웹정보시스템 개발 방법: 문헌 서베이

Web Information System Development Methodologies: A Survey of the Literature

최재화 (Jae Hwa Choi)   단국대학교 경영학부 교수

요약

웹정보시스템 개발은 적합한 개발방법을 정확하게 사용할 때만 성공할 수 있다. 웹정보시스템이 중요한 비중을 차지하게 되면서 웹정보시스템 개발자는 웹정보시스템 개발과정을 지원하는 방법론에 대한 바른 이해가 필요하다. 지금까지의 연구 문헌은 웹정보시스템 개발자에게 적절한 가이드라인을 제공하지 못하고 있다. 이 연구는 현재 웹정보시스템 분야에서 사용할 수 있는 웹정보시스템 개발을 위한 다양한 개발방법론과 기법들을 종합하여 체계적으로 요약하고 비교한다. 이 논문의 목표는 웹정보시스템 개발방법론과 기법에 대한 문헌을 비교하고, 실용적 가이드라인을 제시하고, 새로운 연구과제를 도출하는데 있다.

키워드 : 웹정보시스템, 정보시스템 개발방법론, 웹엔지니어링

I. Introduction

There is currently a rapid growth of the development of Web information systems (WIS). WIS has expanded its scope enormously and has moved to become a platform that can support all facets of organizational work (Isakowitz *et al.*, 1998; Ginige and Murugesan, 2001). Web applications are becoming increasingly business critical (Cartensen and Vogelsan, 2001). The rapid emergence of WIS in recent years presents a serious challenge for the WIS developer.

The increased development of WIS has produced, among others, a rising interest in the methodological approaches providing a suitable support for the construction of WIS. Many different approaches to WIS development have occurred in the short history of the field. Quite a number of books, journal papers and conference proceedings have produced a variety of methodologies and techniques for WIS development. Yet, the current effort of developing appropriate WIS methodology and relevant techniques has not led to the maturity of the field. Many practitioners in the field of Web engineering and software engineering have commented on the lack of suitable software engineering processes that can be used to build Web application

† The present research was conducted by the research fund of the Dankook University in 2005.

(Pressman, 1998, 2000; Reifer, 2000; Constantine and Lockwood, 2002; Escalona *et al.*, 2002). The field is not yet mature and WIS developers are still looking for better ways to follow.

The success of a WIS can be achieved only if it is developed in a proper and rigorous manner. As WIS has currently assumed a greater significance in organizations, WIS developers are often asked to find a good match between the life cycle development methodology and appropriate design techniques. That is, WIS developers have to consider carefully and choose correctly the appropriate methodology and techniques to base their development on. Thus, they need a better understanding of methodologies supporting the process of developing WIS. Existing literature provides little guidelines to the WIS developers. A critical review with helpful guidelines will aid in WIS developers' decision making with respect to the selection of WIS methodologies and techniques for their projects.

To provide an overview of the WIS development methodology field and a better understanding of the current methodologies and techniques, it is of critical importance to assess the state of the art and the practice of WIS development methodologies and techniques. This work provides a comprehensive review and a comparative study of the current approaches available in the WIS field that use different methodologies and techniques to develop WIS. The objective of this paper is to compare the WIS development methodologies and techniques in literature, provide practical guidelines, and generate new research issues.

In the next section, motivation and conceptual framework of this review are presented. Section 3 and 4 present the state of the art survey of WIS development methodologies and techniques in literature. Practical implications for WIS developers and research issues are drawn in section 5. The final section provides a conclusion and future work.

Ⅱ. Background and Review Framework

2.1 Background and Motivation

The development of an information system (IS) is a challenging and complex activity for almost every organization, requiring substantial investment in time and resources. In an effort to support the development of IS a large number of methodologies, techniques, and tools have been, and continue to be developed (Iivari *et al.*, 2001). Lately, WIS development methodologies are gaining popularity as the Web becomes the platform of most IS.

