Journal of Web Engineering, Vol. 0, No. 0 (2003) 000–000

REQUIREMENTS IN WEB ENGINEERING: A SYSTEMATIC LITERATURE REVIEW

JOSÉ ALFONSO AGUILAR; IRENE GARRIGÓS; JOSE-NORBERTO MAZÓN
Lucentia Research Group, DLSI, University of Alicante
San Vicente del Raspeig, 03080, Spain.
ja.aguilar@dlsi.ua.es; igarrigos@dlsi.ua.es; jnmazon@dlsi.ua.es

Received (received date)
Revised (revised date)

Requirements engineering is one of the most crucial phases in software development in which designers can attempt to fully satisfy users' needs, and an effective definition of requirements both contributes towards making the right design decisions and helps to support change management, thus improving the quality of the final software product. Web engineering methods should consider a requirement engineering phase that suits the Web's large heterogeneous user community. The objective of this work is to classify the literature with regard to the requirements engineering applied in the Web domain, with which has allowed us to formally obtain the current state-of-the-art. The present work is based on the systematic literature review method proposed by Barbara Kitchenham, we reviewed publications from ACM, IEEE Computer Society Digital Library, Science Direct, DBLP and World Wide Web. From a population of 3059 papers, we identified 43 primary studies that provide information about dealing with requirements in Web engineering.

Keywords: Web Engineering, Requirements Engineering, Systematic Literature Review
Communicated by: to be filled by the Editorial

1 Introduction

A Web system is an application that is invoked with a Web browser commonly over the Internet and it differs from traditional stand-alone software in that their user community is both large and heterogeneous. In this respect, Web applications are designed to be widely accessed, and their may have therefore different needs, goals and preferences. Furthermore, a Web system must satisfy the needs of many types of stakeholders[a]apart from the users themselves, i.e., the persons who maintain the system; the organization wishes the system; and those who fund the system development budget.

Consequently, several special requirements must be considered in the development of Web systems: (i) what information the Web application should offer (content requirements), (ii) which scenarios should be defined in the Web application in order to offer this information (navigational requirements), and (iii) how the user or groups of users should be provided with this information (functional requirements). As a result, the development of a Web system calls

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

for knowledge and expertise from many different disciplines and requires a team of diverse group of people with a high degree of expertise in different areas [2]. This makes the design and development of the Web system further complex and difficult. In this respect, Web engineering methods [3, 4, 5, 6, 7] have provided different mechanisms with which to consider the content, composition, and navigation features of Web applications; including appropriate steps to consider requirements [8].

Bearing these considerations in mind, this paper presents a systematic literature review [9] with which to analyse the current state-of-the-art with regard to requirements engineering in the Web domain in order to reveal their advantages and disadvantages. Roughly speaking, a systematic literature review is useful for formally defining several important issues to obtain those relevant papers related to a research topic: (i) the problem to be addressed, (ii) the information sources, (iii) keywords to search for information, (iv) the inclusion and exclusion criteria to be applied to the papers found in the searches; and (v) the templates used to order and classify the information collected from the papers found. This research technique originated in the field of medical research and was successfully adapted to software engineering by Barbara Kitchenham [9][b].

This paper is an extension of our previous review [17] in which we highlighted the importance of considering requirements in the development of Web systems. In particular, in the present systematic literature review we have improved our previous work as follows.

- The search strategy has been improved, and more methods have been therefore added to the review.

- The study has been focused on analyzing the requirements engineering process in the Web systems development, specifically, in which form the requirements are treated with regard to elicitation, analysis, specification, validation and the management of them.

- How the studied methods have been spreaded in the academic community has been shown.

- Our systematic review analyzes the requirements vocabulary adopted by each Web engineering method in a more methodical and comprehensive manner by using the classification previously proposed by Escalona and Koch [8].

In summary, the aim of this review is to: (i) structure the conceptual basis to deal with the requirements engineering process in the development of Web systems, and (ii) shed light on potential avenues for future research in this area.

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

[b] It is worth noting that the approach followed for conducting the systematic literature review presented in this paper has been inspired by others such as [10, 11, 12, 13, 14, 15, 16]

2 Requirements and Web engineering concepts

Requirements engineering is the process of discovering, analysing, documenting and verifying the services that should be provided by a software system, along with its operational constraints. According to [18], a requirements engineering process can be divided into five steps:

- **Requirements elicitation.** This is the initial step in requirements engineering whose objective is to discover what problem needs to be solved. The main goals of this step are the identification of the *stakeholders*, the objectives a software system must meet, the tasks that users currently perform and those that they might wish to perform. Requirements elicitation is often carried out through the application of various techniques [19, 20, 21], such as (i) traditional techniques (questionnaires, interviews, etc.), (ii) group elicitation techniques (brainstorming or focus groups), (iii) prototyping (e.g., provoking discussion in a group elicitation), (iv) modelling techniques (e.g., goal-based methods such as *i** [22]), (v) cognitive techniques (specially developed for knowledge acquisition), and (vi) contextual techniques (e.g. the use of ethnographic techniques such as participant observation).

- **Requirements analysis.** This step can include the creation of conceptual models or prototypes with which to achieve the completeness of the requirements. Requirements analysis deals with understanding an organization's structure, its business rules, and the goals, tasks and responsibilities of its members and the data that is needed. This stage allows designers to detect and resolve conflicts that may occur among different *stakeholders* [23].

- **Requirements specification.** This step is an integral description of the behavior of the system to be developed. This description aims to facilitate an effective communication of requirements among different *stakeholders*. The most widely used techniques for requirements specification are, among others, templates, scenarios, use case modelling, and natural language [24].

- **Requirements validation.** This step aims to establish whether the requirements and models elicited provide an accurate representation of the actual *stakeholder* requirements. Some techniques for requirements validation are reviews, audit and traceability matrices. A requirement must be traceable, since it is defined throughout the development process. A traceable requirement ensures that change management is able to assess the impact of a change on the rest of the application. It is therefore necessary for *stakeholders* to be able to see that the final *work product*[a]is useful for achieving their requirements, and that each requirement has been derived within a particular work product. To do this, requirements engineering needs to ensure traceability. Requirements traceability refers to the ability to describe and follow the life of a requirement, in both a forward and a backward direction [25]. While forward traceability is related to following the requirement to its final implementation, backward traceability refers to following the *work product* to its source requirement (i.e., the associated requirement

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

that originated it). Interestingly, traceability between requirements and *work products* is a recommendation made by the CMMI (Capability Maturity Model Integration)[b], specifically in CMMI's Level 2 (within the requirements management process area).

- **Requirements management.** This step consists of recognizing changes through continuous requirements elicitation, the re-evaluation of risks, and the evaluation of systems in their operational environment. Requirement management includes techniques and tools for configuration management and version control [26].

