Highways of Requirements Engineering

Sanjeev Narayan Bal
Asst. Prof. Dept. of Comp.Sc.
Trident Academy of Creative Technology, Bhubaneswar

Abstract

This paper describes the activities and work products that contribute to the specification and validation of the software requirements of different applications. The activities of elicitation, analysis, specification, validation, and requirements management are discussed in each of those areas are highlighted. This paper presents a detailed requirements engineering process for the development of semantic applications, describes the requirements engineering for developing an enterprise applications. The development activities include software implementation, maintenance, and enhancement and support for online transaction processing and overnight batch processing and also the requirements engineering process for the development of Web applications . In addition, Web systems have additional requirements for the navigational and multimedia aspects as well as for the usability .

Keywords

Enterprise applications development, enterprise information systems, business process, requirement engineering-requirement standards, software development activities, software requirement reviews, Web methodology.

1. Introduction

The requirements process can be described by the five primary disciplines of requirements elicitation, analysis, specification, validation, and management. Although there is overlap between these activities they are described here as individual phases for pedagogical purposes.

•Elicitation is concerned with proactively working with stakeholders to discover their needs, identify and negotiate potential conflicts, and establish a clear scope and boundaries for the project.

•Analysis involves gaining a deeper understanding of the product and its interactions; identifying requirements with global impact in allocating requirements to architectural components; and finally identifying additional conflicts that emerge through considering architectural implementations and negotiating agreements between stakeholders.

• Specification involves the production of a series of documents that capture the system and software requirements in order to support their systematic review, evaluation, and approval.

•Management of requirements is an ongoing activity that starts from the moment the first requirement is elicited and ends only when the system is finally decommissioned. Requirements management includes software configuration management, traceability, impact analysis, and version control.

• Validation occurs throughout the other four activities. It involves ensuring that the product meets stakeholders' requirements through activities such as formal and informal reviews and for more complex or critical systems through the use of formal verification techniques.

Activities across these disciplines are tightly coupled and therefore there is significant overlap and iteration between them. Figure 1 illustrates the iteration that occurs between various requirements activities and depicts the ongoing progress towards a validated requirements specification.

2. Requirements Engineering for Semantic Applications

A characterisation of semantic applications can be extracted taking into account the characteristics of this kind of applications presented in [1-4].

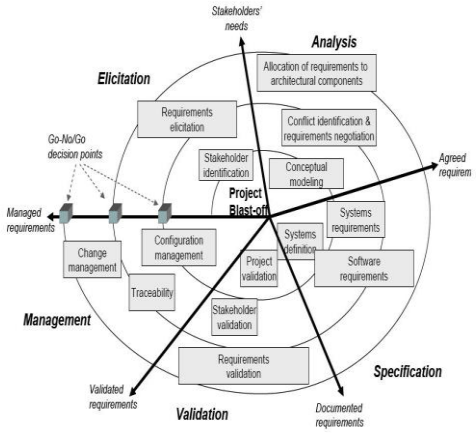

Figure 1. The requirements process

The tasks for carrying out the requirements elicitation and analysis activity can be seen in the activity diagram shown Figure 2 and are explained below.

Task 1. To identify the use cases. The objective of this task is to gather information about the application from the business requirements facilitated by the customer and to distil use cases from this information. Scenario-based elicitation and, in concrete, use cases are an appropriate approach to Requirements Engineering when implementing an agile method. However use cases are not as effective for eliciting constraints or high-level and nonfunctional requirements, as for example those related to the characteristics presented previously.

In order to speed-up this task, we provide a catalogue of use cases that commonly appear in semantic applications. When performing this task, these common use cases can be selected, adapted and appended to the identified set of requirements. The use cases have been obtained by analysing the scenarios presented previously from the viewpoint of the system user goals. The use cases identified are the following: (1) Query Information, where the user goal is to obtain integrated information from several resources given a query; (2) Search resources, where the goal is to find resources related to a given search; (3) Browse information, where the user wants to navigate through categorised information; (4) Analyse Information, where the goal is to obtain some analysis from collected information; (5) Extract Information, where the goal is to extract meaningful information by processing a search result; and (6) Manage Knowledge where the aim is to collaboratively construct and evolve shared knowledge.

