

Requirements in Web engineering: a systematic literature review

José Alfonso Aguilar^a, Irene Garrigós^a, Jose-Norberto Mazón^a

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

Abstract

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

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

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

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

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

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

1. Introduction

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

engineering methodologies [13, 14, 16, 18, 43] have therefore provided different mechanisms with which to consider these new requirements while developing Web applications [26].

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

*ja.aguilar@dlsi.ua.es

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

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

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

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

2. Requirements and Web engineering concepts

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

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

rules, and the goals, tasks and responsibilities of its members and the data that is needed. This stage allows designers to detect and resolve conflicts that may occur among different *stakeholders* [63].

- **Requirements specification.** This step is an integral description of the behavior of the system to be developed. This description aims to facilitate an effective communication of requirements among different *stakeholders*. The most widely used techniques for requirements specification are, among others, templates, scenarios, use case modelling, and natural language [8].
- **Requirements validation.** This step aims to establish whether the requirements and models elicited provide an accurate representation of the actual *stakeholder* requirements. Some techniques for requirements validation are reviews, audit and traceability matrices. A requirement must be traceable, since it is defined throughout the development process. A traceable requirement ensures that change management is able to assess the impact of a change on the rest of the application. It is therefore necessary for *stakeholders* to be able to see that the final *work product*² is useful for achieving their requirements, and that each requirement has been derived within a particular work product. To do this, requirements engineering needs to ensure traceability. Requirements traceability refers to the ability to describe and follow the life of a requirement, in both a forward and a backward direction [37]. While forward traceability is related to following the requirement to its final implementation, backward traceability refers to following the *work product* to its source requirement (i.e., the associated requirement that originated it). Interestingly, traceability between requirements and *work products* is a norm recommended by CMMI (Capability Maturity Model Integration)³, specifically in CMMI's Level 2 (within the requirements management process area).
- **Requirements management.** This step consists of recognizing changes through continual requirements elicitation, the re-evaluation of risks, and

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

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

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

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

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

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

browsing experience”, “attract more users”, “improve efficiency”, etc.

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

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

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

3. A systematic literature review for requirements in Web engineering

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

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

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

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

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

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

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

3.1. Research questions

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

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

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

3.2. Search strategy

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

- **eJournals (repositories with restricted access):** ACM⁴, IEEE Computer Society Digital Library⁵, Science Direct⁶.

⁴<http://portal.acm.org>

⁵<http://ieeexplore.ieee.org>

⁶<http://www.sciencedirect.com>

- **Digital library of scientific literature:** DBLP Computer Science Bibliography⁷.
- **Research resources in the World Wide Web:** Google Scholar⁸.

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

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

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

- **ACM.** (*Web Engineering*) OR ((*Web requirements*) OR (*requirements Engineering*)) AND (*Web engineering methods*) AND (*techniques*) OR (*requirements model*) AND (*requirements traceability*) AND (*tool support*)
- **IEEE.** (*Web engineering*) OR ((*Web requirements*) OR (*Web system*) OR (*requirements engineering*)) AND ((*Web engineering methods*) OR (*techniques*)) AND ((*Web site design*) OR (*requirements model*)) OR (*requirements modeling*) AND (*requirements traceability*) AND (*tool support*)
- **Science Direct.** *Web engineering* OR *Web requirements* OR *requirements engineering* AND *Web engineering methods* AND *techniques* AND *Web site design* OR *requirements model* OR *requirements modeling* AND *requirements traceability* AND *tool support*
- **DBLP.** *Requirements* AND *Web*
- **World Wide Web (Google scholar).** *Web +engineering +requirements +phase -stage +specification —requirements analysis +techniques —methods —approaches +tool support +model +traceability support +applications*

3.3. Study selection

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

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

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

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

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

⁷<http://www.informatik.uni-trier.de/~ley/db/index.html>

⁸<http://scholar.google.com>

The search for primary studies was conducted by using the search strategy defined in Section 3.2. In this phase, the *Title*, *Author*, and *Year* of the paper were used to keep track of the information extraction after the first exclusion. Table 1 shows the results of 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>i*</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. Gutierrez	2009
On the generation of requirements specifications from software engineering models: A systematic literature review	Joaquin 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 Vanderdonck	2008
Model Transformations from Requirements to Web System Design	Nora Koch, Gefei Zhang, Maria 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 Through Animation	Joumana Dargham, Rima Se-maan	2008
From Task-Oriented to Goal-Oriented Web Requirements Analysis.	Davide Bolchini , John Mylopoulos	2003
NDT. A Model-Driven Approach for Web Requirements	María José Escalona, Gustavo Aragn	2008
Impact Analysis from Multiple Perspectives: Evaluation of Traceability Techniques	Salma Imtiaz, Naveed Ikram, Saima Imtiaz	2008