After an overview of requirements engineering concepts, it is worth noting that the development of Web applications have some particular requirements that differ from the traditional requirements. These new requirements are defined in the seminal work of Escalona and Koch [8]. In this work, the authors put forward the argument that functional requirements for Web engineering are related to three main features of Web applications: navigational structure, user interface and personalization capability, and that the data structures required by the Web application should also be specified. An overview of each kind of requirement for Web engineering is described below:

- **Content Requirements.** These define the information that is useful for the Web application which should be presented to users. For example, in an online book store some examples might be the information about a "book" or a "book category".

- **Service Requirements.** This refers to the internal functionality that the Web application should provide to its users. Following the online book store example, service requirements might be: "register a new client" or"add book to shopping cart".

- **Navigational Requirements.** A Web application must also define the navigational paths which are available. In this respect, some examples are: the user navigation from "index page" to different options such as"consult products by category" or "consult shopping cart".

- **Layout Requirements.** Requirements can also define the visual interface for users, such as "present a colour style", "multimedia support", or "user interaction", among others.

- **Personalization Requirements.** The designer should specify the desired personalization actions to be performed in the final Web application (e.g., "show recommendations based on interest in previously acquired books", "adapt font for visually impaired users", etc.).

- **Non-Functional Requirements.** These are related to quality criteria that the intended Web application should achieve and that may be affected by other requirements. Some examples might be "good browsing experience", "attract more users", "improve efficiency", etc.

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

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

These types of requirements are considered in the Web engineering methodologies for the definition of several models. Web engineering approaches commonly use five main models to define a Web application [27], namely:

- **Domain model (DM).** This encapsulates the structure and functionality required of the relevant concepts of the application domain and reflects the static part of the Web application.
- **Navigation model (NM).** This model aims to specify the structure and behavior of the navigation view over the domain data defined. It defines each path on which users can navigate through the Web application.
- **Presentation model (PM).** This model defines the layout of the Web application, i. e. style, font color, etc.
- **Personalization model (PM).** Personalization strategies (Adaptive Web Sites) are specified in this model. Web pages are personalized based on the context, the socioeconomic level and the interest of an individual user. Personalization implies that the changes are based on the items purchased or in the pages viewed.
- **User model (UM).** This enables the users to be described in terms of their personal information and their relations with a particular application domain as well as the navigational actions performed at execution-time. The structure of the information needed for the personalization strategies is also described in this model.

3 Reviewing literature about requirements engineering in Web domain

This study has been undertaken as a systematic literature review (SLR) based on the original guidelines proposed by Kitchenham [9]. A systematic literature review is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, topic area or phenomenon of interest. The goal of the review is to analyse the current state-of-the-art with regard to requirements engineering in the Web domain. Although, there are several reasons for performing a systematic literature review, the most common are:

- To sum up the remaining data (information) concerning a treatment of technology.
- To identify any gaps in research in progress, thus highlighting areas of potential interest for investigation.
- To serve as a framework in which to properly place new research activities.

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

- **Planning the review.** This aims to develop a protocol that specifies the plan that the systematic literature review will follow to identify, assess and accumulate evidence.
- **Conducting the review.** This phase is responsible for achieving the protocol planned in the previous phase.

- **Reporting the review.** This phase is responsible for extending the review to the research community. It culminates with the elaboration of a review report.

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

- Planning the review.
  - Constructing the research questions that will guide the analysis of the research.
- Conducting the review.
  - Defining a search strategy.
  - Defining the inclusion and exclusion criteria for the studies.
  - Study quality assessment.
  - Study selection.
  - Data extraction.
  - Analysis of the primary studies.
- Reporting the review.
  - Elaborating a review report (extending the results to others).

The steps performed in our systematic literature review are based on the proposal from [28], these are detailed next.

3.1 Research questions

As mentioned in Section 1, the goal of this systematic literature review is to summarize the information concerning how requirements engineering process is applied in Web engineering, thus detecting avenues for future research. According to the systematic review process [9], the question structure is divided in four aspects (PICO):

- **Population:** people, projects and application types affected by the intervention. They should be directly related to the question.
- **Intervention:** software technology, tool or procedure, which generates the outcomes.
- **Comparison:** some other type of intervention - only if applicable -
- **Outcomes:** technology impact in relevant information terms for practical professionals.

Based on this strategy, the PICO structure for the research questions is described below:

- **Population:** this question refers to the support of the methods to the Web application development process. The population is composed by designers and developers seeking a way to have a more robust process support and by researchers in Web engineering area aiming at the development of new methods.

- **Intervention:** this review must search for indications that the development of Web application can be fully supported by a systematic process.
- **Comparison:** no applicable.
- **Outcomes:** the objective of this study is to demonstrate how the development of Web applications is supported by a systematic process and if the process is not fully supported with regard to requirements engineering area, since there are many gaps in the development of Web applications.

As a result, four research questions (RQ) should be asked, focusing on specific aspects of the evaluation. These are defined next:

1. **(RQ1) Which are the existing methods for the development of Web applications based on a systematic process?** There are several Web engineering approaches for the development of Web applications, but not all of them necessarily cover requirements engineering. It is therefore necessary to detect to what extent different Web engineering approaches cover the requirements engineering phase.

2. **(RQ2) What are the phases of the requirement engineering process of each method?** The main features, advantages and drawbacks of requirements engineering techniques defined for or adapted to the Web engineering field are studied.

3. **(RQ3) Is there any common terminology with regard to requirements engineering applied by the existing methods?** This research question refers to detect the type of requirements defined for each Web engineering method with regard to requirements phase.

4. **(RQ4) Which tools supports the development process?** The main idea of this research question is to analyse those Web engineering methods that offer tools in order to verify the requirements engineering support.

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

3.2 Search strategy

The search strategy should be systematic, comprehensive and explicit in order to make a formal search of primary studies with a predetermined order. According to [9], in this phase it is necessary to use the search engines by applying a combination of search terms (keywords) extracted from the research questions. Various experts should then verify and review the search results. Once the steps to be followed in the search process have been defined, it is necessary to define the resources that are available to conduct the review of primary studies. In this systematic literature review the research sources used are as follows:

- **eJournals (repositories with restricted access):** ACM[c], IEEE Computer Society Digital Library[d], Science Direct[e].
- **Digital library of scientific literature:** DBLP Computer Science Bibliography[f].
- **Research resources in the World Wide Web:** Google Scholar[g].

According to [29], these libraries were chosen because they are some of the most relevant sources in software engineering (see appendix). On the other hand, Google scholar was selected to complete the set of conferences and workshops searched and to seek grey literature in the field (i.e., white papers, technical reports and PhD thesis).

