

Methodologies to develop Web Information Systems and Comparative Analysis

M. José Escalona, Manuel Mejías, and Jesús Torres

New systems are currently being developed which have different aims from the systems developed some years ago. Current systems tend to be distributed, with huge storage needs and complex functional requirements. But these systems are also distributed via Internet so interface, information recovery, navigation and multimedia are all fundamental aspects. These new systems are known as Web Information Systems (WIS). The complexity of these systems and the fact that they are systems requiring constant maintenance due to the fact that they continually change to adapt to users' new requirements means that they have to be developed using a standardized process. There is currently no standard methodology accepted by all the software engineering community. This paper presents the result of a comparative study analysing the different methodologies which are currently applicable to the development of a Web Information System.

Keywords: Web Information Systems, Navigation, Interface, Web Retrieval, Development Methodologies

1 Introduction

The characteristics of current software systems are different from those of systems developed some years ago. Storage and multimedia information retrieval needs, together with the existence of the Internet, give rise to very complex distributed software systems in which different user roles are identified. They also incorporate an extensive functionality in which navigation and user interface are of paramount importance. We call these systems *Web Information Systems (WIS)*.

In addition to the intrinsic complexity of these systems, their maintenance also tends to be complex. The fact that they are distributed via Internet requires them to be constantly updated and revised.

Another important aspect to consider with regard to these systems is the need to offer the user a simple and suitable navigation with a quality interface which provides the user with effective information retrieval. This, together with the fact that in many cases the user has multiple roles, each one requiring a suitable interface, means that interface, navigation and information retrieval are all vital aspects to be considered when developing WISs.

In spite of the attention that WISs are currently attracting, there is no standardized and complete methodology to serve as a development reference. This paper presents the result of an exhaustive survey of the most important proposals currently applicable to the development of WISs, and analyses the scope of application, the phases of the development lifecycle, and the techniques employed for each one.

In the second section of this document we go into the concept of the WIS and the relation between these systems and other current ones, such as digital library or multimedia application systems. In the third section, we present a short summary of the

methodological proposals that have been included in our survey and in the fourth section we make a comparative analysis of them. The paper closes with the conclusions we have drawn from our survey (section 5), and the bibliographical references used (section 6).

2 Web Information Systems

Due to the large number of terms currently in use in the software engineering world, and the lack of precise definitions for these terms, before starting our comparative study we need

María José Escalona qualified as a computer science engineer in 1999. During her years as a student and up until the year 2001 she worked as an analyst for various public organisations. She is currently writing her doctoral thesis in the *Department of Computer Languages and Systems of the University of Sevilla*, where she has also been teaching since the year 2000. Her doctoral thesis and her research work are oriented towards web environments, especially development methodologies. <escalona@lsi.us.es>

Jesús Torres received his doctorate in computer science in 1997 from the University of Sevilla. Since 1991 he has been a lecturer in the *Department of Computer Languages and Systems of the University of Sevilla*, where he is currently a full professor. His current line of research is in the areas of requirement engineering, web system development, user interfaces and heterogeneous distributed system architectures. <jtorres@lsi.us.es>

Manuel Mejías graduated as an industrial engineer in 1985 and was awarded his doctorate in industrial engineering in 1997, both by the University of Sevilla. He is currently a full professor at that same university, where he has been lecturing in software engineering since 1987. His research activity is centred on such fields as formal description techniques, communication system management, object oriented modelling of software systems, method environments in software development and software development processes. He is currently a member of the Association of Computer Professionals (ATI). <risoto@lsi.us.es>

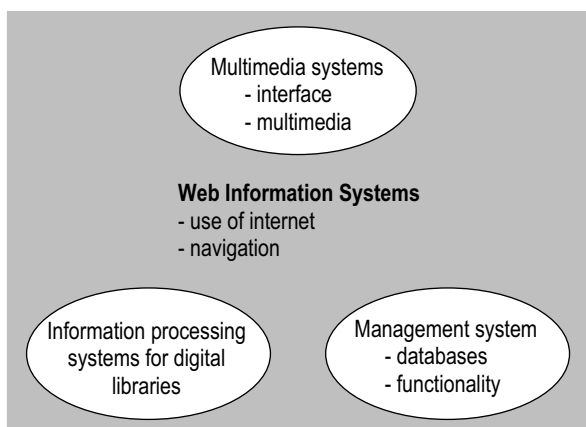


Figure 1: Relationship between Web Information Systems and other systems

to define exactly what types of systems we are going to consider as WISs. To this end we need to study what relation WISs have with others currently under development.

Until a few years ago, most of the systems developed were management systems which were aimed at solving problems arising in a specific environment. These systems put great emphasis on storage and functional aspects. The users of these systems are generally known and there are not a great deal of user roles to be found within these applications. Moreover, the interface tends to play a secondary role, since users only need to be trained how to use the applications properly.

In the mid 80s, a series of systems with a different focus began to appear; the so called *multimedia systems*. These systems tend not to be complex in terms of storage information requirements or functionality requirements. However, both the interface and multiple means of information storage (images, videos, sounds, etc.) are vital to any project's success.

Another concept to have emerged in recent years is that of *information processing systems for digital libraries*. A digital library is a library that has been expanded and improved by applying digital technology. It brings computers, storage systems and communication networks together with content and the software necessary to reproduce, emulate and extend the services provided by conventional libraries. A digital library must fulfil all the tasks of a conventional library and make use of digital technology advantages in storage, search and networks, as well as integrating new types of media (texts, images, sounds, videos, animations etc). A digital library provides a user community with coherent access to large, organized information repositories. Digital libraries are built (by collecting and organizing the information) by a community of users, and their functionalities depend on the informational needs of this community. Users' ability to access, reorganize and make use of this repository is enhanced by digital technology.

In the light of these definitions, we must ask ourselves what a WIS is. Any of the systems mentioned above may be considered as a WIS. Due to their information storage needs and their functional complexity, WIS are similar to classic information systems. However, the multimedia information which usually

appears in WIS and the need for a suitable, user-friendly interface make them similar to multimedia systems. Moreover, coming back to our definition of a digital library information system, if the WIS is aimed at managing a digital library, these concepts will be indistinguishable.