The Web has had a significant impact on the process of developing IS. Although some argue that the Web is no different to other new technologies and we could use the traditional information systems development (ISD) principles (Dennis, 1998; Pressman, 1998), it is in general agreement that WIS development process is different from the traditional ISD process because of certain characteristics of WIS (Genige and Murugesan, 2001; Carstensen and Vogelsan, 2001). WIS requires new methodologies, new tools, new people, and new working practices. The most commonly mentioned four characteristics for the introduction of a WIS development methodology are (Powel, 1998; Baskerville and Pries-Heje, 2001; McDonald and Welland, 2001; Holck, 2002):

- new, incremental development process: ongoing maintenance and evolution
- time pressure: Web-time (Internet-time) system development cycle

- interdisciplinary collaboration: new professions
- external focus: diverse, disloyal, remote user group

In recognition of these challenges and of the purported inadequacy of conventional ISD methodologies, there is much support in literature for the view that new, specialized methodologies are needed (Murugesan *et al.*, 2000; Carstensen and Vogelsan, 2001). During the last decade steady streams of research about various phases of WIS development have been introduced in literature. As the Web becomes the ideal platform for the development of WIS, several hypermedia/web application development methods were introduced (described in 3.1). At the same time a variety of user centered design (UCD) techniques were applied to the development of WIS (described in 3.2). To address the growing need for disciplined approaches to WIS development, Web Engineering was initiated as a new discipline and approached from a variety of established disciplines (described in 4.1). Some proposals of integrating traditional ISD methodologies with UCD techniques also appeared (described in 4.2).

Results of empirical research into the use of WIS development methodologies show the theory-reality gap. It is observed that practitioners are not using WIS development methodologies. Lang and Fitzgerald (2005) found that companies follow overwhelmingly home-grown approaches (95%) rather than specialized nonproprietary methods advocated by the WIS research community. UCD techniques in the literature are still underused and difficult to understand by WIS development teams and organizations or considered far too little and far too late in the WIS development process (Seefah and Metzker, 2004; Zhang *et al.*, 2005). Taylor *et al.* (2002) found that few UK organizations have formal framework/standards/best practice guides for assisting IS staff in web development projects.

These studies also show that developers are lacking in understanding of how to most effectively carry out WIS development. As organizations scramble to offer WISs, WISs were developed by ad hoc, hacker-type approaches, which lacked rigor, systematic techniques, quality assurance, and sound methodologies (Murugesan *et al.*, 2000). UCD is not used widely in practice since its structure and techniques are still relatively unknown, underused, difficult to master, and essentially inaccessible to common developers and small and medium-sized software development teams (Gould *et al.*, 1991; Seffah and Metzker, 2004). This underutilization of design methodology or technique is partly due to the difficulty of application by developers and to insufficient support of tools (Lowe and Webby, 1999; Barry and Lang, 2001).

In practice IS developers match methodology to application and choose techniques best suited to the application (Skyrme, 1997; Vessey and Glass, 1998; Murugesan and Ginige, 2005). In most cases, no "out of the box" methodology is perfectly suited to IS development. Recognizing that methodologies are collections of related techniques, Vessey and Glass(1998) suggest that IS professionals may use collections of appropriate techniques rather than unified methodologies. This approach allows IS professionals to use 'the best of practice' techniques which have been successful in solving organizations' problems.

Since no one methodology or technique alone is suitable for all situations, IS professionals need to know when to use a particular methodology and technique. Until now there has been no work to help IS professionals to understand different methodologies and techniques available and to decide which methodologies and techniques to be used for WIS development. This is partly due to the short history of the field and the rapid pace of change in web technology. This motivates us to provide a comprehensive survey of WIS methodologies and techniques. This review and comparison of methodologies and techniques will be useful to analyze the degree of suitability of these methodologies and techniques for developing a WIS. In addition, this work should help in the continuous process of improvement of the existing WIS methodologies and their tools, and generate new research issues in the field.

2.2 Review Framework

To review and compare different methodologies and techniques for the development of WIS, we first look at the notion of ISD methodology since the language used in the ISD field is sometimes confusing. Then we describe a review framework which allows us to capture a global picture of WIS methodologies and relevant techniques in literature. This framework is basically a classification scheme based on the scope and type of methodology and technique with respect to the WIS development life cycle.