After defining the research questions and the sources to be consulted, the strings to be used in the search were then defined. Based on the structure of the research questions, some keywords were extracted and used to search the primary study sources. We initially had the following keywords: *Web*, *engineering*, *requirements*, *development*, *method* and *tool*. However, in order to obtain more concrete and specific results in the field, we decided to link Web with the keywords *engineering* and *requirements*, *requirements* with the keyword *engineering*, and the keyword *Web* with the keywords *engineering* and *methods*. In this respect, the choice to concatenate "Web" with "engineering" was motivated by our goal, which was to retrieve papers specifically focused on requirements engineering in the Web domain. As a result, the search string "(Web OR WWW OR World-Wide Web OR Internet) AND engineering" was not considered. The search string was used in all instances, even when examining papers from special issues in Web engineering. Also, a list of synonyms was constructed for each of these keywords. Nevertheless, other words were also added to increase the size of potential relevant studies: *system*, *techniques*, *phase*, *design*. These words were linked with the keywords *Web*, *requirements*, *methods*, *engineering* and *tool*.

The search string was thus constructed by using the boolean operators (AND and OR): (*Requirement* OR *requirements* OR *requirements engineering* Or *Web engineering*) OR ((*Web requirements*) OR (*Web system*)) AND (*method* OR (*techniques*)) AND (*Tool support*). However, the results from this search string were too imprecise in the case of research sources such as IEEE and ACM because of by using their search engines were retrieved a few papers. Bearing this consideration in mind, we decided to adapt the search keywords to match each research question and the individual requirements of each of the search engines on our search resources. Based on the SLR proposed by [30], we created a specific type of string for each search engine (using the synonyms list) making the search as accurate and comprehensive as possible.

Moreover, corresponding authors of main texts were e-mailed directly to find out some particular issue about their Web method. Finally, it is important to highlight the fact that we conducted a secondary search based on the references found in our primary studies in order to ensure that we do not leave aside any major work in this area. This idea is based on the SLR performed by [31].

---

[c] http://portal.acm.org
[d] http://ieeexplore.ieee.org
[e] http://www.sciencedirect.com
[f] http://www.informatik.uni-trier.de/ ley/db/index.html
[g] http://scholar.google.com

3.3 Inclusion and Exclusion criteria

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

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

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

Basically, this systematic literature review included any primary studies related to the requirements engineering in the Web field. In this respect, we deemed that at least the part related to requirements specification in Web engineering must be present in each primary study, since we assumed that not all approaches implement another phase of the requirements engineering process such as validation or management.

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

3.4 Study quality assessment

The quality assessment of the selected studies was performed using as indicators the publication place and the diffusion of the methods. The publication place refers to the journals and conferences where the primary studies were published (this applies to Google Scholar which go in search of a wider spectrum of papers such as white papers). The diffusion of the methods corresponds to the academic or professional application of the method, also if the method has a tool support and if the tool is a prototype or a professional tool.

3.5 Study selection

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

Table 1. Search results by source and exclusion criteria.

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

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

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

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

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

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

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

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

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

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

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

3.6 Data extraction

The goal of this phase is to design data extraction forms with which to accurately record the information obtained from the primary studies. The data extraction form must be designed to collect all the information required in order to fully address the research questions [9]. In this phase, the form shown in Table 8 was used to store the information extracted from the search results. This form contains the title of the publication, the journal or workshop in which the paper was published, the publication date, the main author, the requirements engineering process, the shortcomings with regard to the requirements engineering and the tool support.

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

Data synthesis was then performed. Data synthesis involves collating and summarizing the results of the primary studies included. The synthesis is descriptive (non-quantitative) [9] and is carried out by answering the research questions, as in this review (see questions presented in Section 3.1).

4 Analysis of the primary studies

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

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

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

4.1 UML-based Web Engineering (UWE)

UWE [43] is a methodological approach for the development of Web applications which is entirely based on UML. It covers the whole life cycle of the development of Web applications and focuses on adaptive applications.

Development process: UWE is based on Model-Driven Architecture (MDA) [44]. Within MDA, requirements are considered in the CIM (Computational Independent Model) in order

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

Title
Metamodeling the requirements of Web systems [32].
Model transformations from requirements to web system design [33].
Requirements engineering for Web Applications-a comparative study [8].
Introducing requirements traceability support in model-driven development of web applications [34].
The object-oriented hypermedia design model. [35].
Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems [36].
From task-oriented to goal-oriented Web requirements analysis [37].
Transformation techniques in the Model-Driven Development Process of UWE [38].
NDT. A Model-Driven Approach for Web requirements [39].
A requirement Analysis Approach for Using i* in Web Engineering [27].
Web Modeling Language (WebML): a modeling language for designing Web sites [40].
WSDM: a user centered design method for Web sites [41].
Hera: Development of semantic web information systems [42].

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

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

to show what the Web application is expected to do without showing details of how it is implemented. Through this requirements model, UWE can generate content, navigation, presentation and process models. Generation is done by clearly and formally establishing a set of Query/View/Transformation (QVT) transformations[h] to be automatically performed [45]. The authors claim that the main benefit of MDA is that less time and effort are needed to develop the whole Web application, thus improving productivity.

Requirements Engineering:. UWE proposes the use of interviews, questionnaires and checklists as requirements elicitation mechanisms [8]. With regard to requirements specification, UWE models requirements with UML use case diagrams together with textual descriptions of use cases and UML activity diagrams. Use case diagrams are used to represent an overview of the functional requirements, while activity diagrams provide a more detailed view. In recent works [33], UWE has been extended in order to support the UML profile for Web requirements WebRE (Web REquirements metamodel) [32]. This profile has been defined by UWE's authors in a joint work with the authors of NDT (presented below) [32]. Interestingly, UWE considers any kind of requirements according to the taxonomy presented in Section 2: content, service, navigational, layout, personalization, and non-functional requirements. This method does not support the requirements validation (i.e., traceability).

Tool support: A plugin called MagicUWE was developed to be used with the CASE tool MagicDraw[i] [46], which allows UWE to provide a CASE (Computer-Aided Software En-

[h] http://www.omg.org/spec/QVT/1.0
[i] http://www.magicdraw.com

gineering) tool. Unfortunately, requirements analysis is missing in this tool.

It is worth mentioning that in [47] the author proposes an approach called UWE4JSF for the automatic generation of Web applications in JSF (Java Server Faces) derived from UWE models.

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

4.2 Navigational Development Techniques (NDT)

NDT [39] is a model-driven approach that covers the requirements analysis phase in the development of Web applications.

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

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

Requirements Engineering: In the NDT method, the requirements process consists of three phases (capture, definition, and validation) which permit the derivation of three conceptual models (content, navigational and abstract interface). To this aim, NDT applies use case diagrams and a set of textual templates for the requirements description [39]. With regard to the types of requirements, NDT considers content, service, navigational, layout, personalization, and non-functional requirements. In [32], the authors of NDT and UWE have jointly developed a UML-Profile for Web requirements called WebRE. This metamodel is useful for specifying a set of QVT transformations in order to obtain several conceptual models (content, navigational and abstract interface) from requirements specification. NDT supports requirements validation, specifically supports the requirements traceability by means of a requirements matrix. This technique is applied to observe the correspondence between the requirements and artefacts (documents in the requirements phase) that satisfy it [48].