To sum up, the concept of WIS is a broad concept that includes what are known as multimedia systems, management systems and information processing systems for digital libraries. In figure 1 we can see how these are interrelated. In short, a WIS is a system of information made available over the Internet and is characterized by having large storage needs and complex functionality requirements, as well as other critical aspects such as interface, navigation and multimedia.

3 Proposals analysed

In the previous section we saw that a WIS is a broad concept which includes the other types of systems.

Therefore, when we are looking for a methodology to apply to the development of a WIS we need to look at other environments, such as multimedia or management systems, since there are currently no methodologies oriented towards these systems in existence. Fifteen proposals have been studied. Obviously, in this paper not all of them can be exhaustively presented. We will present the most significant aspects of each of them in

| Proposition | Reference | Environment |
|---|------------------|---|
| HDM | [Garzoto 1993] | Multimedia |
| RMM | [Isakowitz 1995] | Multimedia |
| EORM | [Lange 1995] | Multimedia |
| The MacWeb Hypermedia Design Method | [Nanard 1995] | Web and multimedia |
| OOHDM | [Rossi 1996] | Multimedia. In last versions oriented to the web |
| WSDM | [De Troyer 1997] | Web |
| OO-Method; OO-HMethod | [Pastor 1997] | Deals with management systems. OO-HMethod with multimedia |
| SOHDM | [Lee 1998] | Multimedia |
| RNA | [Bieber 1998] | Web |
| HFPM | [Olsina 1998] | Web and Multimedia |
| OO/Pattern Approach | [Thomson 1998] | Web and Multimedia |
| Unified Process | [Jacobson 1999] | Deals with management systems |
| Building Web Applications with UML | [Conallen 1999b] | Web |
| Specification and modeling of multimedia and hypermedia systems | [Mandel 2000] | Web and Multimedia |
| A UML-Based methodology for Hypermedia Design | [Hennicker 2001] | Web and Multimedia |

Table 1: Abstract of studied methodologies

order to have a better understanding of the results of the comparative analysis that we will be showing in the next section.

The methodologies are presented in chronological order. In table 1, all the proposals to be studied are listed, together with their references and the environment they work in.

3.1 HDM— A model-based approach to hypermedia application design

HDM (Hypermedia Design Model) [Garzoto et al. 93], is one of the first methods developed in order to define the structure and navigation characteristics of multimedia applications. HDM is based on the model E/R, although it extends the concept of entity and introduces new elements such as units or links.

In HDM the application is specified using an extended E/R model. This model represents the global structure of the application without being concerned with the development of the unit elements. To do this HDM defines new elements that can be added to the E/R model. HDM was the first approach to multimedia application modelling. One of the most important proposals is that it highlights the need to separate stored information from the information presented to the user. This idea is common to most of the following proposals.

However, HDM is not a methodology for the development of multimedia applications but simply a modelling technique. Although the elements defined by HDM (entities, perspectives, links, units, etc.) are used to define this type of applications, they are insufficient to guide the designer in the development process. Another fundamental problem with HDM is that it has become somewhat obsolete in the sense that the current design trend is to use the object-oriented paradigm. In response to this problem new proposals such as EORM or OOHDM have emerged, which, while sharing HDM's ideas and aims, adopt the object oriented paradigm for designing multimedia applications.

But one aspect which is missing from an HDM model is that it does not specifically deal with such aspects as interface, multimedia or information retrieval. HDM assumes that these aspects will be dealt with at a lower level of development.

To sum up, HDM would set the bases for future development proposals, offering ideas such as the separation of the conceptual part (or the information stored) from presentation (or the information presented). It was also the first attempt to standardize the development of multimedia applications, but it is far from being a methodological proposal for the development of WISs.

3.2 RMM— Relationship Management Methodology

RMM is proposed by Tomas Izsakowitz, Arnold Kamis and Marios Kounfaris in 1995 [Izakowitz et al. 95]. It can be considered as a methodology because it includes analysis and design phases. RMM proposes a process based on seven phases or stages, among which are navigation design, interface and testing.

The proposal is based on the E/R model and on HDM, since RMM includes all the extensions to the E/R model made by HDM. Taking them as a starting point, it defines a new model, RMDM, which proposes a language which allows developers

to describe the application's domain objects, interrelations and hypermedia navigation mechanisms.

RMM is a step forward with regard to HDM. It proposes a methodology based on the E/R model and on HDM to represent multimedia applications. Precisely for this reason it did not enjoy any great popularity; it is an E/R based technique when by that time (1995) the trend was towards object-oriented. One advantage that made RMM interesting was the fact that it proposes a defined and structured process to follow for developing applications. This process, however, lacks the first phases to be taken into account in any software development process, such as requirements gathering, for example. Also the phases of the process are too open-ended for it to be considered as a suitable development tool, because the only phase in which it specifies a technique is the one in which the RMDM model is used (phase 2). The other phases are left open to the designer's discretion.

3.3 EORM— Enhanced Object Relationship Methodology

Like OOHDM which we will be seeing later, EORM [Lange 95] is one of the multimedia application design methodologies which is most referred to in papers on the subject. It too was developed from RMM and HDM, but it uses the object oriented paradigm.

EORM proposes an iterative process consisting of enhancing an object model to represent the existing relationships between objects (*links*). Its structure has three phases: analysis, design and construction. In the first phase, a class model which represents the system without taking into account aspects such as navigation or interface is developed. The aspect of navigation is left for the design phase, in which the model obtained in the previous phase is enhanced to represent navigability within the system. In the construction phase, techniques are proposed to transfer the design model to the computer.

Since EORM is an object oriented proposal it guarantees all the advantages offered by this paradigm, but it also increases the possibility of reusing the applications, by defining a series of included libraries and repositories. EORM is also suitable because, following the original idea of HDM, it separates navigational from conceptual design. This guarantees reusability and easier maintenance. If there is a change in navigation, the conceptual structure is not altered. Also the application of EORM methodology can be made quite simple by using ODM-Tool, a tool developed by the author of EORM. This tool is used alongside the commercial graphical user interface builder, ONTOS Studio, and an object oriented data base management system, so as to allow interactive design of EORM schemas and automatic code generation, initially in C++.

In spite of all these advantages, certain criticisms can be levelled at EORM. First of all its methodological process is insufficient in many cases, mainly because it only deals specifically with aspects of storage and navigation, leaving aside such issues as system functionality or interface aspects. Neither does it mention at any time the techniques to use in order to obtain the models it proposes, or the products that should be generated in the development. EORM also leaves aside a very important aspect in most applications: requirements gathering. Not only

does it not offer any proposal but it gives no indication of any that might be used.

