha sido financiado mediante una beca otorgada por el Deutscher Akademischer Austauschdienst (DAAD) y por el Ministerio de Educación y Ciencia y los fondos FEDER ((TIC2000-1673-C06-03).

y el éxito de los resultados que se obtengan depende en gran medida tanto del equipo de análisis y desarrollo como de los propios clientes o usuarios que en ella participen.

El proceso de especificación de requisitos se puede dividir en tres grandes actividades [23]:

- 1- captura de requisitos
- 2- definición de requisitos
- 3- validación de requisitos

En la figura 1 se presenta el proceso de ingeniería de requisitos que incluye estas tres actividades. Para la representación se ha usado la notación de diagrama de actividades propuesta en UML [35].

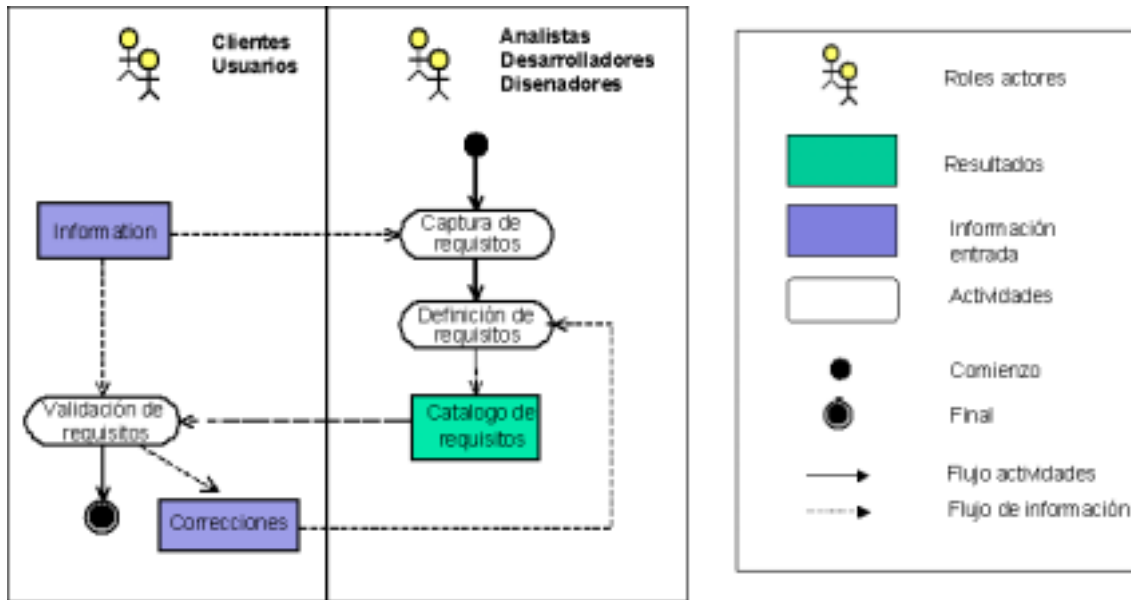


Figura 1: Proceso de Ingeniería de Requisitos

El proceso comienza con la realización de la captura de requisitos, el grupo de técnicos toma la información suministrada por los usuarios y clientes. Esta información puede provenir de fuentes muy diversas: documentos, aplicaciones existentes, a través de entrevistas, etc. En base a ella, el equipo de desarrollo elabora el catálogo de requisitos. Finalmente con la validación de requisitos se realiza la valoración de los mismos, comprobando si existen inconsistencias, errores o si faltan requisitos por definir. El proceso de definición-validación es iterativo y en algunos proyectos complejos resulta necesario ejecutarlo varias veces.

En los siguientes apartados se presentan brevemente algunas técnicas clásicas para realizar las actividades de captura, definición y validación de requisitos. Estas técnicas pueden resultar más o menos apropiadas para la ingeniería de requisitos para aplicaciones en el entorno web y resulta muy difícil establecer criterios para seleccionar técnicas apropiadas. Entre estos criterios pueden ser considerados la facilidad de aprendizaje y de uso, la escalabilidad, el costo, la calidad y completitud de los resultados y el tiempo requerido para aplicar las técnicas.

2.1 Captura de requisitos

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema [6]. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Por la complejidad que todo esto puede implicar, la ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa.

A continuación se presentan un grupo de técnicas que de forma clásica han sido utilizadas para esta actividad en el proceso de desarrollo de todo tipo de software.

- **Entrevistas:** resultan una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Existen muchos tipos de entrevistas y son muchos los autores que han trabajado en definir su estructura y dar guías para su correcta realización [7][27]. Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y documentación de los resultados (protocolo de la entrevista). Las entrevistas, sin embargo, no es una técnica sencilla de aplicar [27]. Requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo siempre limitado. Bajo este aspecto la preparación de la entrevista (ver punto cuestionarios y checklists) representa un papel esencial.