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

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

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

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

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

3.4. Data extraction

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

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-Francois Roy	2007
Requirements Traceability and Transformation Conformance in Model-Driven Development	Joao Paulo Almeida, Pascal van Eck, Maria-Eugenia Jacob	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

comings with regard to the requirements engineering techniques used, and finally, tool support.

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

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

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

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

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

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

4. Analysis of the primary studies

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

of requirements used by each approach are also discussed (according to the taxonomy described in Section 2).

4.1. UML-based Web Engineering (UWE)

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

Development process: UWE is based on Model-Driven Architecture (MDA) [45]. Within MDA, requirements are considered in the CIM (Computational Independent Model) in order to show what the Web application is expected to do without showing details of how it is implemented. Through this requirements model, UWE can generate content, navigation, presentation and process models. Generation is done by clearly and formally establishing a set of Query/View/Transformation (QVT) transformations⁹ to be automatically performed [46]. The authors claim that the main benefit of MDA is that less time and effort are needed to develop the whole Web application, thus improving productivity.

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

Requirements traceability: This methodological approach does not support traceability.

Tool support: A plugin called MagicUWE was developed to be used with the CASE tool MagicDraw¹⁰ [12], which allows UWE to provide a CASE (Computer-Aided Software Engineering) tool. Unfortunately, requirements analysis is missing in this tool. It

is worth mentioning that in [49] the author proposes an approach called UWE4JSF for the automatic generation of Web applications in JSF (Java Server Faces) derived from UWE models.

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

4.2. Navigational Development Techniques (NDT)

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

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

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

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

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

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

⁹<http://www.omg.org/spec/QVT/1.0>

¹⁰<http://www.magicdraw.com>

a set of tools composed of NDT-Profile, NDT-Driver, NDT Quality and NDT Report¹¹.

- NDT-Profile. This is a profile defined in Enterprise Architect¹².
- NDT-Driver. This uses NDT-Profile to allow model-driven transformations to be executed in order to obtain analysis artefacts.
- NDT Quality. This tool verifies whether NDT constraints and relations have been followed correctly. Traceability errors between phases are also numbered and grouped by phases.
- NDT Report. This tool obtains the requirements, the analysis, the design and the test documents.

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

4.3. Web Modeling Language (WebML)

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

Development process: WebML consists of different phases that must be applied in an iterative and incremental manner. The process has several cycles, each of which produces a prototype or a partial version of the application which permits conducting, testing and evaluation to take place during the early development phases¹³. The development process of this Web engineering approach is divided into six phases:

- Requirements analysis to collect requirements.
- Conceptual modelling to represent the Web system independently of any technological detail.

- Implementation to derive conceptual models in a specific technology.
- Testing and evaluation of the Web application.
- Deployment of the Web application in a specific architecture.
- Maintenance and evolution.

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

Requirements traceability: This approach does not support requirements traceability.

Tool support: WebML has tool support: WebRatio¹⁴ [2]. The support that WebRatio offers to WebML is mainly focused on providing facilities for a conceptual modeling phase and for the automatic generation of J2EE code (as opposed to traditional prototyping tools, which generate application mockups). In this context, WebML models are implicitly defined within WebRatio with a document type definition (DTD) and with a UML 2.0 profile. WebML models can therefore be used in conjunction with other notations and can ensure interoperability with other tools.

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

4.4. Web Site Design Method (WSDM)

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

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

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

¹¹<http://www.iwt2.org>

¹²<http://www.sparxsystems.com/>

¹³<http://www.webml.org>

¹⁴<http://www.webratio.com>

- Conceptual design, in which a navigational model and a class diagram are designed for the navigational paths.
- Implementation design, whose aim is to create a consistent, pleasing and efficient look and feel for the conceptual design made in the previous phase.
- Implementation of the Web application tailored to a chosen implementation environment.