2.2.1 ISD Framework

Iivari *et al.* (2001) propose a four-tiered ISD framework to provide a discourse. The framework includes paradigms, approaches, methodologies, and techniques. Paradigms represent the highest (fourth) level of abstraction. The authors suggest that there should only be a limited number of paradigms. The ISD approach is the next (third) level and inherits fundamental features from one or two dominant paradigms. The framework also defines an ISD approach as a set of related features that drive interpretations and actions in IS development. The ISD methodology is the next (second) level. It is defined as a codified set of goal-oriented procedures which are intended to guide the work and cooperation of various parties involved in the building of an IS application. Finally, the lowest (first) level represents techniques. A technique consists of a well-defined sequence of elementary operations that more or less guarantee the achievement of certain outcomes if executed correctly. <Table 1> shows four levels of the framework and their key components and examples.

〈Table 1〉 Four-tiered ISD Framework (Iivari *et al.*, 2001)

Level	Key Components	Example
ISD Paradigms	Ontology, Epistemology, Methodology, Ethics	Functionalism, Social Relativism, Neohumanism, Radical Structrualism
ISD Approaches	Goals, Guiding Principles, Fundamental Concepts, Principles of ISD Process	Structured, Information Modeling, Object-Oriented, Socio-Technical Design
ISD Methodologies	Relationships between Techniques, Detailed ISD Process	Structured Analysis and Design Technique, Modern Structured Analysis, Information Engineering, Object-Oriented Analysis and Design, Object Oriented Software Engineering
ISD Techniques	Detailed Concepts, Notations	Data Flow Diagram, Entity Relationship Diagram, State Transition Diagram, Object Diagram, Object Model, Use Case, Job Satisfaction Questionnaire

Besides providing the common terminology for IS development, this four level framework supports that methodologies are instantiations of existing ISD approaches. This allows IS developer to broaden their available methodology choices. Specific ISD methodologies are composites of specific techniques. Techniques have their detailed concepts and notations. Methodologies more or less successfully integrate these techniques into more coherent wholes. Because of the variability in the utility of constituent techniques, practitioners are not using methodologies as defined by their originators, but rather are tailoring them (Vessey and Glass, 1998; Fitzgerald, 1998; Fitzgerald and Russo, 2003). Different techniques may be mixed and matched on a single project. Also, portions of a technique may be used without using all aspects of that technique.

As the objective of this paper is to review current WIS methodologies and techniques, we consider the bottom two levels of Iivari *et al.*'s (2001) ISD framework. This allows us to survey a variety of the state of the art WIS methodologies and techniques in the literature. To describe different WIS methodologies in a comparative manner we look at them from the perspective of the standard ISD life cycle: planning, requirement analysis, design, implementation, and maintenance. Upon reviewing the literature of WIS methodologies, we use, the terms 'methodology and method' interchangeably as methodology and method now seem largely conflated concepts in IS parlance (Baskerville and Pries-Heje, 2001).

2.2.2 Review Framework

In literature different authors use different terminologies with regard to WIS. This makes the area rather confusing as Holck (2002) shows different names such as Web Information System, Web-based Information System (both often abbreviated as WIS), Web-Based System, Web-Based Application, Web Application, Web Software Application, Web Hypermedia Application, Interactive Web Application, Web Solution, Web System, and Web Site. In this paper we will use the terms "WIS (Web Information System)" and web application. A Web application is a small-scale WIS.

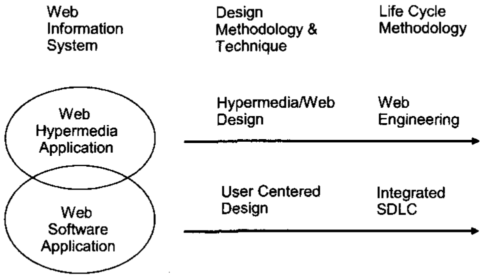
The Web is used as the delivery platform for roughly two types of applications: web hypermedia applications and web software applications (Christodoulou *et al.*, 2001). A web hypermedia application is a non-conventional application characterized by the authoring of information and its delivery over the web. A web software application represents conventional software applications that depend on the web or use the web's infrastructure for execution. WIS development has experienced the transition from a document-oriented approach to a software-oriented approach as the Web evolves from a document-centric platform towards an application-centric platform (Norton, 2001; Mayhew, 2003).

As the Web becomes the ideal platform for the development of web hypermedia applications, several *hypermedia/web design* methodologies were introduced (described in 3.1). At the same time the Human Computer Interaction (HCI) community applied a large variety of *user centered design (UCD)* techniques and usability engineering to make web applications usable (described in 3.2). WIS methodologies and techniques in these categories emphasize the involvement of experts in many different disciplines to satisfy very broad and 'remote' user group, which are major characteristics of WIS development methodology compared to the traditional ISD methodology.