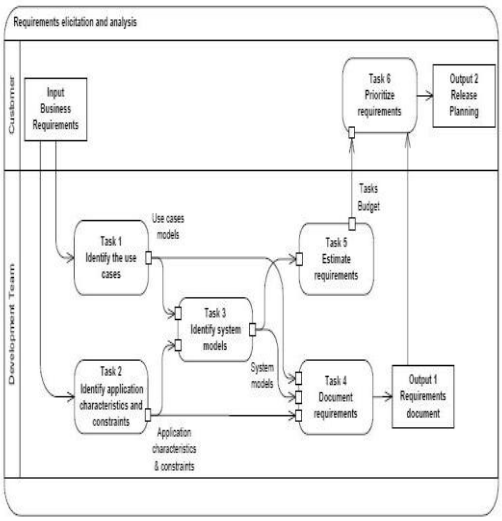

Figure 2.The requirements elicitation and analysis activity

Task 2. To identify application characteristics and constraints. The objective of this task is to collect non-functional requirements, that is, constraints on the services or functions offered by the system.

In order to speed-up this task, we provide a set of characteristics that commonly appear in semantic applications. When performing this task these common characteristics can be selected and appended to the identified set of requirements. The common characteristics have been obtained through an analysis of the State of the Art, are summarised previously, and have been classified according to four dimensions regarding the nature of the ontologies and data that the application will use, the kind of reasoning that the application will apply and the interoperability of the application with other systems.

Task 3. To identify system models. The objective of this task is to preliminary specify the system in form of system models, which are an important bridge between the requirements engineering and the design processes because they are often more understandable than detailed natural language descriptions of the system requirements [5]. The system models will reflect the scenarios identified during the use case identification task, constrained by the application characteristics. The system models allow representing the system from the following different perspectives: (1) an external perspective, where the context or environment of the application is modelled by showing the limits of the system where the application to develop will be deployed as well as the external systems; and (2) a structural perspective, where the structure of the data processed by the system is modelled. According to the process presented in this paper, the scenarios described previously can be viewed as system models.

In order to speed-up this task, we provide a catalogue of system model templates. When performing this task, these system model templates can be selected according to the identified use cases and the application characteristics and constraints. As an example, Figure 3 shows a the template used for an application that queries information to a group of data sources that are aligned with a shared vocabulary.

Task 4. To document requirements. In this task, the requirements discovered in the previous tasks are consolidated in a single description as the official statement of what the application developers should implement. The result of this task is the requirements document.

Task 5. To estimate requirements. In this task, the effort needed for implementing a set of requirements is estimated. As happens in Extreme Programming [30], the developers work together to break the system models down into development tasks. The result of this task reflects the estimation cost for each identified task.

Task 6. To prioritise requirements. In this task, the customer prioritises the development tasks

estimated in Task 5. The output of this task is the release planning that reflects the system models to be implemented for the next application release.

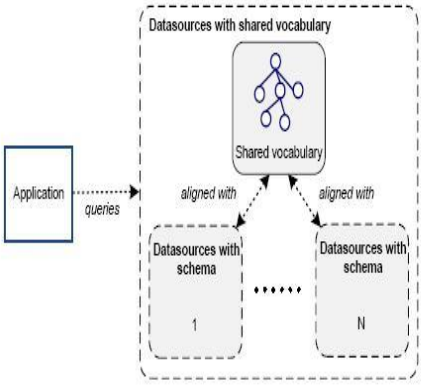

Figure 3. Example of a system model template

3. Requirements Engineering for Enterprise Applications

(i)To Understand The Problem Domain

The enterprise application problem has many symptoms: systems that don't work together, too much data and not enough information; incompatible and incorrect data: and excessive maintenance costs. There are lots of reasons for this, but the primary cause is the way the applications were developed:

• Independently as stand-alone, "stove-pipe" systems — not designed to be interoperable.