- **JAD (Joint Application Development/Desarrollo conjunto de aplicaciones):** esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes [13]. Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la entrevista se trabajará sobre lo que se generará. Tras una fase de preparación del JAD al caso concreto, el equipo de trabajo se reúne en varias sesiones. En cada una de ellas se establecen los requisitos de alto nivel a trabajar, el ámbito del problema y la documentación. Durante la sesión se discute en grupo sobre estos temas llegándose a una serie de conclusiones que se documentan. En cada sesión se van concretando más las necesidades del sistema. Esta técnica presenta una serie de ventajas frente a las entrevistas tradicionales ya que ahorra tiempo al evitar que las opiniones de los clientes se tengan que contrastar por separado. Pero requiere un grupo de participantes bien integrados y organizados.
- **Brainstorming (Tormenta de ideas):** es también una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre [29]. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. Como técnica de captura de requisitos es sencilla de usar y de aplicar, contrariamente al JAD puesto que no requiere tanto trabajo en grupo como éste. Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del sistema, por lo que suele aplicarse en los primeros encuentros.
- **Concept Mapping:** Los concept maps [27] son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Son muy usados dentro de la ingeniería de Requisitos pues son fáciles de entender por el usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en el lenguaje de éste. Sin embargo, deben ser usados con cautela porque en algunos casos pueden ser muy sugestivos y pueden llegar a ser ambiguos en casos complejos si no se acompaña de una descripción textual.
- **Sketches y Storyboards:** Esta técnica es frecuentemente usada por los diseñadores gráficos de aplicaciones en el entorno web. La misma consiste en representar sobre papel en forma muy esquemática las diferentes interfaces al usuario (sketches). Estos sketches pueden ser agrupados y unidos por enlaces dando idea de la estructura de navegación (storyboard).
- **Casos de Uso:** Aunque inicialmente se desarrollaron como técnica para la definición de requisitos [15], algunos autores proponen casos de uso como técnica para la captura de requisitos [27][22]. Los casos de uso permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los *actores* del mismo para realizar una determinada función. Los actores son elementos externos (personas, otros sistemas, etc.) que interactúan con el sistema como si de una caja negra se tratase. Un actor puede participar en varios casos de uso y un caso de uso puede interactuar con varios actores. La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria [33][14] si no se acompañan con una información textual o detallada con otra técnica como pueden ser diagramas de actividades [35].
- **Cuestionarios y Checklists:** Esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario (Checklist). Este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista.
- **Comparación de terminología:** Uno de los problemas que surge durante la elicitación de requisitos es que usuarios y expertos no llegan a entenderse debido a problemas de terminología. Esta técnica es utilizada en forma complementaria a otras para obtener consenso respecto de la terminología a ser usada en el proyecto de desarrollo. Para ello es necesario identificar el uso de términos diferentes para los mismos conceptos (correspondencia), misma terminología para diferentes conceptos (conflictos) o cuando no hay concordancia exacta ni en la vocabulario ni en los conceptos (contraste) [27].

Existen más técnicas para la captura de requisitos (análisis de otros sistemas, estudio de la documentación, etc.) incluso también es común encontrar alternativas que combinen varias de estas técnicas [27]. Sin embargo, las presentadas ofrecen un conjunto representativo de las más utilizadas y resultan suficientes para el objetivo de este trabajo.

2.2 Definición de requisitos

También para la actividad definición de requisitos en el proceso de ingeniería de requisitos hay un gran número de técnicas propuestas. Describimos brevemente las más relevantes para este trabajo.

- **Lenguaje natural:** Resulta una técnica muy ambigua para la definición de los requisitos. Consiste en definir los requisitos en lenguaje natural sin usar reglas para ello. Pero, a pesar de que son muchos los trabajos que critican su uso, es cierto que a nivel práctico se sigue utilizando.
- **Glosario y ontologías:** La diversidad de personas que forman parte de un proyecto software hace que sea necesario establecer un marco de terminología común. Esta necesidad se vuelve más patente en los sistemas de información web puesto que el equipo de desarrollo en ellas suele ser más interdisciplinario [18]. Por esta razón son muchas las propuestas que abogan por desarrollar un glosario de términos en el que se recogen y definen los conceptos más relevantes y críticos para el sistema. En esta línea se encuentra también el uso de ontologías, en las que no sólo aparecen los términos, sino también las relaciones entre ellos. Ninguna de las metodologías para el entorno web incluidas en este estudio comparativo por su relevancia en la ingeniería de requisitos propone la definición de una ontología. Por ello la técnica ontología no ha sido incorporada en la tabla 2.
- **Plantillas o patrones:** Esta técnica, recomendada por varios autores [7][9], tiene por objetivo el describir los requisitos mediante el lenguaje natural pero de una forma estructurada. Una plantilla es una tabla con una serie de campos y una estructura predefinida que el equipo de desarrollo va cumplimentando usando para ello el lenguaje del usuario. Las plantillas eliminan parte de la ambigüedad del lenguaje natural al estructurar la información; cuanto más estructurada sea ésta menos ambigüedad ofrece. Sin embargo, si el nivel de detalle elegido es demasiado detallado, el trabajo de rellenar las plantillas y mantenerlas puede ser demasiado tedioso.
- **Escenarios:** La técnica de los escenarios consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos [22]. La representación del escenario puede variar dependiendo del autor. Esta representación pueden ser casi textuales o ir encaminada hacia representaciones gráficas en forma de diagramas de flujo [34]. El análisis de los escenarios, hechos de una forma u otra, pueden ofrecer información importante sobre las necesidades funcionales de sistema [23].
- **Casos de uso:** Es como técnica de definición de requisitos como más ampliamente han sido aceptados los casos de uso. Actualmente se ha propuesto como técnica básica del proceso RUP [19]. Sin embargo, son varios los autores que defienden que pueden resultar ambiguos a la hora de definir los requisitos [6][33][14] por lo que hay propuestas que los acompañan de descripciones basadas en plantillas o de diccionarios de datos que eliminen su ambigüedad.
- **Lenguajes Formales:** Otro grupo de técnicas que merece la pena resaltar como extremo opuesto al lenguaje natural, es la utilización de lenguajes formales para describir los requisitos de un sistema. Las especificaciones algebraicas como ejemplo de técnicas de descripción formal, han sido aplicadas en el mundo de la ingeniería de requisitos desde hace años [28]. Sin embargo, resultan muy complejas en su utilización y para ser entendidas por el cliente. El mayor inconveniente es que no favorecen la comunicación entre cliente y analista. Por el contrario, es la representación menos ambigua de los requisitos y la que más se presta a técnicas de verificación automatizadas.