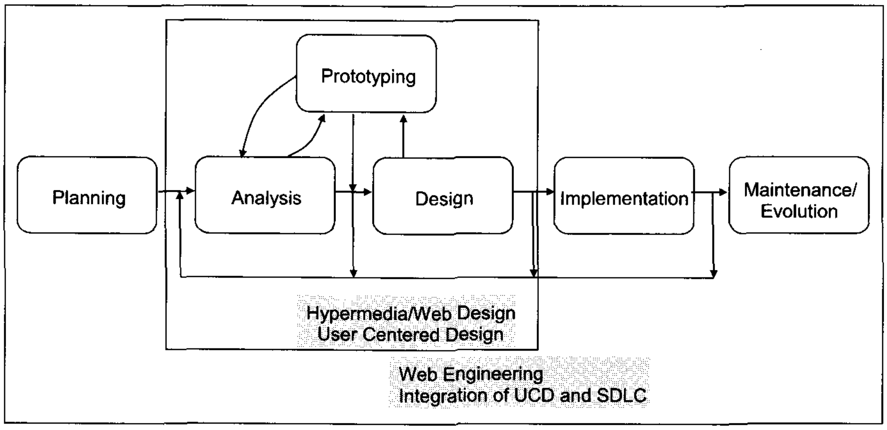
Although there is still no consensus on a general model of the lifecycle of WIS development, a scheme of typical activities involved in developing a WIS can be obtained by interpolating the life cycle model of the traditional ISD process. Figure 1 resembles a typical ISD life cycle, except the last phase is "maintenance/evolution". This is because in the WIS development life cycle "evolution" is required; Web applications are developed in an evolutionary process (Ramler *et al.*, 2004). Another characteristic of the WIS development life cycle is that the whole life cycle is characterized as 'internet time' or 'web time' and evolutionary (Baskerville and Pries-Heje, 2001, Holck, 2002).

Hypermedia/web design and UCD methods do not cover all phases of the ISD process. Instead they focus on the design phase and sometimes include the analysis phase. As WIS development projects become large and complex, the methodology needs to cover the entire life cycle of WIS development. One approach to this demand is the introduction of *Web engineering* as a discipline of IS development (described in 4.1). Another approach is the *integration of UCD with the system development life cycle (SDLC)*, the traditional ISD methodology (described in 4.2). Web engineering and Integration of UCD and SDLC approaches cover the entire life cycle of WIS development. This is shown in <Figure 1>.

The above perspectives allow us to develop a review framework in reviewing WIS methodologies and techniques in literature. WIS development methodologies and techniques are typically targeting either web hypermedia applications or web software applications from the perspective of the type of web applications on which they focus. WIS development methodologies and techniques can also be categorized as either the design methodology and technique or life cycle methodology from the perspective of the ISD life cycle process. As shown in <Figure 2> these two dimensions generate four classes of WIS development methodologies in literature.



⟨Figure 2⟩ WIS Development Methodologies



⟨Figure 1⟩ WIS Development Lifecycle

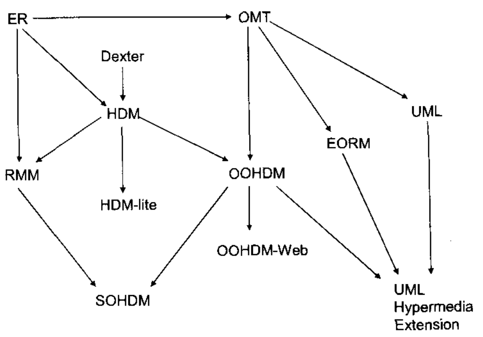
In the following, using this review framework we present a palette of WIS development methods and techniques which could aid WIS developers in their work. Not all of the methodologies are exhaustively presented. And we look at the most significant aspects of each of them in order to have a better understanding of the different methods and techniques. It is noted that comparing phases of different methodologies and techniques in this way hides other important aspects. Sometimes the depth with which a methodology describes a phase and the guidelines given for that phase varies significantly. Some methodology descriptions have only a couple of textual paragraphs while others provide even tools to support the phase. Section 3 compares different Hypermedia/Web Design methods and UCD techniques by putting them in the analysis and design process of WIS development. Section 4 looks at two groups of the life cycle approaches of WIS development: Web Engineering and Integration of UCD/HCI and SDLC.