• Not based upon a single enterprise standard

• Without a common data architecture to ensure data sharing

• In unstructured programming languages

• Poorly documented if documented at all

• Designed to reflect developers interpretation of business requirements ten years, twenty years, or even longer ago

All of these approaches have value and some will even provide at least temporary benefit. However, unless they are business-driven and model-based they are more likely to further compound the problems than provide a solution.

Misunderstanding and lack of understanding of the domain or the real problem is a major challenge of this case. Even the users who are experts in the domain may not possess the kind of knowledge that can be easily communicated to others, and they may define their problem too broadly or too narrowly.

(ii)To Identify the Business Process

'"Business process" is a potentially wide-ranging term. For the purposes of this paper, a business process is a set of actions facilitating the transaction of business with an external or other internal entity. Although using information systems to automate business processes is much more efficient and cost-effective than employing human auditors, it brings consequences that have yet to be fully appreciated. A key component in the implementation and management of an efficient environment is the ability to automate standard processes. The IT Personnel will have a tough challenge in providing the power to streamline and automate processes that are within and reach beyond the walls of educational environment using a business process automation engine. Thus to identify the business process for this application development become more challenging.

Business Processes for enterprise application is identified as sequences of linked functional-level activities, which take inputs and produce outputs. The IT Personnel must emphasize the fact that the description of a business process specifically does not address the way the process transforms its inputs into its outputs – it describes what the process does, not how it does it.

(iii)To Emerge the Standards

Emerging the standard is also among big challenges where only the best standard can help the requirements engineering activity and workflow. For this study, IEEE Software Engineering Standard has been reviewed. Adoption of the IEEE software engineering standards is a more efficient and preferable approach. It also represents the most comprehensive and mature set of standards available. However, they may be complex and vexing at times and this is where implementing the IEEE Software Engineering Standards can help. It produces necessary information of the requirements for software systems as specified by either potential customers/users or designers producers and constituting the substance of an agreement between them. The Software Requirements Specification (SRS) focuses on the collection and organization of all requirements surrounding the project. It provides a complete vision of the As-Is process model analysis and the To-Be process model analysis. As-Is is the requirement study and analysis of the current business processes. To-Be shall be the proposed solution. The To-Be process model analysis will have input from the As-Is analysis and will go through some form of business process. It also defines all functionality, behavioral requirements, external interfaces, attributes, and performance of the application system. This document shall be used in designing the system. Specifying requirements is recognized as one of most difficult, yet important areas of application development. One particularly critical issue is the lack of real-life examples of requirements specifications. Thus this is the challenge that IT Personnel need consider especially who has practice so many standards at in their development activities.

(iv)To Analyze and Elicit the Requirements

In principle, the requirements come from users and stakeholders of the system. Part of requirements

work is to
elicit the requirements from stakeholders. The assumption is that stakeholders have some demands and the role of the
analyst is to elicit these demands, analyze them for consistency, feasibility, completeness and formulate them as
requirements. The people who are eliciting and formulating requirements are called requirement engineers by some
people, analysts by others. In practice, the analysts can be developers, expert users (preferably a team of both),
independent consultant, and marketing people and so on. Stakeholders include users with various roles, the customer
(who pays for the product), the customer's IT department and sometimes external parties with whom the customer co-
operates. If the system is a product offered to a broader market, stakeholders may include the distributors and
sometimes software houses adding special features to the product. Even when users earn express their needs, requirements engineers find it difficult to write them down in a precise way without designing the solution at the same time. The result is that the real demands and the written requirements do not match. For this reason, it is important for stakeholders to check that requirements meet their demands.

(v)To Validate, Verify & Trace

(a) Validation

The customer must be able to validate the requirements to see that they correctly reflect his needs. This means that he must be able to read the specification, understand it, and say "Yes, this is what I need. This system will solve my problems". In practice it is good idea to validate intermediate work products, for instance designs of screen pictures, to see that everything still matches the customer's expectations.