In summary, EORM is the first object oriented proposal which offers a series of ideas and models that are very suitable for representing navigational and conceptual design, but it omits critical aspects in the development cycle such as requirements gathering and definition, and specification of the different user roles.

3.4 The MacWeb Hypermedia Development Environment

The MacWeb Hypermedia Development Environment, proposed by the Nanard brothers in 1995 [Nanard/Nanard 95] cannot be considered as a real methodological proposal. In the MacWeb environment, a hypermedia application is developed on the basis of the interface. The authors of this environment stress that the most important thing in a web application is communication with the user. They therefore focus all the development process on the interface. According to this proposal, a hypermedia application design must be divided into two major phases: the *development of the mental process*, in which the work group must know the environment where the system and working mechanisms are going to be implemented; and the *execution of the methodological steps*, in which the system's conceptual model or basic class model, the navigational model and the interface are all designed, and in which the implementation is carried out and the testing plan is executed.

The MacWeb development environment proposes the use of classic object oriented techniques, such as generalization and instancing to represent navigational aspects. Together with MacWeb, a tool was created which helps to execute the development process. This tool has predefined classes such as nodes or links.

The MacWeb development environment is interesting because of the separation it proposes between content and development aspects. This separation is found in real developments in which IT specialists have to develop a multimedia application about a subject on which they are not usually experts. The collaboration between thematic and technical experts is becoming ever more necessary, and consequently both aspects need to be borne in mind in the development process. It is also the first proposal to take the user into account in the system development cycle.

3.5 OOHDM— Object Oriented Hypermedia Design Model

OOHDM is a development methodology proposed by Rossi and Schwabe [Rossi 96] for the production of multimedia applications. OOHDM is based on the HDM model in so far as it uses many of its definitions, especially in the navigational aspects. However, OOHDM is far superior to its predecessor, since it is not simply a modelling language, but it defines some working patterns, mainly design focused, to develop multimedia applications in a methodical way. OOHDM has evolved a great deal since it first appeared. It is currently being used by its authors for the development of web applications.

OOHDM divides the development process into four phases. In the first phase, *conceptual design*, the system's class model

is produced. After that, navigation is modelled in the *navigational design* phase. This is done by means of two models: navigational classes and navigational contexts. These models are based on object orientation and they are designed taking the conceptual model of the previous phase as the starting point. The following step is the *abstract interface design*. Here, by using a graphical interface abstract modelling language, and abstract data views (ADVs) [Schwabe/Rossi 01], an interface representation is produced without going into any design aspects. Finally, in the implementation phase, these models would be transferred to the computer.

OODMD is without a doubt one of the most widely accepted methodologies in multimedia application development [Schwabe/Rossi 98]. It is currently being used as the basis for the development of new methodological proposals for WIS [Mandel et al. 00]. OOHDM is a design oriented proposal which makes use of a series of ideas which have been adopted by many other proposals, and which have produced some very good results. The first of these proposals is that it makes a clear distinction between conceptual, navigational and visual design. This independence makes for much easier application maintenance. It is also the first proposal which makes an in depth study of interface aspects, which is essential not only in multimedia applications but is also a critical point in any system developed these days.

OOHDM also uses the object oriented and such a standardized diagram as the class diagram in order to represent the navigation aspects by means of what is known as navigational classes. This idea has produced very good results and it seems very adequate when working.

OOHDM also uses object orientation and the highly standardized class diagram in order to represent navigational aspects by means of what is known as navigational classes. This idea has produced very good results and seems very to work very well. However, in spite of all this, OOHDM does have some drawbacks. OOHDM has omitted one essential aspect from its environment which is the handling of the system's functionality. What can be done in the system, and at what moment in the navigation or the interface it can be done, are things which are not dealt with and are left as a part of implementation. Furthermore, OOHDM does not offer any mechanism to work with multiple actors. For example, let us imagine that the interface and the application navigation vary substantially depending on who is connected to the application. The navigational diagram, the navigational contexts and the ADVs could all be too complex to represent this variability. Another OOHDM proposal which does not appear to be suitable concerns navigational contexts. The degree of complexity that they could reach in large systems may make their use unviable.

To sum up, OOHDM has a series of characteristics which are very interesting for developers looking for a methodology to take navigation and interface into account. There are many examples developed using OOHDM which help show the viability of these ideas. However, the use of this proposal has to be limited to simple web or multimedia applications, with minimal functional complexity.

3.6 WSDM– Web Site Design Method

The WSDM is a proposal in which the system is defined on the basis of user groups. It was proposed by De Troyer and Leune in 1997 [De Troyer/Leune 97] and it has four phases: *user modelling*, in which the users that are going to interact with the system are studied and defined; *conceptual design*, where the conceptual model is studied and the system's navigational model is designed by means of a class diagram; *implementation design*, where the interface and the user environment are designed; and, finally, *implementation*, where the designed system is coded.

WSDM authors divide the web sites into two groups: *web kiosk* and *web applications* [Izakowitz et al. 95]. The former offer users a specific piece of information and let them navigate towards it. Web applications comprise those interactive information systems whose user interface is a set of web pages. Having made this division, WSDM's authors specify that their proposal is oriented towards the former (web kiosk), and so is not suitable for web applications.

In principle WSDM's proposal may be quite interesting in the sense that it offers a new perspective on the handling of users. It is a methodology which is completely oriented towards designing the application based on user groups from the outset. However, it makes no mention of aspects that may be important with this approach. For example, the same information could be presented to two users in two different ways. In many cases, it may be difficult for the programmer to detect that it is the same information. Furthermore, as the authors themselves point out, it can only be used to develop web kiosks, that is, applications that show information. This proposal does not deal with aspects such as functionality, security, etc. which are necessary in web applications. It only focuses on how to present the information to the user.

3.7 OO-Method and OO-HMethod

OO-Method [Pastoret et al. 97] is an object oriented method developed by the Polytechnic University of Valencia which combines the use of a formal specification language (OASIS [Pastor et al. 92]) with a graphical notation taken from the most commonly used standards. This approach makes use of the positive properties of formal specification languages with the experience accumulated in methods commonly used in industrial software production. OO-Method methodology has evolved as a result of recent work into OO-HMethod [Gómez et al. 01]. This is simply an extension of OO-Method in which a new model is added to represent user interoperability in web systems.