2.3 La validación de requisitos

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea [23]. Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de requisitos son correctos. Pocas son las propuestas existentes que ofrecen técnicas para la realización de la validación y muchas de ellas consisten en revisar los modelos obtenidos en la definición de requisitos con el usuario para detectar errores o inconsistencias. Aún así, existen algunas técnicas que pueden aplicarse para ello:

- **Reviews o Walk-throughs:** Está técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante.
- **Auditorías:** La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una checklist predefinida o definida a comienzos del proceso, es decir que sólo una muestra es revisada.
- **Matrices de trazabilidad:** Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo [7]. Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.
- **Prototipos:** Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario [26]. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

3. Tratamiento de Requisitos en Propuestas para la Web

El desarrollo de sistemas de aplicaciones web, agrupa una serie de características que lo hacen diferente del desarrollo de otros sistemas [18]. Por un lado, hay que tener en cuenta que roles muy diferentes de desarrolladores participan en el proceso: analistas, clientes, usuarios, desarrolladores, diseñadores gráficos, expertos en multimedia y seguridad, etc. Por otro lado, la existencia en estos sistemas de un importante estructura de navegación obliga a un desarrollo preciso de este aspecto que garantice que el usuario no se “pierde en el espacio navegacional del sistema” [28]. Estas ideas unidas al hecho que los sistemas web suelen tratar con múltiples medios y es esencial que ofrezcan una interfaz adecuada en cada momento, obligan a que estos aspectos propios de la web deban ser tratados de una forma especial en el proceso de desarrollo.

Estas características especiales también hay que tenerlas en cuenta en la fase de especificación de requisitos [8]. Por ello, la mayoría de las propuestas estudiadas ofrecen diferentes clasificaciones de los requisitos. Sin embargo, la terminología usada no es siempre la misma. Para facilitar la comprensión de las propuestas, antes de presentarlas, presentamos una clasificación de requisitos relevantes en sistemas web.

- *Requisitos de datos*, también denominados requisitos de contenido, requisitos conceptuales o requisitos de almacenamiento de información. Éstos requisitos responden a la pregunta de qué información debe almacenar y administrar el sistema.
- *Requisitos de interfaz (al usuario)*, también llamados en algunas propuestas requisitos de interacción o de usuario. Responden a la pregunta de cómo va a interactuar el usuario con el sistema.
- *Requisitos navegacionales*, recogen las necesidades de navegación del usuario.
- *Requisitos de personalización*, describen cómo debe adaptarse el sistema en función de qué usuario interactúe con él y de la descripción actual de dicho usuario.
- *Requisitos transaccionales o funcionales internos*, recogen qué debe hacer el sistema de forma interna, sin incluir aspectos de interfaz o interacción. También son conocidos en el ambiente web como requisitos de servicios.
- *Requisitos no funcionales*, son por ejemplo los requisitos de portabilidad, de reutilización, de entorno de desarrollo, de usabilidad, de disponibilidad, etc.

En este apartado se van a presentar aquellas técnicas que incluyen el proceso de definición de requisitos dentro del ciclo de vida. Alguna de las propuestas que describimos cubrían la captura y definición de requisitos desde sus primeras propuestas. Otras, como se verán, la han incluido en revisiones a sus trabajos iniciales. El criterio que se ha seguido para presentar las propuestas es el cronológico, desde la primera publicación que incluye tratamiento de requisitos. Ello nos permitirá en la comparativa evaluar la evolución de las propuestas para requisitos.

3.1 WSDM: Web Site Design Method

WSDM[5] es una propuesta para el desarrollo de sitios web, en la que el sistema se define en base a los grupos de usuarios. Su proceso de desarrollo se divide en cuatro fases: modelo de usuario, diseño conceptual, diseño de la implementación e implementación. La fase que más repercusión tiene para este trabajo es la primera en la que intenta detectar los perfiles de usuarios para los cuales se construye la aplicación. Para ello, se deben realizar dos tareas:

- *Clasificación de usuarios*: en este paso se deben identificar y clasificar a los usuarios que van a hacer uso del sistema. Para ello, WSDM propone el estudio del entorno de la organización donde se vaya a implantar el sistema y los procesos que se vayan a generar, describiendo las relaciones entre usuarios y actividades que realizan estos usuarios. Para la representación gráfica de estas relaciones WSDM propone una especie de mapas de conceptos de roles y actividades.
- *Descripción de los grupos de usuarios*: en esta segunda etapa se describen con más detalles los grupos de usuarios detectados en la etapa anterior. Para ello, se debe elaborar un diccionario de datos, en principio con formato libre, en el que indican los requisitos de almacenamiento de información, requisitos funcionales y de seguridad para cada grupo de usuarios.

El resto de las fases del proceso de WSDM se hacen en base a la clasificación de usuarios que se realiza en esta primera etapa.

3.2 SOHDM: Scenario-based Object-Oriented Hypermedia Design Methodology

Esta propuesta [21] presenta la necesidad de disponer de un proceso que permita capturar las necesidades del sistema. Para ello, propone el uso de escenarios. El proceso de definición de requisitos parte de la realización de un diagrama de contexto tal y como se propone en diagramas de flujos de datos (DFD) de Yourdon [36]. En este diagrama de contexto se identifican las entidades externas que se comunican con el sistema, así como los eventos que provocan esa comunicación. La lista de eventos es una tabla que indica en qué eventos puede participar cada entidad. Por cada evento diferente SOHDM propone elaborar un escenario. Éstos son representados gráficamente mediante los denominados

SACs² (Scenario Activity Chart). Cada escenario describe el proceso de interacción entre el usuario y el sistema cuando se produce un evento determinado especificando el flujo de actividades, los objetos involucrados y las transacciones realizadas. SOHDM propone un proceso para conseguir a partir de estos escenarios el modelo conceptual del sistema que es representado mediante un diagrama de clases. El proceso de SOHDM continua reagrupando estas clases para conseguir un modelo de clases navegacionales del sistema.

3.3 RNA: *Relationship-Navigational Analysis*

RNA [3] plantea una secuencia de pasos centrándose fundamentalmente en el flujo de trabajo de análisis. El proceso de trabajo que presenta RNA se basa en la realización de las siguientes fases:

- Fase 1- Análisis del entorno: El propósito de esta fase es el de estudiar las características de la audiencia. Consiste en determinar y clasificar a los usuarios finales de la aplicación en grupos según sus perfiles.
- Fase 2- Elementos de interés: En esta fase se listan todos los elementos de interés de la aplicación. Por elementos de interés se entienden los documentos, las pantallas que se van a requerir, la información, etc.
- Fase 3- Análisis del conocimiento: Esta fase consiste en desarrollar un esquema que represente a la aplicación. Para ello RNA propone identificar los objetos, los procesos y las operaciones que se van a poder realizar en la aplicación, así como las relaciones que se producen entre estos elementos.
- Fase 4- Análisis de la navegación: En esta fase el esquema obtenido en la fase anterior es enriquecido con las posibilidades de navegación dentro de la aplicación.
- Fase 5- Implementación del análisis: Una vez obtenido el esquema final en el que ya se encuentran incluidos los aspectos de navegación, se pasa el esquema a un lenguaje entendible por la máquina.