Tool support: NDT has tool support for the requirements engineering phase [48]: NDT-Suite. This is actually a set of tools composed of NDT-Profile, NDT-Driver, NDT Quality and NDT Report.[j]

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

[j] http://www.iwt2.org
[k] http://www.sparxsystems.com/

- NDT Report. This tool obtains the requirements, the analysis, the design and the test documents.

Approach limitations: NDT is not a complete methodology for developing Web applications since it needs to be combined with UWE and WebRE. Importantly, NDT does not consider design and implementation phases, and focuses solely on requirements analysis. NDT hampers the development of a complex Web application, since templates are difficult to complete as they require intensive interviews [32]. Furthermore, the requirements are difficult to maintain in NDT owing to the use of textual templates for their specification. Solving these drawbacks was one of the reasons for developing WebRE metamodel.

4.3 Web Modeling Language (WebML)

WebML [40] is a visual language for specifying the content structure of a Web application and the organization and presentation of such content in a hypertext.

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

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

Requirements Engineering: Although a requirements specification phase is considered in WebML, a requirements analysis phase is missing. Requirements are specified by using UML use case and activity diagrams. WebML considers six types of requirements, namely: content, service, navigational, layout, personalization and non-functional requirements [49]. This approach does not support requirements validation.

Tool support: WebML has tool support: WebRatio[m] [50]. The support that WebRatio offers to WebML is mainly focused on providing facilities for a conceptual modeling phase and for the automatic generation of J2EE code (as opposed to traditional prototyping tools, which generate application mockups). In this context, WebML models are implicitly defined within WebRatio with a document type definition (DTD) and with a UML 2.0 profile. WebML models can therefore be used in conjunction with other notations and can ensure interoperability with other tools.

[l] *http://www.webml.org*
[m] *http://www.webratio.com*

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

4.4 Web Site Design Method (WSDM)

The WSDM method is a user-centered approach [41] since the Web application is defined according to the requirements of the groups of users.

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

- Mission statement, in which the purpose of the Web application must be expressed.
- User modelling, in which the users of the Web applications are classified and grouped.
- Conceptual design, in which a navigational model and a class diagram are designed for the navigational paths.
- Implementation design, whose aim is to create a consistent, pleasing and efficient look and feel for the conceptual design made in the previous phase.
- Implementation of the Web application tailored to a chosen implementation environment.

Requirements Engineering: The initial two phases of WSDM (denominated as mission statement and user modeling) are responsible for managing requirements through techniques such as concept maps (of roles and activities) and the data dictionary for the definition of functional and security requirements. The requirements in this approach are specified in a textual form and are considered to be content, service, navigational, personalization and non-functional requirements [41, 51]. Requirements validation is not supported by WSDM.

Tool support: No tools are publicly available for WSDM. There have been some implementations to support parts of the method but these were never integrated, i.e. a code generation tool called WSDMtool [52].

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

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

OOWS [53] is an object-oriented software production method that provides conceptual modeling extensions (in terms of models and abstraction primitives) to facilitate the Web application specification [54]. OOWS extends the OO-Method [55] by adding navigational and presentation characteristics.

Development process: OOWS has a model-driven development process [56] in which there are two main steps: conceptual modeling and solution development [57].

- Conceptual Modeling (i.e., problem space). The aim of this step is to obtain the system specification. It is divided into three sub-steps:
  1. Functional requirements elicitation: techniques based on use cases and scenarios are applied to build a conceptual schema.
  2. Classic conceptual modeling: structural, functional and dynamic models are used to capture the system structure and its behavior.
  3. Navigation and presentation modeling: a navigational model is built in order to specify navigational requirements from the class diagram. Once the navigational model has been built, presentation requirements are specified by using a presentation model which is strongly based on the navigational model.
- Solution development (i.e., solution space). Some patterns are applied to obtain the Web application from the system specification.

Requirements Engineering: The requirements analysis phase is carried out through a set of strategies, namely: (i) FRT (Functional Refinement Tree), (ii) use cases, and (iii) tasks, task specification and data description diagrams for navigation requirements. Finally, OOWS considers six types of requirements: content, service, navigational, layout, personalization and non-functional requirements [53, 58, 59, 60]. With regard to requirements validation, this approach focuses on tracing requirements (traceability) by using the navigational model in a model-driven process through a set of transformation rules defined by graph theory, thus omitting traceability from requirements to the other models involved in the development process [34].

Tool support: This approach has a development framework called OOWS-Suite [61], which is integrated with the OlivaNova tool[n] to provide support for requirements engineering.

Approach limitations: Task analysis is a technique used in the requirements specification which may be time-consuming, complex and depends largely on the experience of the analyst for its correct implementation. Moreover, according to [37], the users' needs are not necessarily well defined within their own mind so as to be defined as tasks. From the designer's perspective, the user goals are ill-defined (no good criterion is defined to be satisfied), i.e., they are not easily reducible to specific tasks that can be easily mapped onto interface features. Finally, the last limitation of this approach concerns requirements traceability, which is only from the requirements to the navigational model, signifying that the other conceptual models involved in the Web application development process are ignored.

4.6 Adaptive Object Oriented Hypermedia (A-OOH)

The A-OOH method [27], [62] is an extension of the Object-Oriented Hypermedia (OO-H) modelling method [63]. The main difference between A-OOH and OO-H are the personalization capabilities [64] which support the modelling of adaptive Web sites.

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

- Computational Independent Model (CIM). At this level, the requirements are specified by the Web application designer. This is done by means of a graphic editor that al-

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

lows the specification of the Web application requirements through the i^* modelling framework, considering the user goals, needs and the *softgoals*.

- Platform Independent Model (PIM). There are several models at this level: domain, navigational, presentation, user and personalization models.These models are generated from the CIM, and can be further completed.
- Platform Specific Model (PSM). This represents the implementation phase of the Web application.

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

Requirements Engineering: A-OOH uses the i^* framework [22] by means of a UML profile implemented such as ecore metamodel[o] By means of the metamodel, the requirements specification is performed in order to create a requirements model. Conceptual models are then generated by means of model transformations, which are applied automatically. A-OOH applies the taxonomy presented in Section 2 to classify the types of requirements used (content, service, navigational, layout, personalization and non-functional) [27]. With regard to requirements validation, traceability in A-OOH is partially supported [65], since the authors only provide support for traceability from the CIM level (requirements) to the PIM level [66] by means of weaving models [67]. Moreover, this approach has support for impact analysis [68].

Tool support: The development process is supported by the Eclipse[p]development platform [69]. The A-OOH method has an GMF (Graphical Modeling Framework) editor[q]called WebREd (Web Requirements Editor) with which the requirements are specified, once the requirements are specified the Web application conceptual models are generated by means of transformation rules, the rules are implemented by means of the Atlas Transformation Language (ATL[r]).