The basic principles of OO-Method are that it supports notions of object oriented conceptual modelling, and permits the use of the concepts of object oriented and formal specification languages. Thus, it designs the system by means of three models: object model, dynamic model and functional model. OO-Method also has an advanced automatic prototyping environment which includes both automatic generation of a formal, object oriented specification in OASIS, and a prototyping environment functionally equivalent to the conceptual model, plus

complete code generation in imperative environments such as Visual C++ or Delphi.

However, OO-Method does not cover tasks such as requirements definition, and once again it is a methodology oriented almost entirely to the design and implementation phase.

3.8 SOHDM– Scenario-based Object-Oriented Hypermedia Design Methodology

Another proposal, somewhat more recent than the previous ones, is proposed by H. Lee, C. Lee and C. Yoo: SOHDM [Lee et al. 98]. This proposition has six phases:

- Analysis, in which a requirements gathering is carried out based on scenarios.
- Object Modelling, where the system is modelled by means of a class model.
- View design, which studies how the above model objects are going to be presented.
- Navigational design, where the navigation is designed following OODHM proposals.
- Implementation design, where the pages are created.
- Construction, where the application is implemented.

SOHDM is quite similar to its predecessors RMM, OOHDM and EORM. However, there is something which makes this methodology different from the previous ones, which is that the development of the system is scenario based. Thus, SOHDM is so far the only proposal that takes into account such aspects as requirements gathering based on scenarios. It is quite an interesting proposal since it covers all the development process phases, while leaving aside implementation and testing.

SOHDM is a relatively new proposal that has not been used much, though it is being constantly revised [Suh/Lee 01]. It has the advantage of being a simple process to follow, although some might criticise the fact that its nomenclature is quite closed. For example, for the interface development, it defines how an image or a button is represented within the model, although nothing is said about how to represent an audio element, and the designer is not given the option of defining his own representation.

3.9 RNA– Relationship Navigational Analysis

RNA was proposed by Bieber, Galnares and Lu in 1998 [Bieber et al. 98]. RNA proposes a sequence of steps for the development of web applications, focusing mainly on the analysis phase. Although this proposal is oriented towards the development of applications dealing with legal matters or law, its ideas can be applied to other environments. However, RNA does not present any new proposals of models or techniques with which to carry out the process. Instead RNA offers a sequence of steps to follow in order to carry out the analysis of web applications, without saying how these are to be done.

It has been included in our study because it can be interesting when considering methodologies for application development, especially in the analysis phase. RNA stresses the importance that the user study (phase 1), the classic concepts (phase 3) and navigation (phase 4) all have in web applications. The idea of considering these three aspects independently and studying them individually in different phases is both widespread and

highly suited to the development of these applications. It is also the only proposal which stresses the importance of studying the environment and the elements of interest in order to understand the environment and the full scope of the problem before tackling it.

3. 10 HFPM– Hypermedia Flexible Process Modelling Strategy

HFPM was proposed by Luis Olsina in 1998 [Olsina 98]. It is the only one of all the methodologies proposed in this section which covers all phases of the development process, from analysis to the development of documentation and maintenance. It also separates and lists in detail each of the tasks included in every phase. However, although the tasks and subtasks to execute are given, this proposal does not offer new models or modelling techniques. It is broadly based on OOHDM and on classic object oriented methodologies (OMT, UML, etc.).

HFPM divides the development process into the thirteen phases listed below, but, as has already been mentioned, it does not state the techniques to be applied in each of them, it only gives the process.

- Requirements modelling, both functional and interface or navigational.
- Execution of project planning.
- Conceptual modelling following OOHDM guidelines.
- Navigational modelling, following OOHDM guidelines.
- Abstract interface modelling, designing the interface prototypes.
- Environment and architecture design.
- Design of multimedia aspects.
- Implementation of the system in machine language.
- Verification and validation of the results by means of a testing system.
- Evaluation of the environment where the application is to be implemented.
- Evaluation of product quality.
- Planning of the maintenance process.
- Documentation of the models, the testing and the user manual.

In short, FPM integrates the classic object oriented proposals with OOHDM, reuses the models they present, and offers clear guidelines to follow in the development process.

3. 11 OO/Pattern Approach

The approach proposed by Thomson, Greer and Cooke in 1998 [Thomson et al. 98], OO/Pattern Approach, is quite similar to HFPM, since both propose the use of patterns and object orientation for navigational design and the interface. However, this proposal, unlike HFPM, does not cover the whole development cycle. OO/Pattern Approach is nevertheless interesting because it is the first proposal which uses the technique of known use cases [Jacobson 95] to execute the application analysis phase.

This methodology divides the life cycle into six phases. In the first one, *design of the use cases*, a study of use cases is carried out which will be the basis for the class diagram to be produced in the second phase of the life cycle, the *conceptual*

design. After this comes *collaboration design*, for which the use of the collaboration diagram technique proposed by UML [Booch et al.99] is suggested. After the collaboration design, the *class data dictionary* must be produced. In principle no format is proposed for it, but it is interesting since this is the only proposal that stresses this need. The fifth phase is *navigational design*. The model proposed for this phase is quite similar to OOHDM's model, since it uses a class diagram to represent navigation. Finally, the system *implementation* must be executed.

This methodology is quite complex in so far as it makes reference to the analysis phase, design and implementation. However it does not make it very clear what documentation should be presented and there is no reference to the interface. However, it is important that it establishes the need for producing a data dictionary for the conceptual model.

We can therefore conclude by saying that OO/Pattern Approach is a proposal that covers all the system's life cycle, which shows that this is necessary. However, it does not deal with basic aspects such as the interface nor does it say what has to be done in each of the phases. It only states what objectives should be reached in each one.

3. 12 The Unified Process

Since the development of computer applications started to be considered as an engineering process, a number of development methodologies have emerged to support the project development cycle, the most important of which are MÉTRICA, MERISE, SSADM and, more recently, OMT [Rumbaugh 91] or the current Unified Process or RUP [Jacobson et al. 99]. Clearly, in a study of methodologies applicable to the development of a WIS, we could not leave out the classic proposals. Of all the ones to choose from, we have opted to include the Unified Process in our study as the most widely accepted nowadays.