La propuesta de RNA es quizás una de las que más ha resaltado la necesidad de trabajar con la especificación de requisitos, incluyendo tareas como el análisis del entorno y de los elementos de interés. Además resulta interesante pues plantea la necesidad de analizar los requisitos conceptuales de manera independiente a los navegacionales.

3.4 HFPM: *Hypermedia Flexible Process Modeling*

La propuesta de HFPM[25] describe un proceso detallado que cubre todo el ciclo de vida de un proyecto software. HFPM propone un total de trece fases para las cuales se especifican a su vez una serie de tareas. Para este estudio es principalmente relevante la primera fase denominada de modelado de requisitos cuyas tareas son las siguientes:

- Descripción breve del problema. No indica ninguna técnica concreta pudiendo realizarse esta descripción mediante el lenguaje natural.
- Descripción de los requisitos funcionales mediante casos de uso.
- Realizar un modelo de datos para esos casos de uso, proponiendo el uso de un modelo de clases.
- Modelar la interfaz de usuario. Para ello, propone el uso de sketches y prototipos que permitan presentar los datos al usuario.
- Modelar los requisitos no funcionales. En éstos incluyen la navegación, la seguridad, etc.

Se puede observar que la propuesta de HFPM ofrece mayor detalle a la hora de realizar el tratamiento de los requisitos. Sin embargo, no ofrece técnicas concretas, especialmente a la hora de trabajar con los requisitos no funcionales.

3.5 OOHDM: *Object Oriented Hypermedia Design Model*

OOHDM es una propuesta metodológica ampliamente aceptada para el desarrollo de aplicaciones de la web [31]. En sus comienzos no contemplaba la fase de captura y definición de requisitos, pero actualmente propone el uso de user interaction diagrams (UIDs) [33]. Esta propuesta parte de los casos de uso que considera una técnica muy difundida, ampliamente aceptada y fácilmente entendible por los usuarios y clientes no expertos, pero que resulta ambigua para el equipo de desarrollo en fases posteriores del ciclo de vida. Igualmente, resalta la necesidad de empezar el diseño del sistema, especialmente en los entornos web, teniendo un claro y amplio conocimiento de las necesidades de interacción, o lo que es lo mismo de la forma en la que el usuario va a comunicarse con el sistema.

Partiendo de estas dos premisas, OOHDM propone que la comunicación con el usuario se realice utilizando los casos de uso y a partir de ellos los analistas elaboran los UIDs. Estos UIDs son modelos gráficos que representan la interacción entre el usuario y el sistema, sin considerar aspectos específicos de la interfaz. El proceso de transformación de un caso de uso a un UIDs es descrito detalladamente en la propuesta, y se basa en detectar la interacción necesarias para la realización del caso de uso. OOHDM centra el desarrollo de un sistema de información web entorno del modelo conceptual de clases. Este diagrama debe surgir de los requisitos que se definan del sistema, pero los casos de uso resultan demasiado ambiguos para ello. Así, propone refinar el proceso de desarrollo descrito en UML de forma que de los casos de uso se generen los UIDs que concreten más la definición de los requisitos para, a partir de ellos,

² La descripción de la representación de los SACs está fuera de los límites de este trabajo, ver [21].

obtener el diagrama conceptual. En algunos de los primeros trabajos OOHDMM propone la descripción de escenarios en forma textual y gráfica para cada tipo de usuario como etapa previa al diseño de la navegación.

3.6 UWE: UML-Based Web Engineering

UML-Based Web Engineering (UWE) es una propuesta metodológica basada en el Proceso Unificado [16] y UML para el desarrollo de aplicaciones web [12][18]. UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrandose además su atención en aplicaciones personalizadas (adaptivas). Para este trabajo, nos interesa principalmente analizar la propuesta de captura de requisitos de UWE. Esta metodología distingue entre la tarea de elicitar requisitos, definir y validar los requisitos. El resultado final de la captura de requisitos en UWE es un modelo de casos de uso acompañado de documentación que describe los usuarios del sistema, las reglas de adaptación, los casos de uso y la interfaz. UWE clasifica los requisitos en dos grandes grupos: funcionales y no funcionales. Los requisitos funcionales tratados por UWE son:

- requisitos relacionados con el contenido
- requisitos relacionados con la estructura
- requisitos relacionados con la presentación
- requisitos relacionados con la adaptación
- requisitos relacionados con los usuarios

Además, UWE propone como técnicas apropiadas para la captura de los requisitos de un sistema web las entrevistas, los cuestionarios y los checklists y los casos de uso, los escenarios y el glosario para la definición de requisitos. Para la validación propone walk-throughs, auditorías y prototipos.

3.7 W2000

W2000 [1] supone una propuesta que amplía la notación de UML con conceptos para modelar elementos de multimedia heredados de la propuesta HDM (Hypermedia Design Model)[11]. El proceso de desarrollo de W2000 se divide en tres etapas: análisis de requisitos, diseño de hipermedia y diseño funcional. El primero de ellos es el que resulta interesante para este trabajo. La especificación de requisitos en W2000 se divide en dos subactividades: análisis de requisitos funcionales y análisis de requisitos navegacionales. La especificación de requisitos comienza haciendo un estudio de los diferentes roles de usuario que van a interactuar con el sistema. Cada actor potencialmente distinto tendrá su modelo de requisitos de navegación y de requisitos funcionales. El modelo de requisitos funcionales es representado como un modelo de casos de uso tal y como se propone en UML. En él se representa la funcionalidad principal asociada a cada rol y las interacciones que se producen entre el sistema y cada rol. El segundo modelo consiste en otro diagrama de casos de uso pero que no representa funcionalidad sino posibilidades de navegación de cada actor. La representación gráfica es realizada con una extensión de UML.