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

4.7 Object-Oriented Hypermedia Design Method (OOHDM)

The OOHDM [70, 35] is an approach for the development of Web applications based on HDM [71], one of the first methods designed for hypertext or hypermedia applications. This approach was the first to propose the separation of aspects, an idea that has been applied by different Web approaches to date.

Development process: The OOHDM development method considers five phases:

- Requirements gathering. *Stakeholder* requirements are captured in this phase.

[o]http://www.eclipse.org/emf
[p]http://www.eclipse.org/
[q]http://www.eclipse.org/gmf
[r]http://www.eclipse.org/atl

- Conceptual design. The OOHDM approach models the application domain using object oriented modelling constructs and some design patterns.

- Navigational design. In this phase the OOHDM approach takes into account the user profile and the task, and builds the navigational structure of the Web application.

- Abstract interface design. OOHDM describes the interface for navigational objects and defines the layout of interface objects (responses to external events).

- Implementation. The implementation of interface objects.

Requirements Engineering: The first step is to gather the *stakeholders'* requirements. To achieve this, it is first necessary to identify the actors (*stakeholders*) and the tasks they must perform. Next, scenarios are collected (or drafted), for each task and type of actor. The scenarios are then collected to form a use case, which is represented by using user interaction diagrams. These diagrams provide a concise graphical representation of the interaction between the user and the system during the execution of a task. This approach presents guidelines to define conceptual and navigational schemas by means of rules described in natural language, and these rules indicate how the conceptual and navigational schemas can be defined from user interaction diagrams. OOHDM only considers three types of requirements: conceptual, navigational and layout requirements. OOHDM does not support requirements validation.

Tool support: This approach has an environment called OOHDM-Web. OOHDM-Web allows the implementation of hypermedia applications such as CGI scripts which produce dynamically generated pages, whose contents are fed from a database and integrated with pre-defined templates [72]. This tool can be downloaded from the OOHDM wiki.[s]

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

4.8 Hera

Hera [73] is a model-driven methodology that supports the design of Web information systems. This approach focuses on the processes of integration, data retrieval, and presentation generation. These processes lead to data transformations based on RDF (Resource Description Framework) models. Hera is principally focused on the development of Web semantic information systems. This approach has been extended and called Hera-S, which is a Web design method that combines the strength of Sesame (a popular open source RDF framework) and the rich modeling capabilities of Hera [74].

Development process: As with other Web engineering methods, Hera includes a phase in which the hypermedia navigation is specified. The Hera methodology has the following phases [73, 42]:

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

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

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

Requirements Engineering: Hera does not have a well-defined requirements analysis phase, but in [75], the authors propose a specification method and a procedure for the automatic generation of a target model from a workflow model. The workflow model describes the business process in the system and the collaboration with users and external systems by means of a task model, in which the actors' tasks only contain activities providing interaction with the system process. This approach considers requirements of the content, service, navigational, layout and personalization types. This approach does not support requirements validation.

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

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

5 Discussion

This section provides an analysis of the methods studied in the previous section, with the objective of answering the research questions presented in Section 3.1. This analysis is principally focused on the techniques applied to each method in the requirements engineering phase, requirements terminology and tool support.

5.1 Requirements engineering in Web methods (RQ1)

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

[t] http://saxon.sourceforge.net
[u] http://www2002.org/CDROM/refereed/329
[v] http://www.openrdf.org

other hand, NDT focuses solely on the requirements analysis phase and uses the UWE models to cover the entire Web application development process.

5.2 Techniques applied in the requirements engineering process (RQ2)

The techniques applied to each approach in the requirements engineering phase involve a specific set of technologies. Table 9 shows the requirements engineering techniques applied by each Web engineering approach for the analysis and specification of Web requirements. This table shows (i) a trend towards the application of use cases, since this technique is used by OOWS, WebML, NDT, UWE and OOHDM for requirements specifications, and (ii) the persistence of the technique of UML profiles, which is applied by UWE and A-OOH respectively.

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

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

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

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

Table 9. RE techniques by each Web engineering approach.

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

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

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

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

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

With regard to the requirements validation support in the requirements engineering process, the traceability is specifically studied with reference from requirements to the work products involved in the Web application development process. The traceability is a highly important success factor in software engineering for the reason that, during the design phase, requirements traceability allows us to keep track of what happens when a change request is implemented before a system is redesigned (impact analysis). Traceability can also provide information about the justifications, important decisions and assumptions behind requirements [76]. However, one common problem in the requirements engineering phase in the development of Web applications is the absence of requirements traceability in most of the approaches studied in this SLR (Table 10). Only NDT, A-OOH and OOWS cover the traceability issue to some extent. NDT offers traceability support by means of a requirements matrix [48], OOWS focuses on tracing requirements throughout the navigational model by using a set of transformation rules defined by graph theory [34]. However, both approaches NDT and OOWS omit traceability from requirements to the other models involved in the development process. In this respect, the A-OOH method covers traceability from requirements to the conceptual models of the Web application by means of weaving models [66], even though, this method does not cover traceability to the code level. In this respect, some approaches maintain unidirectional traceability between the models generated at the initiation of the development process and the rest of the models involved, but not from requirements specification to the rest of the models involved in a bidirectional manner. We believe that bidirectional traceability could be very useful in a Web engineering development process, particularly during validation with customers.

Finally, but of no less importance, the impact analysis is only considered by the A-OOH method, and it is done by an algorithm implemented to analyze the dependencies among requirements in the A-OOH requirements model [68].

Table 10. Requirement engineering phase and traceability support by Web engineering approach.

Method	Requirements Engineering Support	Traceability Support
NDT	+	+/-
WebML	+	-
WSDM	+	-
UWE	+	-
OOWS	+	+/-
OOHDM	+	-
Hera	-	-
A-OOH	+	-

5.3 Requirements terminology (RQ3)

The research addressing requirements in Web engineering has produced a heterogeneous terminology for requirements which hinders further progress. A unifyied vocabulary proposed by [8] (introduced in Section 2) is therefore provided to shed light on (i) the expressivity of current approaches when considering requirements in Web engineering, and (ii) the correspondences between the types of requirements used by each approach with the aforementioned taxonomy. An overview of this is shown in Table 11.

Table 11. Types of requirements and terminology used by each Web engineering approach.
Type of Requirements

Method	Content	Service	Navigational	Layout	Personalization	NFRs
UWE	Content	Process	Navigation	Presentation	Adaptation	NFRs
NDT	Storage Information	Functional	Interaction	Interaction	Actor	NFRs
WebML	Content	Service	Navigational	Presentation	Personalization	NFRs
WSDM	Content	Functional	Navigational	X	Personalization	Security, Usability
OOWS	Functional	Functional	Navigational	Presentational	Presentation	NFRs
OOHDM	Content	X	Navigational	Layout	X	X
HERA	Content	Service	Navigational	Presentation	Personalization	X
A-OOH	Content	Service	Navigational	Layout	Personalization	NFRs

In the same order of ideas, of all the approaches presented in Section 4, only A-OOH [27], UWE [77], WebML [49], OOWS [53], and NDT [39] cover every type of requirements relevant to the classification introduced in Section 2. However, these approaches use their own terminology to name each type of requirement, with the exception of A-OOH which applies the classification directly.