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

Requirements traceability: This is not supported by WSDM.

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

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

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

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

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

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

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

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

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

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

Tool support: This approach has a development framework called OOWS-Suite [67], which is integrated with the OlivaNova tool¹⁵ to provide support for requirements engineering.

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

¹⁵<http://www.care-t.com/>

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

4.6. Adaptive Object Oriented Hypermedia (A-OOH)

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

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

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

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

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

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

Tool support: The development process is supported by the Eclipse¹⁶ development platform [23].

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

4.7. Object-Oriented Hypermedia Design Method (OOHDM)

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

Development process: The OOHDM development method considers five phases:

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

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

¹⁶<http://www.eclipse.org/>

of requirements: conceptual, navigational and layout requirements.

Requirements traceability: OOHDM does not support requirements traceability.

Tool support: This approach has an environment called OOHDM-Web. OOHDM-Web allows the implementation of hypermedia applications such as CGI scripts which produce dynamically generated pages, whose contents are fed from a database and integrated with pre-defined templates [60]. This tool can be downloaded from the OOHDM wiki¹⁷.

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

4.8. *Hera*

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

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

- The integration and data retrieval phase, which generates a conceptual model from a user query. This model contains the data for which the application will generate a presentation (here this is considered to be the navigation).
- The presentation generation phase considers how to select and obtain the data from the storage part of the application. The handling of the interaction from users is also specified (querying, navigation, or application-specific user interaction).

¹⁷<http://www.tecweb.inf.puc-rio.br>

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

Requirements Traceability: This approach does not support requirements traceability.

Tool support: In the Presentation generation phase, the Hera method uses Saxon 7.0¹⁸ (XPath 2.0, XSLT 2.0). For the purpose of data retrieval, Hera uses the RDF query language RQL¹⁹ and its Java-based interpreter called Sesame²⁰.

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

5. Discussion

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

5.1. *Requirements terminology*

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

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

¹⁸<http://saxon.sourceforge.net>

¹⁹<http://www.2002.org/CDROM/refereed/329>

²⁰<http://www.openrdf.org>

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

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

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

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

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

5.2. Requirements analysis

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

to cover the entire Web application development process.

5.3. Techniques applied

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

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

The ***i** framework** is a goal-oriented technique that is applied to requirements analysis in A-OOH in order to describe and evaluate design alternatives and their relationships to the organizational objectives. This framework is used in order to explicitly analyse and model

relationships among multiple *stakeholders* (actors in the *i** notation). The *i** Framework has proved to be useful for representing: (i) the *stakeholders*' intentions, i.e. their motivations and goals, (ii) dependencies between *stakeholders* to achieve their goals, and (ii) the (positive or negative) effects of these goals on each other in order to be able to select alternative designs for the system, thus maximizing goal fulfilment.

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

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

Table 10: RE techniques by each Web engineering approach.

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

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

The requirements specification techniques used by WSDM are **Conceptual Maps of Roles and Activities** and **Data Dictionary**. These techniques are difficult to maintain and analyse owing to the fact that the requirements are basically defined in textual form. In this respect, describing navigational requirements by means

of textual descriptions is not an easy task owing to the description of alternative navigation paths through the Web application.

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

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

5.4. Requirements traceability

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

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

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

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

5.5. Tool support

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

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

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

With regard to the requirements analysis phase, only NDT, UWE, OOWS and A-OOH have tool support. NDT does this by means of the NDT-Tool²¹, UWE provides a tool that consists of a MagicDraw²² plugin, the so-called MagicUWE, OOWS combines OOWS-Suite

with OlivaNova tool²³, and finally, A-OOH has requirements analysis support by means of an Eclipse plugin [23].

In terms of tool support for requirements traceability, the two approaches that have implemented traceability have tool support. NDT does this by means of the NDT-Suite. The OOWS approach uses two tools, the first of which is the open source tool called AGG (Attributed Graph Grammar System) and the second of which is TaskTracer²⁴ developed by the authors to generate traceability reports. These reports are helpful to study aspects such as whether all the requirements are supported, the impact of changing a requirement, or how requirements are modelled [66].