Little can be said about RRUP that is not already known. RUP is a process based on the UML modelling language and was developed by the same authors of this language. It consists of a set of activities to be executed in order to carry out the development of a software product. It proposes a life cycle divided first into cycles, then into four phases, and finally each phase is further divided into iterations. Each iteration is composed of five work flows: requirements gathering, analysis, design, coding and testing. For each iteration RUP proposes the activities to execute and the products to obtain.

However if what we want to do is to develop a WIS and we decide to use Unified Process, we will discover the shortcomings this proposal suffers from with regard to representing aspects such as the user interface, multimedia information types or navigational complexity within the system. This is due to the fact that UML and RUP are oriented towards dealing mainly with functionality and information storage aspects.

In spite of the fact that when we try to apply the Unified Process to the design of WIS we will find shortcomings with regard to expressing some aspects, it is clear that UML's nomenclature and life cycle provide a standard that is giving very good results. In fact, most of the proposals analysed use it.

3.13 Building web applications with UML

Having seen the shortcomings shown by RUP and UML with regard to the modelling of WIS in general, different research groups have tried to solve the problem by enhancing UML with new resources to be able to represent aspects required by web applications. These research groups fall into two different categories, each following a different trend.

The first one would include researchers who aim to enhance UML by adding new stereotypes which make it possible to represent those aspects that are typical of web applications. In this respect, it is interesting to analyse the proposal presented by Conallen [Conallen 99a]. UML's creators admitted that UML was not a perfect proposal for all kinds of applications, so alongside it they defined a formal path for the definition of new concepts that would cover the needs of new systems. Conallen has made use of this proposal, and seeing the shortcomings shown by UML with regard to designing web applications, he presented a proposal to extend the graphical language with new stereotypes intended for the Internet and the Web.

The stereotypes proposed by Conallen use some graphical representations that he himself makes available to the stakeholder and which can be included in the tool *Rational Rose*. Among these stereotypes are those used to represent client pages and server pages, frames, calls to remote procedures, script codes, etc.

However, although these contributions do make it interesting, this proposal is not a methodology. It simply defines these new stereotypes which can be included in web system representations. And while this UML extension offers semantics to represent such concepts as web pages or links, it does not offer any guidelines about how to represent information stored in multiple media, or how to represent navigation or the user interface. These stereotypes are very close to the implementation phase and they cover such aspects as whether a script is written in Java or not. WISs are not only special at lower level design and implementation phases. Their differences must be dealt with from the very first phases and this extension does not provide us with enough tools to do this.

3.14 Specification and modelling of multimedia and hypermedia systems

The other trend that emerged when RUP problems for the new systems were discovered is based on multimedia system development methodologies, especially OOHDM. They aim to enhance UML, not by means of stereotypes, but by means of new models which make it possible to represent multimedia and navigation. These proposals mainly emerged in Germany, and they have recently given rise to the process for specification and modelling of multimedia and hypermedia systems [Mandel et al. 00].

In this process, there are three design sub-phases within the design flow: conceptual design, navigational design and presentation design. New models and stereotypes which are adapted to the modelling system requirements are proposed for each phase. The most important contribution of this proposal is not, however, these new models, since most of them already appear in proposals such as OOHDM or Conallen's. Its main contri-

bution is that it offers detailed techniques describing how to obtain these models. However it has to be said that both this new proposal and Conallen's add new semantics and a larger representative power to UML, and come quite near to meeting the needs of WIS development.

However, both are too close to design and implementation, and they omit such important flows as requirements gathering and analysis, presupposing that WIS can be handled in these phases in the same way as classical systems. These proposals do not take into account the need for user participation in the development process. In WIS the client's and the users' opinions regarding the interface are essential if the solution is to succeed.

3.15 A UML-Based Methodology for Hypermedia Design

Within the above authors' research environment, a new paper has recently appeared which, while based on the previous ones, comes up with a much more elaborate methodology for the development of hypermedia applications [Hennicker/Koch 01].

This proposal comes from Rolf Hennicker and Nora Koch and it deals with hypermedia applications in a broader sense than normal. When they talk about hypermedia in this paper, they are talking about applications which deal with multimedia elements and hyperlinks, as well as major data warehouses, all in an Internet environment. They are, at the end of the day, what in this paper have been called Web Information Systems. This proposal, like the previous one, bases the development of the application on three aspects:

- The content, represented by means of the conceptual design.
- The navigation structure, represented by means of the navigational design.
- The presentation model, represented by means of the presentation design.

However, one of the most important aspects this proposal contributes compared to the ones before is that it adds the necessity of starting with a previous phase of requirement definition and gathering. To do this it proposes the use cases technique, although it does not say how to get these use cases to capture the concepts needed to define the navigation or the interface. This is precisely one of the aspects that is open to criticism. For the simple reason that it is based on previous proposals, we find the same faults regarding the complexity of its models and the fact that it does not take the user into account in the interface development phase.

Finally, we should point out that this proposal is totally process oriented. It never indicates what has to be presented to the user or what the final results of the work are. It only tells you the models to obtain.

3.16 Other proposals studied

New trends and new proposals are currently emerging in the multimedia and web environment. However, it would be somewhat unviable to cover all of them in this article. In previous sections we have analysed the most relevant of all the proposals. But we cannot conclude this section without mentioning other emerging trends that are becoming widely known. The fact that they have not been studied in detail is due to the fact that

either they do not add any new ideas to the previous proposals or that there is not much published about them.

WebML (Web Modelling Language) [Ceri et al. 01]. is in this situation. This proposal comes from Stefano Ceri, Piero Fraternali and Aldo Bongio from Milan. It is interesting because it has a tool that makes it possible to implement everything that is developed under XML.

Another important piece of work, known as the Araneus project, has been produced by Paolo Atzeni and Paolo Merialdo from the University of Rome and by Giansalvatore Mecca from the University of Basilicata [Mecca et al. 99]. Araneus is a specific data model to define application schemas with hypertexts. It is oriented towards the database world and the E/R model.