Although the approaches presented in Table 11 share, in a few cases, a term with the same name as regards the taxonomy, some types of requirements are used to consider extra functionality, i.e., NDT includes Navigational and Layout requirements in the concept of Interaction requirements and OOWS uses Content and Service requirements in the terms used for Functional requirements [58, 59, 60].

Finally, Non-Functional requirements are considered in a very general manner by almost all the approaches, as can be noted in Table 11. The only approach that considers Non-Functional requirements in a more detailed form is WSDM, which details the Security and Usability Non-Functional requirements to use. [51].

5.4 Tool support (RQ4)

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

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

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

With regard to the requirements phase, only NDT, UWE, OOWS and A-OOH have tool support. NDT does this by means of the NDT-Tool,[w] UWE provides a tool that consists of a MagidDraw[x] plugin, the so-called MagicUWE, OOWS combines OOWS-Suite with OlivaNova tool,[y] and finally, A-OOH has requirements phase support by means of an Eclipse plugin [69].

In terms of tool support for requirements validation, the three (NDT, OOWS, A-OOH) methods that have implemented traceability have tool support. A-OOH does by means of an Eclipse plugin with which the requiremens are specified and when the generation of the conceptual models is performed a weaving model is created to store the traces among requirements and the conceptual models. NDT does this by means of the NDT-Suite. The OOWS approach uses two tools, the first of which is the open source tool called AGG (Attributed

[w] http://www.iwt2.org
[x] http://www.magicdraw.com
[y] http://www.care-t.com/

Graph Grammar System) and the second of which is TaskTracer[z]developed by the authors to generate traceability reports. These reports are helpful to study aspects such as whether all the requirements are supported, the impact of changing a requirement, or how requirements are modelled [34]. With regard to the impact of changing a requirement, A-OOH supports impact analysis (verify the impact resulting from the change in any model) by means of the WebREd prototype tool. The WebREd tool generates a report with the potentially requirements affected because of a change.

5.5 Spreading of the methods for the development of Web applications

Figure 13 shows the number of publications by each Web engineering approach ordered by year with regard to requirements engineering and tool support. This figure shows a period from 1998 to 2009 based on the relevant documents for this systematic literature review, and also shows the conference, journal or workshop in which each study was published. OOWS has the greatest number of publications: three from 2002 to 2009, followed by two publications concerning UWE. According to this research, the most important conference in the Web Engineering field for the main topics in this systematic literature review is ICWE (*International Conference on Web Engineering*). The conferences, journals and workshops in which the publications were presented can be consulted in the appendix, which may serve as a future reference for the publication of research work in this area.

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

5.6 Suggestions for future research

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

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

- Virtually none of the Web engineering approaches do not support requirements validation: they simply place less importance on traceability, do not include it at all, or its

[z]http://eecs.oregonstate.edu/TaskTracer/

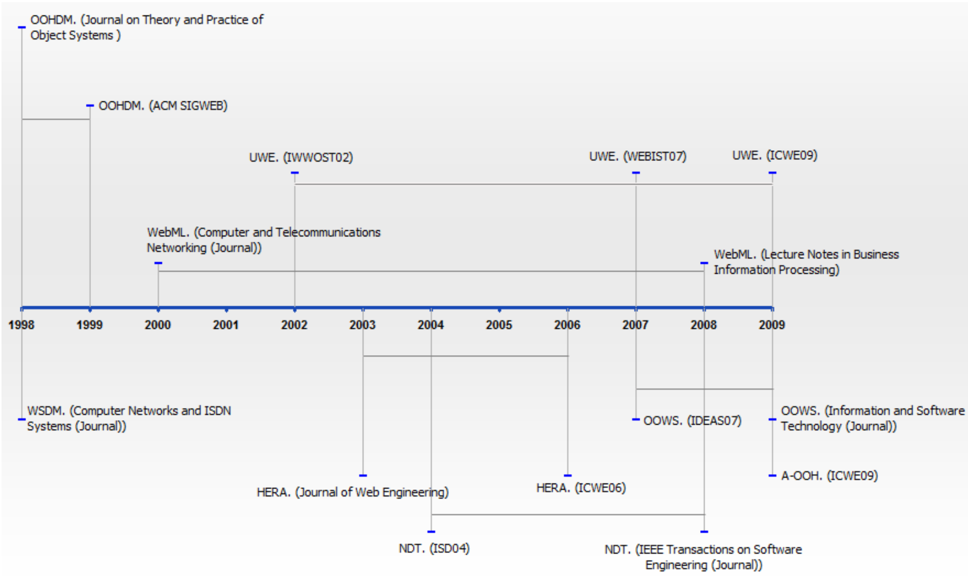

Table 13. Publications for each Web engineering approach ordered by year (refers only to RA phase).

corresponding techniques are poorly applied in the Web engineering field. This deciency must be solved, since the main purpose of traceability is to facilitate the understanding of the product under development and the ability to manage any changes that occur during the development process.

None of the approaches studied has a guided methodology for the analysis of Web engineering requirements that considers real user goals and needs from the beginning of the Web application development process. Empirical studies [18] demonstrate that efforts made to provide a detailed business modelling with which to capture the *stakeholders'* requirements in the system to be built considerably reduce drawbacks in later phases of the development process.

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

Moreover, although traceability is a very important success factor and quality criterion in software engineering, a common problem in Web development processes is the lack of requirements traceability support throughout different phases of the software development life cycle and its multiple levels of abstraction (down to the code level).

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

- Explicitly consider the specification of truly user goals and needs from very early stages of the Web application development process in a systematic manner.
- Consider the non-functional requirements from the requirements specification phase in

a structured and comprehensive manner.

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

6 Conclusion

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

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

In this context, most of the approaches explored in this systematic literature review do not consider the real user expectations of the website or *stakeholders* from the early requirements analysis phase. A-OOH is the exception, as it considers these expectations through the $i*$ Framework, in which requirements are modelled based on user goals and objectives, thus avoiding the specification of requirements in textual form, and based on Task Analysis, which implies certain limitations.

The development of Web applications can no longer be considered as common software systems owing to the diversity of users who have access to these applications. It is therefore necessary to provide solutions in the Web engineering field in order to specify and develop this type of applications by considering the huge heterogeneous user community, and their needs and goals. Our future work will therefore be a model-driven design approach guided by the requirements engineering process applied in the Web engineering domain. The proposal will be entirely based on the A-OOH method. Therefore, it will be helpful in enhancing communication among *stakeholders*, designers and costumers. To this aim, it is necessary to integrate techniques and tools for the automated generation of Web systems. In this respect, this proposal will therefore be supported by an open source tool for requirements specification. The Web designer will be able to use this tool to obtain the Web application conceptual models from requirements, along with support for requirements validation and management.