(b) Verification

Verification is carried out to check that the product satisfies the requirements. As a minimum, this is done in an acceptance test where the parties go through the requirements one by one d check that the product satisfies them. It is also a good idea to verify that intermediate work products satisfy requirements. Developers as well as customers need to convince themselves that all requirements are being considered during development.

(c)Tracing

Requirements tracing is needed to compare requirements against other information. There are three types of tracing: Forward, Backward and court cases.

(vi)Reviewing Requirements

Software requirement review are an essential component which play a major role in ensuring that every deliverable
produced under the IEEE 12207 Software Lifecycle Process is complete, accurate and consistent with the quality and control standards. Given their importance, project reviews cannot be omitted or overlooked without the prior consent of the Quality Reviewer Board which inclusive of XP's personnel and clients who were assigned to the project. The main aspect of requirement reviews covers:

The types of project reviews , The scheduling and conduct of project reviews, Quality Assurance Process Flow , Type of Requirements Review.

The requirement reviews can be broadly classified into one of the following groups:

Quality Reviews , Technical Reviewer , User Reviews.

In general, Quality Reviewers are appointed at the commencement of the project and continue to perform their specific role throughout the project lifecycle. Quality, User and Technical Reviewers are, on the other hand, often appointed during the project according to the particular skills required for the review and to fit in with resource and scheduling priorities. They all, however, play a significant part in ensuring the quality and accuracy of the deliverable items
produced and it is therefore important that the correct people are appointed.

(vii)To Manage Change and Control

Many higher education institutions have developed various applications for their community uses. This applications
contains much of data that includes sources codes, inputs etc. This data will be changed time to time, either systematically or automatically. A clear understanding of how an application that supports a business process has changed is the only way to reconstruct a transaction days, weeks or even months after it has occurred. This application can include mainframe databases, Web-facing applications, business-to-business processes, and third-party connections, among others. Thus, all presentation, application, and data tiers of the enterprise application architecture that handle many transactions and touch a business process must have sufficient controls. This is also tough challenge for IT Personnel because it's not so easy to manage the control and change. An appropriate technique and tools must be selected in order to manage change and control in enterprise application development.

4. Requirements Engineering in Web Methodologies

The development of Web applications has several characteristics that differ from the development of other kinds of applications. On the one hand, many different kinds of stakeholders participate in the

development process: analysts, customers, users, graphical designers, marketing, multimedia and security experts, etc. On the other hand, the main features of these systems are the navigational structure, the user interface and the personalization capability. The structure requires an intuitive guide to avoid that the user "gets lost in the navigational space" [21]. The design of the user interface often has to take into account multimedia and marketing aspects. These special design aspects not only have to be handled differently during design, but already be considered during the requirements specification [20].

4.1. WSDM: Web Site Design Method

WSDM is a user-centered approach for the development of Web sites that models the application based on the information requirements of the users' groups [19]. Its development process is divided into four phases:

•User modeling, where users are classified and grouped in order to study system requirements according to each user group.

•Conceptual design, where a class diagram is designed to represent the static model of the system and a navigational model to represent the possibilities of navigation.

•Implementation design, where models of the conceptual design are translated into an abstract language easily to be understood by the computer, and

•Implementation, where the implementation design result is written in a specific computer language.

We focus on the user modeling phase, which is the relevant one for this work. It aims on the identification of the different users' roles by performing the following two tasks:

• Users' classification is the identification of the potentials users/visitors of the Web site and their classification according to their interests and navigation preferences. WSDM proposes to analyze the organization environment where the application will be used, and centers the attention on the stakeholders of the business processes supported by the application. In WSDM the relationships between stakeholders and the business process activities performed are graphically represented by conceptual maps of roles and activities.

•Users' group description is the detailed description of the users' groups identified in the previous task. The information requirements, functional requirements and security requirements for each user's group are described with the help of a data dictionary. The remaining phases in the WSDM process are based on the users' classification of this first phase.