3.8 UWA: Ubiquitous Web Applications

UWA ha nacido de la colaboración entre diferentes grupos de trabajo, por lo que resulta realmente una agrupación de propuestas y técnicas. En concreto, la propuesta de W2000 se encuentra incluida en UWA. Sin embargo, W2000 ha sido incluida en UWA sólo en la fase de diseño hipermedia, siendo ambas propuestas diferentes en la fase de definición de requisitos. Por esta razón han sido incluidos en este trabajo en forma separada. El proceso de captura de requisitos en UWA [32] comienza definiendo los diferentes roles de usuario que pueden interactuar con el sistema, los objetivos globales del sistema y la relación entre éstos. El proceso continúa haciendo un refinamiento de esos objetivos globales, concretándolos en subobjetivos. Estos subobjetivos son estudiados y refinados para detectar conflictos entre ellos. De esta forma, se concretizan aún más dividiéndolos en requisitos. Los requisitos son clasificados en varios tipos: de contenido, de estructura de contenido, de acceso, de navegación, de presentación, de operaciones de usuario y de operaciones del sistema. De esta forma, los requisitos se van refinando hasta que solo pertenezcan a uno de estos grupos. Y finalmente los requerimientos son asignados a artefactos de diseño o a reglas de customización. Para definir los objetivos, UWA propone una notación propia, basada en una plantilla. La definición de los actores y la relación con los objetivos se hace usando un diagrama basado en casos de uso. Por último, para definir y refinar los subobjetivos y los requisitos, utilizan una notación gráfica propia que denominan *grafo de refinamiento de objetivos*, el refinamiento de este grafo permite ir representando la relación entre los requisitos y hacer un seguimiento para validar la consecución de los objetivos del sistema. Una vez que los requisitos son detectados, hacen uso de XML para definirlos de una manera formal.

3.9 NDT - Navigational Development Techniques

NDT (Navigational Development Techniques)[9] es una técnica para especificar, analizar y diseñar el aspecto de la navegación en aplicaciones web. Para este trabajo, solo es relevante la propuesta que ofrece para la definición y captura de requisitos. El flujo de especificación de requisitos de NDT comienza con la fase de captura de requisitos y estudio del entorno. Para ello, plantea el uso de técnicas como las entrevistas o el brainstorming y JAD. Tras esta fase, se propone la definición de los objetivos del sistema. En base a estos objetivos, el proceso continúa definiendo los requisitos que el sistema debe cumplir para cubrir los objetivos marcados. NDT clasifica los requisitos en:

- Requisitos de almacenamiento de información
- Requisitos de actores
- Requisitos funcionales
- Requisitos de interacción, representados mediante:
 - Frases, que recogen cómo se va a recuperar la información del sistema utilizando un lenguaje especial denominado BNL (Bounded Natural Language) [4].
 - Prototipos de visualización, que representan la navegación del sistema, la visualización de los datos y la interacción con el usuario.
- Requisitos no funcionales

Todo el proceso de definición y captura de requisitos y objetivos que propone NDT se basa principalmente en plantillas o patrones, pero también hace uso de otras técnicas de definición de requisitos como son los casos de uso y los glosarios. La propuesta ofrece una plantilla para cada tipo de requisito, lo que permite describir los requisitos y objetivos de una forma estructurada. Algunos de los campos de los patrones son cerrados, es decir, solo pueden tomar valores predeterminados. Estos campos permiten que en el resto del proceso del ciclo de vida de NDT se puedan conseguir resultados de forma sistemática. El flujo de trabajo de especificación de requisitos termina proponiendo la revisión del catálogo de requisitos y el desarrollo de una matriz de trazabilidad que permite evaluar si todos los objetivos han sido cubiertos en la especificación.

La propuesta viene acompañada de una herramienta case, NDT-Tool, que facilita la cumplimentación de los patrones y que permite automatizar el proceso de consecución de resultados.

3.10 *Design-driven Requirements Elicitation*

Design-driven Requirements Elicitation es parte del proceso design-driven que proponen Lowe y Eklund[24] para el desarrollo de aplicaciones en el entorno Web. La propuesta consiste en realizar la captura, definición y validación de requisitos durante el proceso de diseño. Ello hace necesario que las actividades de diseño sean realizadas de modo que los requerimientos pueden ser tratados y administrados. El proceso se basa en el uso de prototipos para ayudar al cliente en la exploración de las posibles soluciones y de los problemas que tienen que ser resueltos. Los usuarios o clientes definen sus requerimientos basándose en la observación o trabajo con estos prototipos. Es un proceso iterativo que consiste en reducir la incertidumbre del cliente. El ciclo tiene tres fases: evaluación, especificación y construcción.

Este proceso fue definido sobre la base de un exhaustivo análisis de “best practices” en el desarrollo de aplicaciones comerciales para el entorno Web. Esta metodología trata a todos los requisitos de la misma manera; estos requisitos son: de contenido, de protocolo de interfaces, de estructura navegacional, de “look and feel”, de representación interna de datos, de versionamiento, de control de cambios, de seguridad, de gestión de contenido, de acceso de control, de eficiencia, de monitoreo del usuario, de soporte de funcionalidad, de adaptación del sistema, de identificación del usuario y sus derechos de acceso, etc. En las tablas comparativas usamos la abreviatura DDDP para design-driven development process (los autores no usan esta forma abreviada).

4. Comparativa

Una vez enunciadas las propuestas, se van a presentar una serie de estudios comparativos de las mismas. Los estudios que se presentan se basan en tres aspectos. El primero analiza qué requisitos son cubiertos en cada metodología. El segundo estudio presenta las fases dentro del proceso de tratamiento de requisitos que cada una afronta y las técnicas que para ello proponen. El último estudio, evalúa como de detallada es cada propuesta a la hora de definir su proceso de desarrollo, las técnicas que aplica y los resultados que propone obtener. Por último, se presentan algunos otros aspectos que han surgido durante la realización de las comparativas.