7 Acknowledgements

This work has been partially supported by the MANTRA project (GRE09-17 from the University of Alicante, and GV/2011/035 from the Valencia Government), and by the MESOLAP (TIN2010-14860) from the Spanish Ministry of Education and Science. José Alfonso Aguilar is funded by CONACYT (Consejo Nacional de Ciencia y Tecnología) Mexico and University of Sinaloa, Mexico.

References

1. Sommerville, I. *Software Engineering.* Addison-Wesley, 6th edition, (2001).
2. Ginige, A. and Murugesan, S. *Cutter IT Journal* **14**(7), 24–35 (2001).
3. Cachero, C. and Gómez, J. In *Interfaces in Navigation Design. 21th International Conference on Conceptual Modeling. El Escorial*, (2002).
4. Casteleyn, S., Garrigós, I., and Troyer, O. D. In *APWeb*, 453–463, (2005).
5. Casteleyn, S., Woensel, W. V., and Houben, G.-J. In *Hypertext*, 189–198, (2007).
6. Ceri, S. and Manolescu, I. In *VLDB*, 1151, (2003).
7. Koch, N. In *International Workshop on Web-oriented Software Technology (IWWOST)*, 40–41, (2002).
8. Escalona, M. J. and Koch, N. *J. Web Eng.* **2**(3), 193–212 (2004).
9. Kitchenham, B. Technical report, Keele University and NICTA, (2004).
10. Mendes, E. In *Empirical Software Engineering, 2005. 2005 International Symposium on*, 10. IEEE, (2005).
11. Staples, M. and Niazi, M. *The Journal of Systems and Software* **80**, 1425–1437 (2007).
12. Jorgensen, M. and Shepperd, M. *Software Engineering, IEEE Transactions on* **33**(1), 33 –53 (2007).
13. Kitchenham, B., Mendes, E., and Travassos, G. *Software Engineering, IEEE Transactions on* **33**(5), 316 –329 May (2007).
14. Nicolás, J. and Toval, A. *Information and Software Technology* **51**(9), 1291–1307 (2009).
15. Lisboa, L., Garcia, V., Lucrédio, D., de Almeida, E., de Lemos Meira, S., and de Mattos Fortes, R. *Information and Software Technology* **52**(1), 1–13 (2010).
16. Kitchenham, B., Pretorius, R., Budgen, D., Pearl Brereton, O., Turner, M., Niazi, M., and Linkman, S. *Inf. Softw. Technol.* **52**, 792–805 August (2010).
17. Aguilar, J. A., Garrigós, I., Mazón, J. N., and Trujillo, J. In *6th Web Information Systems and Technologies (WEBIST)*, volume 2, 187–190 (SciTePress Digital Library, Valencia, Spain, 2010).
18. Nuseibeh, B. and Easterbrook, S. M. In *ICSE - Future of SE Track*, 35–46, (2000).
19. Maiden, N. and Rugg, G. *Software Engineering Journal* **11**(3), 183 –192 (1996).
20. Shaw, M. and Gaines, B. *Software Engineering Journal* **11**(3), 149 –165 (1996).
21. Viller, S. and Sommerville, I. In *Requirements Engineering, 1999. Proceedings. IEEE International Symposium on*, 6 –13, (1999).
22. Yu, E. *Modelling Strategic Relationships for Process Reenginering.* PhD thesis, University of Toronto, Canada, (1995).
23. Sommerville, I. *Software engineering (5th ed.).* Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, (1995).
24. Bass, L., Merson, P., Clements, P., Bergey, J., Ozkaya, I., and Sangwan, R. Technical report, Software Engineering Institute, Carnegie Mellon University, (2006).
25. Gotel, O. and Finkelstein, A. In *Proceedings of the First International Conference on Requirements Engineering*, 94–101, (1994).
26. Estublier, J. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, 279–289 (ACM, New York, NY, USA, 2000).
27. Garrigós, I., Mazón, J.-N., and Trujillo, J. In *ICWE*, 151–165, (2009).
28. Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., and Linkman, S. *Information and Software Technology* **51**(1), 7 – 15 (2009).
29. Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., and Khalil, M. *J. Syst. Softw.* **80**, 571–583 April (2007).
30. Walia, G. S. and Carver, J. C. *Information and Software Technology* **51**(7), 1087 – 1109 (2009).
31. Beecham, S., Baddoo, N., Hall, T., Robinson, H., and Sharp, H. *Information and Software Technology* **50**(9-10), 860 – 878 (2008).
32. Escalona, M. J. and Koch, N. In *Web Information Systems and Technologies (WEBIST),* Aalst, W., Mylopoulos, J., Sadeh, N. M., Shaw, M. J., Szyperski, C., Filipe, J., Cordeiro, J., and Pedrosa, V., editors, volume 1 of *Lecture Notes in Business Information Processing*, 267–280. Springer