4.2. SOHDM: Scenario-based Object Oriented Hypermedia Design Methodology

The SOHDM approach [22] was the first approach stressing the importance of a process that allows the analysts to capture and define the applications requirements. SOHDM has similarities with OOHDM [23] among others, but it proposes a requirement specification based on scenarios. The following six tasks are performed during the life cycle of SOHDM: for this work, only the first one is relevant:

• Analysis, where requirements are describe using scenarios;

• Object model realization, where a class diagram is built in order to present the static structure of the system;

• View design, which expresses how the system will be presented to the user;

• Navigational design, where a navigational class model is developed in order to express the possibilities of navigation in the system;

• Realization of the implementation, where Web pages. The interface and also the database are developed; and. finally.

• Construction of the system, where the system is built.

The requirements definition starts on designing a so called context diagram, similar to the data flow diagrams (DFD) defined by Yourdon [24]. To build such a context diagram the analyst has to identify the external entities that communicate with the application, and the events that trigger the communication between these entities and the application. The set of events is specified as a table showing the entities that participate in an event. SOHDM proposes to associate a scenario with each event. Scenarios are graphically represented using a proprietary notation called SAC (Scenario Activity Chart). A scenario describes the interaction process between the user and the application when an event triggers an activity. It specifies the activity flow, objects involved and transactions performed.

SOHDM proposes a process to get the conceptual model of the application out of these scenarios. The proposed conceptual model is represented by a class diagram. The next step in the SOHDM development process is the regrouping of these classes with the objective to obtain a navigational class diagram.

4.3. RNA: Relationship-Navigational Analysis

RNA [16] is a methodology That offers a sequence of steps to develop Web applications focusing mainly on analysis. Its phases are:

• Phase 1 - Environment analysis: the objective is to analyze the audience's characteristics. Stakeholders of the application are identified and classified in different groups according to their roles (similar to the user modeling phase of WSDM).

• Phase 2 - Element analysis: in this phase all elements that are of interest to the application are identified, e.g. documents, forms, information, mock-ups. etc.

• Phase 3 - Meta-knowledge analysis; achieves to build a schema of the application. RNA proposes to identify objectives, processes and operations related to the application, and to describe the relationships between those elements.

• Phase 4 - Navigation analysis: in this phase, the schema of the previous one is enlarged with navigation features.

• Phase 5 - Implementation analysis: consists of the identification of how the models described in phase 4 will be produced in a computable language.

RNA only provides some guidelines of the actions to be performed in each phase. Neither modeling concepts nor a notation is proposed, but the RNA approach is one of the methodologies that first focused on the importance of requirements specification in the development process of Web applications. It emphasized the need of the separation between the analysis of conceptual requirements and the analysis of navigational requirements.

4.4. HFPM: Hypermedia Flexible Process Modeling

The Hypermedia Flexible Process Modeling (HFPM) presented by Olsina [25] is a wide engineering-based approach, which includes analysis-oriented descriptive and prescriptive process modeling strategies. It includes technical, management, cognitive and participatory tasks. Therefore. HFPM provides guidelines for the planning and managing of a Web project covering the whole life cycle of such a software project. It consists of thirteen phases; for each phase HFPM defines a set of tasks. For the purpose of this work, the most relevant is the Requirements Model whose related tasks are defined as follows:

• Problem description. HFPM does not prescribe a concrete technique to perform the problem description, e.g. natural language can be used.

• Description of functional requirements using use cases.

• Data modeling for the identified use cases. It proposes the design of a class diagram.

• User interface modeling using sketches and prototypes to be used in the presentation of drafts to the customer.

• Non-functional requirements description, such as security, performance, etc.

HFPM proposes on the one hand a detailed process to handle requirements. On the other hand it does not prescribe specific techniques, which can be chosen freely by analysts and developers.

4.5. OOHDM: Object Oriented Hypermedia Design Model

OOHDM is a widely accepted method for the development of Web applications [23]. whose first versions focused on design and did not include requirements engineering. The process in OOHDM is divided in four phases producing the following results:

•The conceptual model, represented as a class model, is built in order to show the static aspect of the system.