Another interesting proposal is OSM. This is the brainchild of Liddle, Embley and Woodfield [Liddle et al. 01a]. OSM is not a methodological proposal per se. It is an object oriented model that aims to be solid enough to support all the life cycle phases of a software development project (specification, analysis, design, implementation and evolution). It consists of three submodels:

- Object model and relationships between them
- Behaviour model
- Presentation model

OSM's structural components are the objects and the relationships between them. The objects relate with one another through two abstractions: generalization/specialization and aggregation. An object has a unique identity and is active. It can also perform concurrently with other objects. For their part the relationships group together in relationship sets and they can be treated as objects by giving them all the properties that an object could be given. In OSM, behaviour is represented by using a state diagram. Each object has an associated set of sets. To change state events that bring about the transition from one state to another must take place. Finally, the relationship with the exterior is by means of interaction objects. An object can synchronize and communicate with other objects through interaction. In [Liddle et al. 01b], an example of a system modelling using OSM can be found. It actually takes ideas from several models (class model, state diagrams, etc.), and it puts forward new ways to represent structure, behaviour and system interoperability.

But the most important thing about OSM is that it can be translated directly into a formal language, OSM-Logic, which consists of formulae used to represent those systems. One important benefit of this formalization is that an executable model can be defined from an OSM model using formal and logical reasoning. OSM's authors also present a tool which allow us to obtain an executable model.

4 A comparative study

Now we have presented the surveyed methodologies, in this section we are going to make a series of comparisons between them. These comparisons will be useful to analyse the degree of suitability of these methodologies for developing a WIS. Considering the classic phases of the every development

process: specification, analysis, design, coding, testing and maintenance, the first appraisal we made of the surveyed methodologies shows which of these generic phases are dealt with in each proposal.

4.1 Requirement specification

Starting with requirement specification, we reach the conclusion that this is a phase that few proposals handle, only HFPM, SOHDM and Unified Process talk about it. Other proposals like OOHDM or EORM assume that methodological proposals like OMT would be suitable for requirements gathering. Among the techniques available for carrying out requirements gathering, basically only two are mentioned: use cases diagrams and scenarios. Both of them cover the system's functional requirements adequately, but they do not make it possible to specify what the user wants to see or how he wants see it. As has already been mentioned, in WIS, communication with the user is vital in order to get a good result, and the use cases and scenarios techniques are not enough. So, this study leads us to the conclusion that what is needed is a proposal to carry out requirements gathering that offers the possibility of having a fluent communication with the user in order to identify the system's requirements and objectives correctly

When considering the issues of communication and the study of the environment where the system is going to be set up, the proposals made by RNA seem worth looking at. We should recall that this methodology did not offer enough design mechanisms for the whole process development, and was oriented exclusively oriented to the specification and analysis of legal applications on the Web, though it did orient project development toward the user. Its ideas about studying the work environment and system requirements could be appropriate in a proposal for the development of WIS.

4.2 Analysis

Moving on to the analysis phase, this is included in a number of proposals which cover the need to approach this phase in a more specific manner. In this phase all the aspects critical to WIS must be analysed and modelled. Of the proposals studied, RNA and WSDM focus the analysis on the study of user groups, while the others focus on producing a class model at a high level to represent the system's conceptual model. Both ideas are valid, but neither is enough by itself. WISs can vary substantially, especially in their interface and work possibilities, depending on the user who is working with it at any given moment. They also tend to be systems with very complex internal information storage requirements. Therefore, this complexity has to be studied and represented in the first phases of the life cycle, and it would appear that the class model is a suitable technique for this. Thus, in a WIS analysis, both the conceptual model and the system variability regarding users have to be studied. But the study of functionality or the interface, of navigation or multimedia cannot be forgotten either. Consequently, a proposal capable of analysing all these aspects must be produced.

| Methodology | Spec. | Analysis | Design | Codif. | Tests | Mainten. |
|-----------------------|-------|----------|--------|--------|-------|----------|
| HDM | | | X | | | |
| RMM | | | X | X | | |
| EORM | | | X | X | | |
| MacWeb | | | X | X | X | |
| OOHDM | | | X | X | | |
| WSDM | | X | X | X | | |
| OO-Method | | | X | X | X | X |
| SOHDM | X | X | X | X | X | |
| RNA | | X | | | X | |
| HFPM | X | X | X | X | X | X |
| OO/Pattern | | X | X | X | X | |
| RUP | X | X | X | X | X | |
| Building Web ... | | | X | | | |
| Specif. & modeling... | | | X | | | |
| A UML-Based ... | | X | X | | | |

Table 2: Phases considered in each proposal

4.3 Design

The design phase is the most developed phase in all the methodologies studied. All of them, with the exception of RMM and HDM, adopt the object oriented paradigm since it is better suited to these systems' characteristics. Another idea adopted by most proposals is the need, in the design phase, to differentiate conceptual representation from the interface and navigational aspects.

Moving on to the *conceptual design*, most proposals use a class diagram to represent the system's basic storage needs, a technique which has given very good results in all of them, and which is readily adoptable because of how standardized it is.

With regard to the *navigational design*, two separate parts have to be differentiated. On the one hand, most proposals adopt the original idea of OOHDM and EORM of representing the navigation through a class diagram. However, when this navigation varies depending on the context in which we are working, each of these contexts has to be differentiated. There are multiple ways in which these contexts can be represented: OOHDM navigational contexts, Conallen's and Koch's proposals, etc. However, all of them are very complex in systems with complex navigation. Thus alternatives ways of representation need to be found.

Regarding the *interface*, almost all the proposals stress the use of techniques which represent the interface in an abstract way, and they consider it necessary to represent both the screen structure and its dynamism. But most of these models do not appear to be suitable. Some of them, such as the ones proposed by Conallen and Koch, are too close to the implementation, and in many cases, especially when the applications are complex, it is easier to present the designed screen than the model. Other proposals like ADVs from OOHDM are too ambiguous, and the level of abstraction to be reached is not clearly indicated.