4.1 *Requisitos tratados*

En base a la clasificación de requisitos que se realizaron al principio del apartado 3, la primera comparativa que se va a realizar de las propuestas estudiadas consisten en ver qué tipos de requisitos contempla cada propuesta. En la tabla 1 se presentan los diferentes requisitos y se indica cuáles de ellos son tratados en cada metodología.

Analizando los resultados, y teniendo en cuenta que las propuestas están ordenadas por orden cronológico, resulta interesante observar el avance que han tenido los requisitos en el entorno de la web. Las primeras propuestas estaban más centradas en los requisitos de datos y los de interfaz de usuario. Observando la tabla puede verse que las propuestas más actuales resaltan la necesidad de tratar los requisitos de personalización y navegación, así como los transaccionales de forma independiente. Esta idea de separación de conceptos se observó desde el principio en las propuestas para la web (HDM[11], EORM [20], OOHDM [31], etc.) pero solo se planteaba en fases avanzadas del proceso de desarrollo, principalmente en diseño. Se puede observar la tendencia actual de partir de esta separación de concepto ya en la fase de especificación de requisitos.

	Req. datos	Req. interfaz al usuario	Req. navegacionales	Req. personalización	Req. transaccionales	Req. no funcionales
WSDM	✓			✓		✓
SOHDM	✓	✓			✓	
RNA	✓	✓	✓		✓	
HFPM	✓	✓	✓			✓
OOHDM	✓	✓	✓			
UWE	✓	✓	✓	✓		✓
W2000			✓	✓	✓	
UWA	✓	✓	✓	✓	✓	
NDT	✓	✓	✓	✓	✓	✓
DDDP	✓	✓	✓	✓	✓	✓

Tabla 1: Tipos de requisitos contemplados en cada propuesta

También resulta necesario resaltar, aunque no pueda derivarse directamente de la tabla, el hecho de la multiplicidad de términos. La mayoría de las propuestas trabajan con los mismos conceptos pero la terminología usada para nombrarlos es diferente, lo que hace bastante complejo el comparar lo que cada una de ellas propone. Un esfuerzo interesante sería el de unificar la terminología.

4.2 Técnicas y actividades contempladas

Otro estudio interesante surge si se observa en qué fases de las que se presentaron en la figura 1 pone énfasis cada propuesta. Para realizar esta comparativa, se presentan en la tabla 2 las técnicas para capturar, definir y validar requisitos que fueron enumeradas en el apartado 2 del trabajo. Para cada propuesta se indica qué técnicas son descriptas o enumeradas de forma explícita en su proceso de desarrollo. En algunos puntos de la tabla se han indicado las técnicas concretas que proponen, si las mismas no son estándares o son variantes de técnicas conocidas.

		WSDM	SOHDM	RNA	HFPM	OOHDM	UWE	W2000	UWA	NDT	DDDP
Captura	Entrevistas	✓		✓			✓			✓	✓
	JAD									✓	
	Brainstorming									✓	
	Concept Mapping	Rol-Actividad									
	Casos de uso					✓					
	Cuestionario/Checklist						✓				
	Prototipos										✓
	Otra técnicas		DFD								
Definición	Lenguaje natural	✓		✓	✓						
	Glosarios				✓		✓			✓	
	Patrones/plantillas								✓	✓	
	Escenarios		SAC				✓				
	Casos de uso				✓	✓	✓	✓	✓	✓	
	Lenguaje formal								XML		
	Sketches de interfaz				✓						
	Prototipos										✓
	Otras técnicas		Lista evento			UIDs			Grafo requisitos	Frases en BNL	
Validación	Reviews/Walk-throughs						✓			✓	
	Auditorías						✓				
	Matriz trazabilidad									✓	
	Prototipos				✓		✓				✓
	Otras técnicas								Grafo requisitos		

Tabla 2: Técnicas contempladas en cada propuesta

De esta tabla comparativa se pueden sacar varias conclusiones. Por un lado, es necesario indicar que dentro de la fase de captura de requisitos, la técnica más enunciada es la de las entrevistas. Por otro, se puede observar que hay muchas propuestas que no recomiendan ninguna técnica concreta para la realización de la captura. Normalmente estas propuestas tratan el proceso de determinación de requisitos como parte de la definición de requisitos.

Con respecto a la definición de requisitos, se puede observar que es el aspecto central del tratamiento de requisitos para todas las propuestas. Se puede concluir en que existe una tendencia a usar la técnica de casos de uso como base. Sin embargo, ante esto conviene decir que hay dos opiniones encontradas. Algunas propuestas consideran los casos de uso una técnica óptima para representar los requisitos, como es el caso de UWE o HFPM. Sin embargo, hay otras como OOHDM o NDT que aunque indican que es una buena técnica, resaltan que es ambigua y que es necesario obtener modelos más concretos para sistematizar más el resto del proceso de ciclo de vida.

Otra conclusión muy relevante que se obtiene de este estudio es el hecho de la poca importancia que las propuestas han prestado a la validación de requisitos. Las técnicas de validación que proponen se basan principalmente en la revisión de los modelos y resultados de la definición de requisitos. La gran mayoría de las propuestas ni siquiera contemplan esta fase en su proceso de ingeniería de requisitos.

4.3 Grado de detalle en las descripciones

El siguiente estudio que se presenta se basa en la forma en que definen las metodologías el tratamiento de requisitos. Para ello fueron comparadas las descripciones de las propuestas. La definición de algunas propuestas se basan principalmente en describir de forma concreta el proceso de desarrollo, sin llegar a concretar qué técnicas aplicar o la estructura de los resultados que se deben obtener. De esta forma, se van a evaluar las propuestas analizando:

- Si son *orientadas al proceso* o lo que es lo mismo, es si describen un proceso claro y concreto a seguir para realizar la definición, captura y validación de requisitos.
- Si son *orientadas a la técnica* o dicho de otro modo si describen técnicas concretas a aplicar durante el proceso.
- Si son *orientadas al resultado*, es decir, si da una descripción concreta del producto que se debe obtener tras el proceso de desarrollo

Evaluando la definición de las propuestas, se han catalogado cada una de ellas en base a lo detalladas que son a la hora de describir el proceso de definición de requisitos, las técnicas a aplicar o los resultados a obtener. Los resultados de esta evaluación se pueden ver en la tabla 3. Hay que tener en cuenta que la valoración se hace solo con respecto a la fase de ingeniería de los requisitos.