Ⅲ. WIS Design Methodology and Techniques

3.1 Hypermedia/Web Design

Quite a number of hypermedia/web application design methodologies were introduced in the last decade. Hypermedia Design Model (HDM, Garzotto *et al.*, 1993) is not a methodology for the development of a hypermedia application, rather a modeling technique. Yet, HDM is considered as one of the first methods to define the structure and navigation characteristics of hypermedia applications. Relationship Management Methodology (RMM, Isakowitz *et al.*, 1995) is a methodology which is based on the Entity-Relationship (ER) model and HDM. All other methodologies adopt Object-Oriented (OO) model which is better suited to WIS. Object Oriented Hypermedia Design Model (OOHDM, Schwabe and Rossi, 1995) and Enhanced Object Relationship Methodology (EORM, Lange, 1996) use the OO model to take advantage of the paradigm. Web Site Design Method (WSDM, De Troyer and Leune, 1997) is a methodology for the kiosk web site and focused on the user-centered design rather than data-centered design. Scenario-based Object Oriented Hypermedia Design Methodology (SOHDM, Lee *et al.*, 1999) proposes a scenario-based design as well as requirements gathering.

With the growth of the Web and the development of sophisticated hypermedia systems, the development of hypermedia/web applications has become more complex. WebML (Ceri *et al.*, 2002, Ceri *et al.*, 2003) is a conceptual web modeling language that uses the ER model. Recently, as Unified Modeling Language (UML) is considered the standard de-facto in the design practice, many web design methodologies extending UML are proposed. They include UWE (UML-based Web Engineering, Hennicker and Koch, 2000; Koch, 2001), W2000 (Baresi *et al.*, 2001), and OO-H (Object-Oriented Hypermedia, Gomez *et al.*, 2001).



〈Figure 3〉 Evolution of Hypermedia Development Methods (Lang, 2002)

<Figure 3> shows the evolution path of hypermedia/web design methodologies. It shows that the conceptual model of all methodologies is based on either ER or OO model. The Dexter Hypertext Reference Model (Halasz and Schwartz, 1994) is the basis for some hypermedia design models. HDM-lite (Fraternali and Paolini, 2000) and OOHDM-Web (Schwabe and Pontes, 1998) are the extension of HDM and OOHDM for web applications.

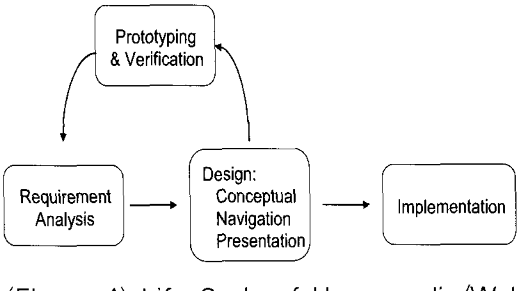
Most of the hypermedia/web application design methodologies do not cover a full life-cycle of the WIS development. As <Table 2> shows all methodologies focus on the design stage of the hypermedia/web application development life cycle.

〈Table 2〉 Hypermedia/Web Design Methodologies

Model	Development Phase			Tool
	Analysis	Design	Implementation	
HDM		1. Authoring-in-the-large 2. Authoring-in-the-small		
RMM		1. ER design 2. Slice design 3. Navigational design 4. Conversion protocol design 5. UI screen design 6. Run-time behavior design	7. Construction and testing	RMCase
EORM		1. Class framework 2. Composition framework 3. CUI framework		ONTOS Studio
OOHDM		1. Conceptual design 2. Navigational design 3. Abstract UI design	4. Implementation	OOHDM-Web
SOHDM	1. Domain analysis	2. OO modeling 3. View design 4. Navigational design 5. Implementation design	6. Construction	
WSDM		1. User modeling 2. Conceptual design 3. Implementation design	4. Implementation	
WebML	1. Requirements specification	2. Data design 3. Hypertext design		WebRatio
UWE	1. Requirements analysis	2. Conceptual design 3. Navigation design 4. presentation design		ArgoUWE tool
W2000	1. Requirements analysis	2. Hypermedia design 3. Functional design 4. Visibility design		
OO-H		1. Class diagram 2. Navigation access diagram 3. Abstract presentation diagram		VisualWADE

Other phases like planning, implementation, and maintenance are handled with less relevance or not included at all. However, all emphasize the need for an incremental and interactive development process. Although most methodologies are model-driven, there are several differences such as the level of coverage of the design process and the level of support provided at different stages. <Table 2> compares the steps to be performed when using a methodology and CASE tools supporting it. In the table the steps of each methodology are numbered.