Berlin Heidelberg (2007).
33. Koch, N., Zhang, G., and Escalona, M. J. In *ICWE '06: Proceedings of the 6th international conference on Web engineering*, 281–288 (ACM, New York, NY, USA, 2006).
34. Valderas, P. and Pelechano, V. *Inf. Softw. Technol.* **51**(4), 749–768 (2009).
35. Schwabe, D. and Rossi, G. *Communications of the ACM* **38**(8), 45–46 (1995).
36. Molina, F. and Toval, A. *Adv. Eng. Softw.* **40**(12), 1306–1317 (2009).
37. Bolchini, D. and Mylopoulos, J. In *WISE '03: Proceedings of the Fourth International Conference on Web Information Systems Engineering*, 166 (IEEE Computer Society, Washington, DC, USA, 2003).
38. Koch, N. In *ICWE '06: Workshop proceedings of the sixth international conference on Web engineering*, 3 (ACM, New York, NY, USA, 2006).
39. Escalona, M. J. and Aragón, G. *IEEE Transactions on Software Engineering* **34**(3), 377–390 (2008).
40. Ceri, S., Fraternali, P., and Bongio, A. *The International Journal of Computer and Telecommunications Networking* **33**(1-6), 137–157 (2000).
41. De Troyer, O. M. F. and Leune, C. J. *Comput. Netw. ISDN Syst.* **30**(1-7), 85–94 (1998).
42. Houben, G., Barna, P., Frasincar, F., and Vdovjak, R. *Web Engineering* , 529–538 (2003).
43. Koch, N. and Kraus, A. In *IWWOST 02: Second Int. Worskhop on Web-oriented Software Technology*, (2002).
44. Koch, N. In *Workshop proceedings of the sixth international conference on Web engineering*, 3. ACM, (2006).
45. Koch, N., Knapp, A., Zhang, G., and Baumeister, H. In *Web Engineering: Modelling and Implementing Web Applications*, Human-Computer Interaction Series, 157–191. Springer London (2008).
46. Busch, M. and Koch, N. In *ICWE '9: Proceedings of the 9th International Conference on Web Engineering*, 505–508 (Springer-Verlag, Berlin, Heidelberg, 2009).
47. Kroiss, C., Koch, N., and Knapp, A. In *Web Engineering,* Gaedke, M., Grossniklaus, M., and Daz, O., editors, volume 5648 of *Lecture Notes in Computer Science*, 493–496. Springer Berlin / Heidelberg (2009).
48. Escalona, M., Mejías, M., and Torres, J. In *13th International conference on information systems development: methods and tools, theory and practice, Vilna, Lithuania*, 149–59, (2004).
49. Bongio, A., Milano, P. D., Fraternali, P., Maurino, A., and Ceri, S. In *The World Wide Web and Databases (WebDB, Selected Papers*, 201–214, (2000).
50. Acerbis, R., Bongio, A., Brambilla, M., Butti, S., Ceri, S., and Fraternali, P. In *Objects, Components, Models and Patterns,* Aalst, W., Mylopoulos, J., Sadeh, N. M., Shaw, M. J., Szyperski, C., Paige, R. F., and Meyer, B., editors, volume 11 of *Lecture Notes in Business Information Processing*, 392–411. Springer Berlin Heidelberg (2008).
51. Troyer, O. D. and Casteleyn, S. In *IWWOST01*, 30–98, (2001).
52. Wilder, K. V. *Implementation Generation for WSDM using Web Applications Framework.* PhD thesis, University of Brussel, (2009).
53. Fons, J., Valderas, P., Ruiz, M., Rojas, G., and Pastor, O. In *International Workshop on Web Oriented Software Technology (IWWOST). (2003) 6570*, 65–70, (2003).
54. Pastor, O., Abrah ao, S. M., and Fons, J. In *EC-Web 2001: Proceedings of the Second International Conference on Electronic Commerce and Web Technologies*, 16–28 (Springer-Verlag, London, UK, 2001).
55. Pastor, O., Pelechano, V., Insfrn, E., and Gmez, J. In *Conceptual Modeling ER 98,* Ling, T. W., Ram, S., and Lee, M. L., editors, volume 1507 of *Lecture Notes in Computer Science*, 183–196. Springer Berlin / Heidelberg (2004).
56. Quintero, R., Pelechano, V., Pastor, O., and Fons, J. *Jornadas de Ingeniería de Software y Base de Datos (JISBD), VIII* , 84–668 (2003).
57. Cors, J. F. *OOWS: Un Metode Dirigit per Models per al Desenvolupament d Aplicacions Web.* Tesis doctoral en informtica, Departamento de Sistemas Informáticos y Computación, Universidad

Politécnica de Valencia, (2008).
58. Fons, J., Garca, F., Pelechano, V., and Pastor, O. In *Web Engineering,* Lovelle, J., Rodrguez, B., Gayo, J., del Puerto Paule Ruiz, M., and Aguilar, L., editors, volume 2722 of *Lecture Notes in Computer Science*, 457–461. Springer Berlin / Heidelberg (2003).
59. Abraho, S., Fons, J., Gonzlez, M., and Pastor, O. In *Adaptive Hypermedia and Adaptive Web-Based Systems,* De Bra, P., Brusilovsky, P., and Conejo, R., editors, volume 2347 of *Lecture Notes in Computer Science*, 358–362. Springer Berlin / Heidelberg (2006).
60. Durán, G. E. R. *Modelling Adaptative Web Applications in OOWS.* PhD thesis, Technical University of Valencia, Spain, (2008).
61. Valverde, F., Valderas, P., and Fons, J. *IDEAS, Venezuela* (2007).
62. Garrigós, I., Gomez, J., and Houben, G.-J. *Inf. Softw. Technol.* **52**(9), 991–1010 (2010).
63. Gómez, J., Cachero, C., and Pastor, O. In *CAiSE '00: Proceedings of the 12th International Conference on Advanced Information Systems Engineering*, 79–93 (Springer-Verlag, London, UK, 2000).
64. Garrigós, I. *A-OOH: Extending Web Application Design with Dynamic Personalization.* PhD thesis, University of Alicante, Spain, (2008).
65. Aguilar, J. A., Garrigós, I., Mazón, J. N., and Trujillo, J. *Journal of Universal Computer Science JUCS* **16**(17), 2475–2494 (2010).
66. Aguilar, J. A., Garrigós, I., and Mazón, J.-N. In *DSDM: Actas del VII Taller sobre Desarrollo de Software Dirigido por Modelos, JISBD, Congreso Espanol de Informatica (CEDI)*, 146–155 (SISTEDES, Valencia, Espana, 2010).
67. Barbero, M., Del Fabro, M., and Bézivin, J. In *3rd ECMDA-Traceability Workshop.* Citeseer, (2007).
68. Aguilar, J., Garrigós, I., and Mazón, J. *Computational Science and Its Applications-ICCSA 2011* , 421–436 (2011).
69. Eclipse. (2010).
70. Garzotto, F., Paolini, P., and Schwabe, D. *ACM Transactions on Information Systems (TOIS)* **11**(1), 1–26 (1993).
71. Garzotto, F., Paolini, P., and Schwabe, D. *ACM Trans. Inf. Syst.* **11**, 1–26 January (1993).
72. Schwabe, D., de Almeida Pontes, R., and Moura, I. *ACM SIGWEB Newsletter* **8**(2), 18–34 (1999).
73. Vdovjak, R., Frasincar, F., Houben, G., and Barna, P. *Journal of Web Engineering* **2**, 3–26 (2003).
74. Casteleyn, S., Van Woensel, W., and Houben, G.-J. In *HT '07: Proceedings of the eighteenth conference on Hypertext and hypermedia*, 189–198 (ACM, New York, NY, USA, 2007).
75. Barna, P., Frasincar, F., and Houben, G.-J. In *ICWE 06: Proceedings of the 6th international conference on Web engineering*, 321–328 (ACM, New York, NY, USA, 2006).
76. Ramesh, B. and Jarke, M. *IEEE Transactions on Software Engineering* **27**, 58–93 (2001).
77. Baumeister, H., Koch, N., and Zhang, G. In *International Conference on Web Engineering (ICWE 2005*, 406–416. Springer Verlag, (2005).

Appendix A

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

Conferences

- International Conference on Information Systems Development (ISD).
- International Conference on Web Engineering (ICWE).

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

Workshops

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

Journals

- Computer Networks: The International Journal of Computer and Telecommunications Networking[a]
- IEEE Transactions On Software Engineering.[b]
- Information and Software Technology[c]
- Journal of Web Engineering.[d]
- ACM Special Interest Group on Hypertext, Hypermedia and the Web[e] (SIGWEB) (Newsletter).
- Communications of the ACM.[f]

[a] http://www.elsevier.com
[b] http://www.computer.org/portal/web/tse/
[c] http://www.elsevier.com
[d] http://www.rintonpress.com/journals/jwe/
[e] http://www.sigweb.org
[f] http://portal.acm.org