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

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

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

²¹<http://www.iwt2.org>

²²<http://www.magicdraw.com>

²³<http://www.care-t.com/>

²⁴<http://eecs.oregonstate.edu/TaskTracer/>

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

5.7. Suggestions for future research

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

- Those Web engineering approaches that have a requirements analysis phase do not provide a systematic method with which to design models from requirements since they consider the requirements analysis phase without real user goals and needs. This gap must be covered, because the successful development of a system depends on satisfying *stakeholders'* needs.
- Virtually none of the Web engineering approaches do not support requirements traceability: they simply place less importance on traceability, do not include it at all, or its corresponding techniques are poorly applied in the Web engineering field. This deficiency must be solved, since the main purpose of traceability is to facilitate the understanding of the product under development and the ability to manage any changes that occur during the development process.

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

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

Moreover, although traceability is a very important success factor and quality criterion in software engi-

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

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

- Explicitly consider the specification of real user goals and needs from very early stages of the Web application development process in a systematic manner.
- Consider the non-functional requirements from the requirements specification phase in a structured and comprehensive manner.
- Full support for requirements traceability (bidirectional) from requirements specification throughout the different phases of the Web application development process.

6. Conclusion

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

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

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

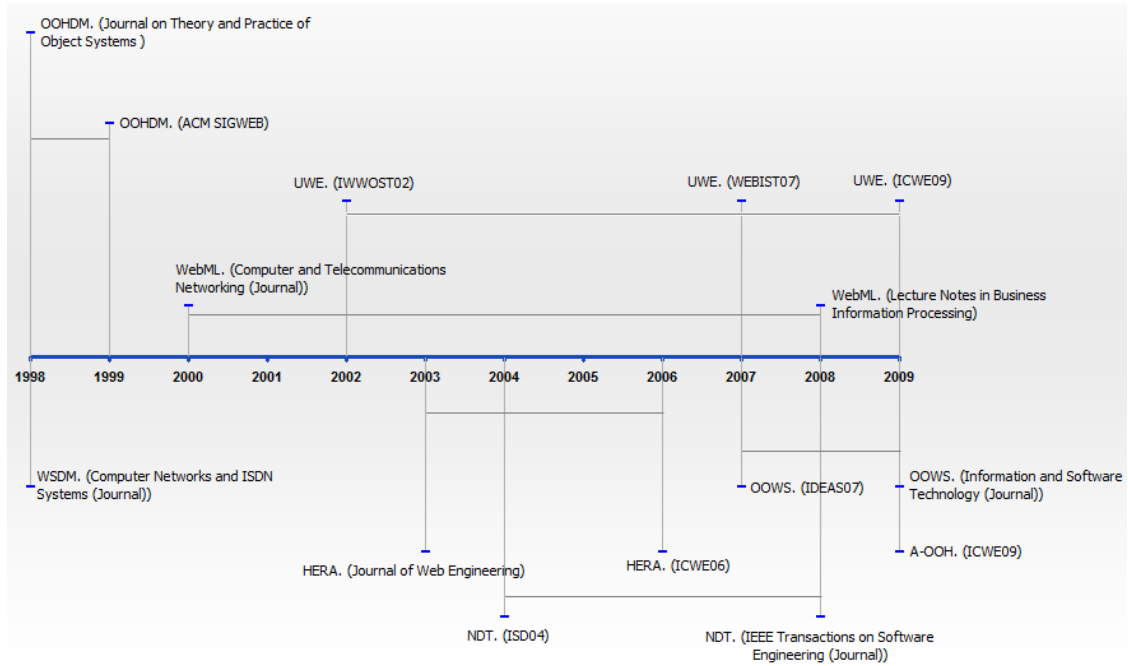


Figure 1: Publications for each Web engineering approach ordered by year (refers only to RA phase).

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

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

automated generation of Web systems. This proposal will therefore be supported by an open source tool for requirements specification. The Web designer will be able to use this tool to obtain the Web application conceptual models from requirements, along with support for bidirectional traceability between requirements and conceptual models.

7. Acknowledgements

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

Appendix A. Journals and conferences

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

Conferences.

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

Workshops.

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

Journals.

- Computer Networks: The International Journal of Computer and Telecommunications Networking²⁵
- IEEE Transactions On Software Engineering²⁶.
- Information and Software Technology²⁷
- Journal of Web Engineering²⁸.
- ACM Special Interest Group on Hypertext, Hypermedia and the Web²⁹ (SIGWEB) (Newsletter).
- Communications of the ACM³⁰.