<Table 2> shows that different methodologies follow various kinds and numbers of steps. Yet, hypermedia/web design methodologies share at a higher level the similarities in the systematic design of web applications. Most of these methodologies make a clear separation of the domain problem analysis and design from the design of the navigation and the presentation. First the domain model of an application is analyzed and designed. Then the methods focus on the design of navigation and presentation. Thus, the life cycle of a generic hypermedia/web application design methodology looks like <Figure 4>. It is noted that some of the design activities, for example conceptual design, may be considered as analysis activities in a traditional ISD process. This is because we followed the hypermedia/web design methodology literature.


Prototyping & Verification
Requirement Analysis
Design: Conceptual Navigation Presentation
Implementation

〈Figure 4〉 Life Cycle of Hypermedia/Web Design

3.1.1 Requirement Analysis Phase

Koch (2001) classifies requirements into two groups: functional and non-functional. Baresi *et al.* (2001) divides the requirements analysis into functional requirements analyses and navigational requirements analyses. Functional requirements include content, structure, presentation, adaptation, and user model. Similarly, Escalona and Koch (2004) classify six types of requirements: data, user interface, navigation, adaptive, transactional, and non-functional. They divide requirement engineering activities into capture, definition and validation. The requirement analysis phase is a phase that few hypermedia/web design methodologies handle in detail. Even if they do, they focus on the requirements capture and definition activities at a primitive level. Many methodologies handle requirements capture as part of the definition activity.

<Table 3> shows what requirement engineering techniques are used in different hypermedia/web design methodologies. The most popular requirements engineering techniques are scenario and use cases. WSDM introduces user modeling in web site design to identify the different users' roles by performing users' classification and users' group description. SOHDM was the first to emphasize the importance of the analysis of capturing and defining application requirements. The original OOHDM was augmented to capture and define requirements using the user interaction diagram (UID), which is based on use cases in the OO design (Vilain *et al.*, 2000). WebML proposes the use of interviewing and analysis of document analysis to capture requirements and the use case specification with semi-structured textual description to specify requirements (Ceri *et al.*, 2003). UWE proposes walk-though, audits and prototypes for validation (Koch, 2001). In most UML extension methodologies use case is com-

〈Table 3〉 Requirements Engineering Techniques in Hypermedia/Web Design Methodologies (partially adapted from Escalona and Koch, 2004)

Model/ Technique	Capture						Definition							Validation			
	I	CM	UC	Q	ST	O	NL	G	T	SC	UC	P	O	R	A	P	O
OOHDM			0								0		UID				
SOHDM						Data Flow Diagram				0			Event List				
WSDM	0	0					0										
WebML	0				0	Document Analysis	0		0		0						Acceptance Test
UWE		0		0				0		0	0			0	0	0	
W2000											0						
OO-H											0						

Note) Techniques: I: Interview, CM: Concept Map, UC: Use Cases, Q: Questionnaire, ST: Storyboard, NL: Natural Language, G: Glossary, T: Template, SC: Scenario, P: Prototype, R: Review, A: Audit, O: Other

plemented with more concrete models, such as UID, template, or UML activity diagram. For example, W2000 represents functional and navigational requirements with UML use case diagrams.

3.1.2 Design Phase

The design phase is the most covered phase in all hypermedia/web design methodologies. Typically design is performed at three different dimensions (Hennicker and Koch, 2000; Selemi *et al.*, 2005):

- Conceptual - define the information organization handled by the application and its relationship
- Navigation - represent how information can be accessed
- Presentation - describe how information and access can be presented to the user

Other researchers use similar classification for web applications: content, hypertext, and presentation (Retschitzegger and Schwinger, 2000; Kappel *et al.*, 2004), structural, navigation, and presentation (Fraternali and Paolini, 1998), conceptual, navigational, and abstract interface (Schwabe *et al.*, 1996).