El peso dado a cada técnica se corresponden con la siguiente evaluación:

- Orientación al proceso: describe claramente los pasos a seguir (+), describe el proceso sin detallarlo (o), no describe ningún proceso (-).
- Orientación a la técnica: describe claramente las técnicas y la forma de aplicarlas (+), enumera las técnicas a aplicar (o), no propone ninguna técnica concreta o referencia a técnicas generales (-).
- Orientación al producto: describe claramente la estructura del producto a obtener (+), describe el contenido del producto sin entrar en detalle de su estructura (o), no comenta nada sobre el producto resultante (-).

	Orientación al proceso	Orientación a la técnica	Orientación al resultado
WSDM	o	-	-
SOHDM	-	+	-
RNA	+	-	-
HFPM	+	o	+
OOHDM	o	+	-
UWE	+	o	o
W2000	o	o	-
UWA	+	o	-
NDT	o	+	+
DDDP	+	o	-

Tabla 3: Orientación de las propuestas

Agrupando los valores descritos en la tabla 3, se pueden representar los resultados mediante los gráficos de la figura 2. Estos gráficos han sido incluidos por lo fácil que es sacar algunas importantes conclusiones de ellos.

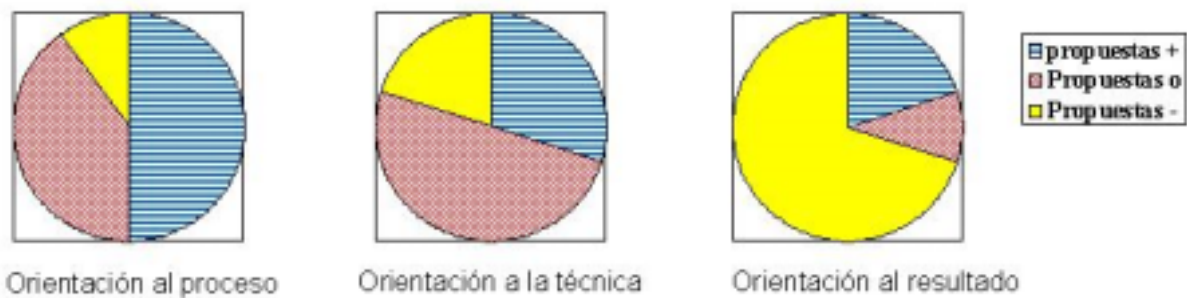


Figura 2: Representación gráfica del grado de detalle de las propuestas

La primera es que ninguna propuesta describe totalmente el proceso, las técnicas y el formato del producto a conseguir. Por otro lado, se puede ver la gran tendencia al proceso que existe. Esta tendencia llega a su límite en propuestas, como RNA, que sólo describen el proceso sin llegar a enunciar técnicas o el formato del resultado a obtener.

Otra importante conclusión es la poca importancia que se da en las propuestas al resultado a obtener. Muchas de ellas sólo enumeran los modelos a conseguir pero no indican la estructura de los documentos a obtener. Solo HFPM y NDT entran a describir el documento de requisitos con detalles. Esta descripción es adecuada pues sirve de soporte al equipo de desarrollo de la estructura del documento, pero no debe ser muy cerrada para que pueda adecuarse a las necesidades, formatos o restricciones de cada entorno de trabajo.

4.4 Otros aspectos

Del estudio comparativo y del trabajo de análisis realizado para realizar el estudio comparativo, se ha podido observar que existen otras ideas que no se han podido incluir en tablas pues supondría una valoración subjetiva de cada propuesta y nos ha parecido oportuno. Si que creemos que sería interesante resaltarlas.

Por un lado, se ha observado que las propuestas presentadas, sobretodo en la fase de definición de requisitos van encaminadas hacia dos tendencias. Por un lado, propuestas como UWE, en las que la definición de requisitos se hace de una manera totalmente gráfica, a propuestas que tienen una tendencia de representación más textual, como es el caso de SOHDM y NDT. La aplicación de una misma técnica en una propuesta u otra puede ser muy diferente dependiendo de la tendencia de la misma. Ejemplo de esto puede ser la variación entre representar un escenario en UWE o en SOHDM.

Otra idea interesante es el comprobar la falta de soporte con herramientas CASE que brindan apoyo para el tratamiento de requisitos en aplicaciones para la web. Realmente el desarrollo de sitios web se encuentra con muy poco soporte para la realización de requisitos, existiendo solo algunas aproximaciones como ArgoUWE para UWE, la herramienta que se ha desarrollado en el proyecto UWA o NDT-Tool para NDT.

5. Conclusiones

Este trabajo ha presentado el estado del arte de las propuestas que se ofrecen en las metodologías para la web en el proceso de definición de requisitos. Se ha comenzado planteando la estructura básica de dicho proceso y las técnicas más comunes aplicadas de forma clásica en la ingeniería de requisitos. Tras esto, se han enumerado las propuestas para la web que cubren esta fase de ingeniería del software y se han comparado desde diferentes puntos de vista.

Como resultado de este estudio podemos afirmar que aún hay mucho potencial de investigación en el campo de ingeniería de requisitos para aplicaciones para el entorno web. Si se compara este trabajo con otros estudios comparativos de propuestas de la web (Koch, 1999, Barry & Lang, 2001 y Escalona, Mejías & Torres, 2002), se pueden observar la gran mayoría de las metodologías que existen son centradas en el diseño de sistemas web en comparación con las que contemplan la especificación de requisitos.

Agradecimientos

Queremos agradecer los comentarios y aportaciones del Prof. Dr. Martin Wirsing y de Andreas Kraus del Institut für Informatik de la Ludwig-Maximilians-Universität München y de los Drs. Manuel Mejías, Jesús Torres y Miguel Toro del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla.