| | HDM | RMM | EORM | MacWeb | OOHDM | WSDM |
|-------------------------------------|-----|-----|------|--------|-------|------|
| ERD | X | X | | | | |
| Class diagram | | | X | X | X | X |
| Navigational classes: nodes, links. | | | X | X | X | |
| Secnaries | | | | | | |
| GUI Techniques | | X | X | X | X | X |
| Pattern design | | | | X | | |
| Use cases | | | | | | |
| Other models and techniques | | X | | | X | |

| | OO-Method | SOHDM | RNA | HFPM | OO/Pattens |
|-------------------------------------|-----------|-------|-----|------|------------|
| ERD | | | | | |
| Class diagram | X | X | | X | X |
| Navigational classes: nodes, links. | X | X | | X | X |
| Secnaries | | X | | | |
| GUI Techniques | | | X | X | X |
| Pattern design | | | | X | X |
| Use cases | | | | | X |
| Other models and techniques | X | | | | |

| | RUP | Building web ... | Specification & modelling... | A UML-Based ... |
|-------------------------------------|-----|------------------|------------------------------|-----------------|
| ERD | | | | |
| Class diagram | X | X | | X |
| Navigational classes: nodes, links. | | X | X | X |
| Secnaries | X | | | X |
| GUI Techniques | X | X | X | X |
| Pattern design | | | X | X |
| Use cases | X | | | X |
| Other models and techniques | X | | | X |

Table 3: Techniques and models used in the proposals

Thus, the representation must be standardized and some guidelines must be set for the designer to follow in order to obtain prototypes.

Another thing that none of the proposals do is to consider it necessary to study the system's architecture or its modular division. This is due to the fact that multimedia and web applications, which most of these proposals are concerned with, are simple and this study is not considered necessary. In WIS these activities must be considered essential in the design. It would also seem to be a good idea to use design patterns, which are so widely used these days. Only MacWeb makes any mention of

this technique. Using these patterns can enable reusing and lead to a less costly system maintenance.

4.4 Implementation

With regard to the implementation phase, few proposals do not mention it. But, practically none of those that do propose implementation techniques. Perhaps Conallen's and Koch's proposals come closest to this, since their models are so close to implementation that it is easy to obtain the code. The OO-Method proposal is also interesting and so is the tool which generates the code it provides to work with.

4.5 Testing and maintenance

The testing and maintenance phases are mentioned in some of the proposals, but none of them offers techniques or methods to apply when executing these phases.

By way of summary, table 2 presents all the proposals surveyed, analysing which phases they stress and in which phases they provide new proposals.

Table 3 also summarises the proposed techniques, and shows which methodologies apply each one.

It can be seen that in the earlier proposals, the tendency was to take ERD as a basis, but this trend has evolved towards the object oriented paradigm. It can be observed that in later proposals the use of requirements gathering techniques and the use of scenarios or use cases has gradually become more popular. The reason for this is that, as these techniques have evolved towards more complex systems, the need for a requirements definition mechanism has also increased.

It can also be seen that there are a number of models that propose their own techniques and models, although if we

analyse the all the proposals which have been presented in the previous section, we will see that they are all quite similar.

Finally, perhaps we should highlight some important aspects that seem suitable for WIS development. These aspects can be summarised as follows:

- a. A proposal for WIS should offer mechanisms to handle application variability depending on the actor that uses it. In the first column of table 4, it can be seen that only WSDM deals with this aspect.
- b. To ensure reusability and to facilitate maintenance in WIS it is advisable to separate the basic design or conceptual design from the navigational design. As is shown in table 4, this is a very widely adopted idea.
- c. For the same reason as in point b, it is also advisable to separate the navigation from the interface. In the third column of table 4, it can be seen that not all the proposals offer this possibility.
- d. Also a proposal for WIS must offer mechanisms to deal adequately with multimedia. As is shown in table 4, only the multimedia methodologies deal with this aspect.

To summarise the conclusions of this comparative study, it may be said that:

- A great majority of the proposals use object oriented modeling techniques.
- A great majority of the proposals focus mainly on design, and most assume the analysis has already been carried out or propose use cases or scenarios to do it.
- A very large number of proposals consider it necessary to separate the navigation and the interface design from the design per se.
- None of them provides a reference framework to work with WIS, since they do not completely cover all the aspects typical of this type of system.

| Methodology | Multiple users | Separation Conceptual-Navigation | Separation Navigation-Interface | Multimedia |
|--------------------------|----------------|----------------------------------|---------------------------------|------------|
| HDM | | X | | X |
| RMM | | X | X | |
| EORM | | X | | X |
| MacWeb | | X | X | |
| OOHDM | | X | X | X |
| WSDM | X | X | | |
| OO-Method | | X | X | |
| SOHDM | | X | X | X |
| RNA | | X | | |
| HFPM | | X | X | |
| OO/Pattern | | X | | |
| RUP | | X | | |
| Building Web ... | | | | X |
| Specific. & modeling ... | | X | X | |
| A UML-Based ... | | X | X | |

Table 4: Abstract about treated aspects

5 Conclusions

After the conclusions obtained from the comparative study in section four, little else can be added except that having studied the different possibilities that software engineering offers the designer to build a WIS, we can conclude by saying that no methodology offers a wholly suitable framework for the development of WIS. A methodology for WIS should cover all phases of the life cycle, indicating the activities to follow in each of them, the techniques to apply, and the products resulting from them. But it must also allow us to give an adequate treatment to the classical aspects of management systems, such as storage and functionality requirements, as well as the new characteristics that WIS inherit from multimedia: navigation, the importance of the interface and multimedia. As we have seen in section four, none of the current methodologies offers all this.

Another aspect detected on studying these methodologies is the need to harmonise definitions. The authors mentioned in our article work with the same terms: navigation, abstract interface, etc. But what they mean by with these terms is not always the same.

By way of summarising section 4, we can also say that, in their work, all the authors stress the need for a standard and a

great deal of them agree on many of the ideas which seem to have been adopted by most of the research groups in this environment: the idea of separating the conceptual from the navigational and the interface, the complexity of communication with the user, etc.

Finally, we should point out that, although these authors work independently in their proposals, in recent years they have been making an effort to standardize criteria and to achieve that standard. Thus, workshops such as I'WEST [I'WEST 2000] or investigation projects such as Hyper-UML [Mandel et al. 00], have emerged and are moving towards this objective.

In conclusion, in this paper we have made a survey of all the methodological proposals which are currently enjoying most popularity in the software engineering world, and which are being applied to the development of WIS. And we have provided a comparative study of them in order to see which tools are currently available to the designer and which aspects have yet to be covered.