References

²⁵<http://www.elsevier.com>

²⁶<http://www.computer.org/portal/web/tse/>

²⁷<http://www.elsevier.com>

²⁸<http://www.rintonpress.com/journals/jwe/>

²⁹<http://www.sigweb.org>

³⁰<http://portal.acm.org>

References

- [1] Abraho, S., Fons, J., Gonzlez, M., Pastor, O., 2006. Conceptual modeling of personalized web applications. In: De Bra, P., Brusilovsky, P., Conejo, R. (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems*. Vol. 2347 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, pp. 358–362.
- [2] Acerbis, R., Bongio, A., Brambilla, M., Butti, S., Ceri, S., Fraternali, P., 2008. Web applications design and development with webml and webratio 5.0. In: Aalst, W., Mylopoulos, J., Sadeh, N. M., Shaw, M. J., Szyperski, C., Paige, R. F., Meyer, B. (Eds.), *Objects, Components, Models and Patterns*. Vol. 11 of *Lecture Notes in Business Information Processing*. Springer Berlin Heidelberg, pp. 392–411.
- [3] Aguilar, J. A., Garrigós, I., Mazón, J.-N., 2010. Modelos de weaving para trazabilidad de requisitos web en a-ooh. In: DSDM: Actas del VII Taller sobre Desarrollo de Software Dirigido por Modelos, JISBD, Congreso Espanol de Informatica (CEDI). SISTEDES, Valencia, Espana, pp. 146–155.
- [4] Aguilar, J. A., Garrigós, I., Mazón, J. N., Trujillo, J., 2010. An mda approach for goal-oriented requirement analysis in web engineering. *Journal of Universal Computer Science* JUCS 16 (17), 2475–2494, http://www.jucs.org/jucs_16_17/an_mda_approach_for.
- [5] Aguilar, J. A., Garrigós, I., Mazón, J. N., Trujillo, J., 2010. Web Engineering approaches for requirement analysis- A Systematic Literature Review. In: 6th Web Information Systems and Technologies (WEBIST). Vol. 2. SciTePress Digital Library, Valencia, Spain, pp. 187–190.
- [6] Barbero, M., Del Fabro, M., Bézivin, J., 2007. Traceability and provenance issues in global model management. In: 3rd ECMDA-Traceability Workshop. Citeseer.
- [7] Barna, P., Frasinca, F., Houben, G.-J., 2006. A workflow-driven design of web information systems. In: ICWE 06: Proceedings of the 6th international conference on Web engineering. ACM, New York, NY, USA, pp. 321–328.
- [8] Bass, L., Merson, P., Clements, P., Bergey, J., Ozkaya, I., Sangwan, R., 2006. A comparison of requirements specification methods from a software architecture perspective. Tech. rep., Software Engineering Institute, Carnegie Mellon University.
- [9] Baumeister, H., Koch, N., Zhang, G., 2005. Modelling adaptivity with aspects. In: International Conference on Web Engineering (ICWE 2005). Springer Verlag, pp. 406–416.
- [10] Bolchini, D., Mylopoulos, J., 2003. From task-oriented to goal-oriented web requirements analysis. In: WISE '03: Proceedings of the Fourth International Conference on Web Information Systems Engineering. IEEE Computer Society, Washington, DC, USA, p. 166.
- [11] Bongio, A., Milano, P. D., Fraternali, P., Maurino, A., Ceri, S., 2000. Modeling data entry and operations in webml. In: *The World Wide Web and Databases (WebDB, Selected Papers*. pp. 201–214.
- [12] Busch, M., Koch, N., 2009. Magicuwe: A case tool plugin for modeling web applications. In: ICWE '9: Proceedings of the 9th International Conference on Web Engineering. Springer-Verlag, Berlin, Heidelberg, pp. 505–508.
- [13] Cachero, C., Gómez, J., 2002. Advanced conceptual modeling of web applications: Embedding operation. In: *Interfaces in Navigation Design*. 21th International Conference on Conceptual Modeling. El Escorial.
- [14] Casteleyn, S., Garrigós, I., Troyer, O. D., 2005. Automatic runtime validation and correction of the navigational design of web sites. In: APWeb, pp. 453–463.
- [15] Casteleyn, S., Van Woensel, W., Houben, G.-J., 2007. A semantics-based aspect-oriented approach to adaptation in web

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

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