•The navigational model consists of a navigation class diagram and a navigation structure diagram. The first one represents the static possibilities of navigation in the system. The second one extends the navigation class diagram including access structures and navigation contexts.

•The abstract interface model is developed using a special technique named ADVs [23].

•The implementation consists in the implemented code and is based on the previous models.

The capture and definition of requirements were introduced later in OOHDM by Vilain,Schwabe and Sieckenius [26]. proposing the use of user interaction diagrams (UIDs). UIDs base on the well known technique of use cases. Use cases are used to capture the requirements but are considered in OOHDM as ambiguous and insufficient for the definition of the requirements that Web applications have, mainly related to the interaction between the user and the system. Therefore, for the specification of the requirements, this approach suggests the refinement of use cases building UIDs. which are used to graphically model the interaction between users and system without considering specific aspects of the interface. The process to get an UID from a use case is described very carefully in the approach.

4.6. UWE: UML-based Web Engineering

UML-based Web Engineering (LAVE) is a methodological approach for the development of Web applications based on the Unified Process [27][17]. It is based mainly on the most relevant concepts provided by other methods, but defines a UML notation (UML profile), sticks to the diagrammatic techniques proposed by the UML and defines a systematic and semi-automatic design process [28].

UWE covers the whole life cycle of Web applications and focuses on adaptive applications. It includes a specific requirements engineering phase where requirements elicitation. Specification and validation are handled as separate activities of the process. The final result of the requirements capture in UWE is a use case model completed with documentation describing the users of the application, the adaptation rules, the interfaces and the details of the use case relevant for the use case implementation. The latter can be described textually or modeled by UML activity diagrams. UWE

classifies requirements into two groups: functional and non-functional. Functional requirements contemplated in UWE are:

• Content requirements,• Structure requirements,• Presentation requirements,• Adaptation requirements • User model requirements

Moreover,UWE proposes interviews, questionnaires and checklists as appropriated techniques for the requirements capture, and use cases, scenarios and glossaries for the requirements specification. To validate them. UWE proposes walk-through, audits and prototypes [29].

5. Conclusion—

In this paper we have introduced a method for carrying out the Requirements Elicitation and Analysis activity for semantic applications. We have defined the tasks involved within the activity in the context of an iterative development life-cycle. This is particularly useful when application requirements are changing continuously during the whole application construction. The enterprise application development is complex, especially because it has multiple perspectives, objectives and purposes. With this seven steps, the IT Personnel should have clear picture and strategies how to development an enterprise application because the requirement for all environment is not similar with other enterprises that have so many requirements. we gave an outline of the methodologies for the Web describing how these approaches cover the aspects related to requirements engineering. As a result of our study we can claim that there is still a great research potential in the field of requirements engineering for Web applications.

6. References—

[1] Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., Guidi. D.: Towards a New Generation of Semantic Web Applications. IEEE Intelligent Systems 23 (2008)

[2] Motta, E., Sabou, M.: Next Generation Semantic Web Applications. In: 1st Asian Semantic Web Conference, Beijing (2006)

[3] Domingue, J., Fensel, D.: Towards a Service Web: Integrating the Semantic Web and Service Orientation. IEEE Intelligent Systems (2008)

[4] Krummenacher, R., Simperl. E., Fensel, D.: Scalability in Semantic Computing: Semantic Middleware. In: Proceedings of the IEEE Conference on Semantic Computing. (2008) 538-544

[5] Sommerville. I.: Software Engineering. Eighth edn. International Computer Science Series. Addison-Wesley (20O7)

[6] Lauesen, Soren (2002), Software Requirements: Style and Technique. Great Britain: Addison Wesley.

[7] Roger S. Pressman (2001). Software Engineering A Practitioner's Approach. 5th ed. United States of America: McGraw-Hill Companies.