Referencias

1. Baresi L., Garzotto F., Paolini P (2001). *Extending UML for Modelling Web Applications*. In proceedings of the 34th annual Hawaii International Conference on System Science. IEEE Computer Society.
2. Barry, C. & Lang, M. (2001) *A Survey of Multimedia and Web Development Tecniques and Methodology Usage*. IEEE Multimedia. April-June 2001, 52-56
3. Bieber M., Galnares, R., Lu, Q. (1998). *Web engineering and flexible hypermedia*. The Second Workshop on Adaptive Hypertext and Hypermedia, Hypertext.
4. Brisaboa, N. R., Penabad, M. R., Places, A. S., Rodríguez, F. J. (2001) A Documental Database Query Language. *String Processing and Information Retrieval -SPIRE 2001*.
5. De Troyer, O., Leune, C. (1997). *WSDM: A User Centered Design Method for Web Sites*. Tilburg University, Infolab. Belgium.
6. Díez A.(2001). *IRqA y el desarrollo de proyectos: Experiencias Prácticas*. I Jornadas de Ingeniería de Requisitos Aplicadas. JIRA 2001. Seville, Spain.
7. Durán A., Bernárdez, B., Ruiz, A., Toro M. (1999). *A Requirements Elicitation Approach Based in Templates and Patterns*. Workshop de Engenharia de Requisitos. Buenos Aires, Argentina.
8. Escalona, M.J., Mejías, M., Torres, J. (2002). *Methodologies to develop web information systems and comparative analysis*. Informatik/Informatique. núm. 2/2002 de I/I.
9. Escalona, M.J., Torres, J., Mejías, M. (2002). *Requirements capture workflow in Global Information Systems*. Proceedings of OOIS. Springer-Verlag. Montpellier, France.
10. Ferreira, M.J., Loucopoulos, P. (2001). *Organisation of analysis patterns for effective re-use*. Proceedings of the International Conference on Enterprise Information Systems. ICEIS 2001. Setubal, Portugal.
11. Garzoto F., Schwabe D. and Paolini P. (1993) *HDM-A Model Based Approach to Hypermedia Aplication Design*. ACM Trnasactions on Information System, 11 (1), pp 1-26.
12. Hennicker, R., Koch, N. (2000). *A UML-based Methodology for Hypermedia Design*. Lecture Notes in Computer Science. Proc. UML'2000. York, England.
13. IBM OOTC (1997). *Developing Object Oriented Software*. IBM Object Oriented Technology Center. Prentice-Hall.
14. Insrán, E., Pastor, O., Wieringa, R. (2002). *Requirements Engineering-Based Conceptual Modeling*. Requirements Engineering Journal, Vol 7 (1).
15. Jacobson, I. (1995). *Modeling with use cases-Formalizing use-case modelling*. Journal of Object-Oriented Programming.
16. Jacobson I., Booch G., Rumbaugh J. (1999). *The Unified Software Development Process*. Addison Wesley.
17. Koch, N. (1999). *A comparative study of methods for Hypermedia Development*. Technical Report 9905. Ludwig-Maximilian-University, Munich, Germany.
18. Koch, N. (2001). *Software Engineering for Adaptative Hypermedia Applications*. Ph. Thesis, FAST Reihe Softwaretechnik Vol(12), Uni-Druck Publishing Company, Munich. Germany
19. Kruchten, P. (1998). *The Rational Unified Process*. Addison Wesley
20. Lange D. (1995). *An Object-Oriented Design Approach for Developing Hipermedia Information Systems*. Research Report RT00112, IBM Research, Tokyo Research Laboratory, Japan.
21. Lee, H., Lee, C., Yoo, C. (1998). A Scenario-based object-oriented methodology for developing hypermedia information systems. Procesings of 31st Annual Conference on Systems Science. Sprague R.
22. Liu, L., Yu, E. (2001). *From Requirements to Architectural Design using Goals and Scenarios* Proceedings of the 6th Micon Workshop. Canada.
23. Lowe, D., Hall, W. (1999). *Hypermedia and the Web. An Engineering approach*. John Wiley & Son.
24. Lowe D., Eklund J. (2002). Client Needs and the Design Process in Web Projects (2002). WWW2002 Web Engineering Track.
25. Olsina, L. (1998). *Building a Web-based information system applying the hypermedia flexible process modeling strategy*. 1st International Workshop on Hypermedia Development, Hypertext 1998.
26. Olsina, L. (1999). *Metodología cualitativa para la evaluaci3n y comparaci3n de la calidad de sitios web*. Ph. Tesis. Facultad de Ciencias Exactas. Universidad de la Pampa. Argentina.
27. Pan, D., Zhu, D., Johnson, K. (2001). *Requirements Engineering Techniques*. Internal Report. Department of Computer Science. University of Calgary. Canada.
28. Peña, R. (1998). *Diseño de programas. Formalismo y abstracci3n*. Prentice Hall.
29. Raghavan, S., Zelesnik, Ford, G. (1994). *Lectures Notes of Requirements Elicitaci3n*. Educational Materials CMU/SEI-94-EM-10.
30. Retschitzegger, W. & Schwinger, W. (2000). *Towards Modeling of Data Web Applications - A Requirement's Perspective*. Proceedings of the American Conference on Informatin Systems AMCIS 2000, Vol 1, 149-155.
31. Schwabe D., Rossi G. (1998). *Developing Hypermedia Applications using OOHDm*. Workshop on Hypermedia Development Process, Methods and Models, Hypertext '98, Pittsburg, USA.
32. UWA (2001), UWA Requirements Elicitation: Model, Notation, and Tool Architecture. www.uwaproject.org
33. Vilain, P., Schwabe, D., Sieckenius, C. (2000). *A diagrammatic Tool for Representing User Interaction in UML*. Lecture Notes in Computer Science. Proc. UML'2000. York, England.
34. Weidenhaupt, K., Pohl, K., Jarke, M., Haumer, P. (1999). *Scenarios in Systems Development: Current Practice*. IEEE Software. 2, 34-45.
35. UML (2001). Unified Modeling Language. Version 1.4. www.omg.org
36. Yourdon E (1989). *Modern Structured Analysis*. Prentice-Hall