References

- [Bieber et al. 98]
M. Bieber, R. Galnares and Q. Lu. Web engineering and flexible hypermedia. The 2nd Workshop on Adaptive Hypertext and Hypermedia, Hypertext 1998.
- [Booch et al.99]
G. Booch, J. Rumbaugh, I. Jacobson. Unified Modelling Language User Guide. Ed. Addison-Wesley, 1999.
- [Ceri et al. 01]
S. Ceri, P. Fraternali, A. Bongio. Web Modelling Language (WebML): a modelling language for designing Web sites. First international workshop on Web-Oriented Software Technology. Valencia, 2001.
- [Conallen 99a]
J. Conallen. Building Web Applications with UML. Addison Wesley 1999.
- [Conallen 99b]
J. Conallen. UML Extension for Web Applications 0.91. Available in <http://www.conallen.com/technologyCorner/webextension/WebExtension091.htm>
- [De Troyer/Leune 97]
O. M. F. De Troyer, C. J. Leune. WSDM: A User Centred Design Method for Web Sites. Tilburg University, Infolab. 1997
- [Escalona et al. 00a]
M. J. Escalona, J.Torres, M. Mejías. Propuesta de metodología para el desarrollo de sistemas para el tratamiento de bibliotecas digitales, Report Interno, 2-2000 del Departamento de Lenguajes y sistemas Informáticos. Universidad de Sevilla. Sevilla, Junio 2000.
- [Escalona et al. 00b]
M. J. Escalona, M. Mejías, J. Torres. Aproximación metodológica al desarrollo de sistemas para el tratamiento de bibliotecas digitales. V Jornadas de Ingeniería del Software. Valladolid, Noviembre 2000.
- [Garzoto et al. 93]
F. Garzoto, D. Schwaie and P. Paolini HDM-A Model Based Approach to Hypermedia Application Design. ACM Transactions on Information System, 11 (1), Jan 1993, pp 1-26.
- [Gómez et al. 01]
J. Gómez, C. Cachero and O. Pastor. Extending a conceptual Modelling Approach to Web Application Design. First international workshop on Web-Oriented Software Technology. Valencia, 2001.
- [González et al. 01]
M. González, M. Mejías, M. J. Escalona, R. Martínez, J. A. Ortega. Interacción con los Usuarios en bibliotecas digitales. I Jornadas Dolmen. Sevilla, Junio 2001.
- [Hennicker/Koch 01]
R. Hennicker, N. Koch. A UML-based Methodology for hypermedia Design. First international workshop on Web-Oriented Software Technology. Valencia, 2001.
- [Izakowitz et al. 95]
T. Izakowitz, E. Stohr, P. Balasubramaniam: RMM:A methodology for structured hypermedia design. Comm. Of ACM, October 1995, pp.34-35
- [Jacobson 95]
I. Jacobson. Modelling with use cases-Formalizing use-case modelling. Journal of Object-Oriented Programming, June 1995
- [Jacobson et al. 99]
I. Jacobson, G. Booch, J. Rumbaugh. The Unified Software Development Process. Ed. Addison-Wesley, 1999.
- [Lange 95]
D. B. Lange. An Object-Oriented Design Approach for Developing Hipermedia Information Systems. Research Report RT00112, IBM Research, Tokyo Research Laboratory, Japan, 1995.
- [Lee et al. 98]
H. Lee, C. Lee, C. Yoo. A Scenario-based object-oriented methodology for developing hypermedia information systems. Processing of 31st Annual Conference on Systems Science. Eds. Sprague R.
- [Liddle et al. 01a]
S. W. Liddle, D. W. Embley, S. N. Woodfiel. A seamless model for Object-oriented systems development. First international workshop on Web-Oriented Software Technology. Valencia, Junio 2001.
- [Liddle et al. 01b]
S. W. Liddle, D. W. Embley, S. N. Woodfiel. An Active, Object-Oriented, Model-Equivalent Programming Language. First international workshop on Web-Oriented Software Technology. Valencia, Junio 2001.
- [Mandel et al. 00]
L. Mandel, A. Helmerich, L. A. Olsina, G. Rossi, M. Wirsing, N. Koch. Hyper-UML. Specification and modelling of multimedia an Hypermedia Applications in Distributed systems. Agosto 2000.
- [Mecca et al. 99]
G. Mecca, P. Atzeni, V. Crescenzi. The ARANEUS Guide to Web-Site Development. Technical Report, Universidad de Roma, 03 1999.
- [Nanard/Nanard 95]
J. Nanard, M. Nanard. HyperTEXT design environments and the hypertext design process. Communication of the ACM, August 1995. Vol 38(8), 49-56. 1995.
- [Olsina 98]
L. Olsina. Building a Web-based information system applying the hypermedia flexible process modelling strategy. 1st International workshop on Hypermedia Development, Hypertext 1998.
- [Pastor et al. 92]
O. Pastor, F. Hayes and S. Bear. OASIS: An object-Oriented Specification Language. CAiSE'92 International Conference, volume 593 of LNCS, pages 948-363. Springer-Verlag, 1992.
- [Pastoret al. 97]
O. Pastor, E. Insfran, V. Pelechano, J. Romero and J. Merseguer. OO-METHOD: An OO Software Production Environment Combining Conventional and Forma Methods. CAiSE'97. International Conference on Advanced Information Systems, 1997.
- [Rossi 96]
G. Rossi. An Object Oriented Method for Designing Hypermedia Applications. PHD Thesis, Departamento de Informática, PUC-Rio, Brazil, 1996.

[Rumbaugh 91]

J. Rumbaugh. Modelado y Diseño Orientado a Objetos. Ed. Prentice Hall, 1991.

[Schwabe/Rossi 98]

D. Schwabe, G. Rossi: An Object Oriented Approach to Web-Based Applications Design. TAPOS – Theory and Practice of Object Systems, vol. 4, 1998.

[Schwabe/Rossi 01]

D. Schwabe, G. Rossi: A Conference Review System with OOHDM. 1st International Workshop on Web-Oriented Software Technology. Valencia, 2001

[Suh/Lee 01]

W. Suh, and H. Lee: A Methodology for Building Content-oriented hypermedia systems The Journal of Systems and Software, Vol. 56, 2001, pp. 115–131

[Thomson et al. 98]

J. Thomson, J. Greer and J. Cooke. Algorithmically detectable design patterns for hypermedia collections. Workshop on Hypermedia development Process, Methods and Models. Hypermedia 1998.