[8] Peter Henderson, Yvonne Howard and Bob Walters, "A tool for evaluation of the Software Development Process", Journal of Systems and Software, Vol 59, No 3, pp 355-362 (2001)

[9]Peter Henderson, "Business Processes, Legacy Systems and a Fully Flexible Future", appears in Systems Engineering for Business Process Change, Springer Verlag, 2000

[10] California Software, "Application Servers and Enterprise Application Development", White Paper. Available: http://www.californiasw.com/tcchccntcr/javaapplicationdcvclopment.html

[11] ClearNova, Inc. "Enterprise Application Development Challenges: An Overview for Managers" White Paper, Available: http://itsolutions. forbes.com/forbes'

[12]Ian Alexander, 'The Basics of Requirements Management: Why Projects Have Requirements", While paper, January 2002. Ian Alexander's Home Page. Available: hup://easy web.casynct.co.uk/~iany/index.htm

[13]Ian Alexander, "Arc There Requirements for BPS?" Proceedings of Workshop on Requirements Engineering for Business Process Support(REBPS'03) Austria, June 17, 2003

[14] Baresi L.. Garzotto F.. Paolini P (2001). Extending UML for Modelling Web Applications. In proceedings of the 34th Annual Hawaii International Conference on System Science. IEEE Computer Society.

[15] Barry. C. Sc Lang. M. (2001) A Survey of Multimedia and Web Development Techniques and Methodology Usage. IEEE Multimedia. April-June 2001. 52-60.

[16] BieberM.. Galnares, R.. Lu. Q. (1998). Web Engineering and Flexible Hypermedia. The Second Workshop on Adaptive Hypertext and Hypermedia. Hypertext'9S. Pittsburg. USA.

[17] Booch G.. Rumbaugh, J.. Jacobson, I. (1999). Unified Modeling Language User Guide. Addison-Wesley.

[18] Ceri, S. Fraternali, P.. Bongio. A.. Brambilla M., Comai S.. Matera M. (2003). Designing Data-Intensive Web Applications. Morgan Kaufman.

[19]De Troyer. O.. Leune. C. (1997). WSDM: A User Centered Design Method for Web Sites. Technical Report of Tilburg University. Infolab. Belgium.

[20] Escalona. M.J.. Mejias. M.. Torres. J. (2002). Methodologies to develop Web Information Systems and Comparative Analysis. Infonnatik/Lifonnatique. num. 2/2002 de I/I.

[21] Olsina, L. (1999). Metodologia Qualitative!para la Evaluationy Comparacion de la Calidad de Sitios Web. Ph. Tesis. Faciiltad de Ciencias Exactas. Universidad de la Pampa. Argentina.

[22] Lee. H.. Lee. C, Yoo. C. (1998). A Scenario-based Object-oriented Methodology for Developing Hypermedia Information Systems. Proceedings of

31st Annual Conference on Systems Science. Sprague R
[23] Schwabe D.. Rossi G. (1998). Developing Hypermedia Applications using OOHDM. Workshop Hypermedia Development Process, Methods and Models. Hypeitext'9S. Pittsburg. USA.
[24] Yourdon E (1989). Modem Structured Analysis. Prentice-Hall.
[25] Olsina, L. (1998). Building a Web-based Information System applying the Hypermedia Flexible Process Modeling Strategy. 1st International Workshop on Hypermedia Development. Hypeitext'9S. Pittsburg.USA.
[26] Vilain. P.. Schwabe, D.. Sieckenius, C. (2000). A diagrammatic Tool for Representing User Interaction in UML. Lecture Notes in Computer Science. Proc. UML'2000. York. England.
[27] Jacobson I.. Booch G.. Rumbaugli J. (1999). The Unified Software Development Process. Addison Wesley.
[28] Hennicker, R.. Koch, N. (2000). A UML-based Methodology for Hypermedia Design. Lecture Notes in Computer Science. Proc. UML"2000. York. England.
[29] Koch. N. (2001). Software Engineering for Adaptive Hypermedia Applications. Ph. Thesis. FAST Reihe Sofnvaretechnik Vol(12). Uni-Druck. Munich. Germany
[30] Beck. K.: Extreme Programming Explained: Embrace Change. Addison-Wesley Professional (1999)