3.1.3 Conceptual design

At the conceptual level, the information domain is modeled using either ER or OO. HDM and RMM use the ER model. EORM and OOHDM use the OO model. OOHDM develops the conceptual model with a class diagram and allows model users into the conceptual schema like other types of objects. In SOHDM scenarios are defined in the domain analysis phase and are used for object modeling. In WSDM, object modeling consists of business object modeling, user object modeling, and perspective object modeling.

UML extension methodologies follow OO modeling with UML. Conceptual models are represented by UML class diagrams. OO-H methodology captures a set of views for both structure (statics) and behavior (dynamics) of the system and develops domain model and user model which are represented as UML compliant class diagrams. UWE builds conceptual models with use cases and presents them as UML class diagrams. W2000 performs hyperbase structural design to structure the core information and access structure design to organize them. Both designs are represented as UML class diagrams.

3.1.4 Navigational design

Movement in the navigation space is typically defined by the navigational semantics in terms of nodes and links independently of the conceptual model. All methodologies introduce node (pages, navigation units, content units, slices or cards), different types of links (perspective, structural, application link, etc.), and access structure (indices, guided tours, etc.) concepts as the compositional and navigational structure of the hypermedia/web applications. HDM and RMM adapt ER model to represent navigational characteristics such as nodes, contents, anchors and links. OOHDM builds different navigation views (navigation context) for the same OO conceptual model. In such navigation views anchors are defined. SOHDM uses scenarios to determine access structure nodes. WebML allows organizations of contents of a node in self-contained regions of the screen (page) and to specify regions to be displayed together, or separately. WebML also includes the notion of group and individual users which are modeled as a special entity allowing personalization.

UML extension methodologies represent navigation through a class diagram. UWE uses UML and a set of stereotypes to define the modeling of navigation aspects. Navigation aids are represented by stereo typed classes. OO-H specifies the view of each user of the system and is represented by the navigational access diagram. OO-H allows modeling of dynamic views, as well as static views. OO-H requires design of as many navigational access diagrams as user types when personalization is needed.

3.1.5 Presentation design

The navigation units (node units) are mapped to conceptual units (entities or classes) to display the information or data at presentation time. RMM uses slices for different views of the information. OOHDM uses abstract data views (ADV) to describe the objects perceived by the user. OOHDM uses ADV to build a specific view for each user for personalization purposes. In WebML interactive behavior is supported by 'operation units' which are linked to other units and can be predefined or built. UWE constructs a presentation model which consists of a static model of an abstract user interface based on frameset and a model of the dynamic aspects of the presentation. UWE uses task models and statecharts of web scenarios to model the dynamic aspects of the application. OO-H methodology develops abstract presentation diagrams and includes the services of each class that will be used in their navigational diagrams.

To support personalization user modeling is employed in a few current methodologies. The user model determines user roles and features, which are important for adaptation and personalization, and includes various characteristics of the user's goals, tasks, interests, knowledge, background, hyperspace experience, and preferences. WSDM provides a language to help the designer to define adaptive behavior. WebML provides a framework for one-to-one personalization and even the support for interface proactive behavior. UWE defines adaptation rules as Object Constraint Language (OCL) expressions. OO-H provides a basic level of personalization, which is based on departing from user profiles and designing a different model for each user type.

<Table 4> shows the design concepts of some of the methodologies at the three typical levels of

〈Table 4〉 Design Constructs of Hypermedia/Web Design Methodologies

	Conceptual Design	Navigation Design	Presentation Design
HDM	Entity, Relationship	Link, Node	Slot, Frame
RMM	Entity, Relationship	Link, Slice, Access Primitive	Slice
EORM	Class, OO Relationship	Link	
OOHDM	Class, OO Relationship	Link, Navigation Class, Navigation Context	Abstract Data View
SOHDM	Scenario, Activity Flow	Navigation Link, OO-View, Access Structure Nodes	UI component
WSDM	Object, Relationship	Link, Navigation Track	
WebML	Entity, Relationship	Link, Unit	
UWE	Class, Association	Navigation class, Direct Navigability	Anchor, Form
W2000	Class Diagram	Link, Node	
OO-H	Class Diagram	Link	

hypermedia/web application design: conceptual, navigation and presentation level.

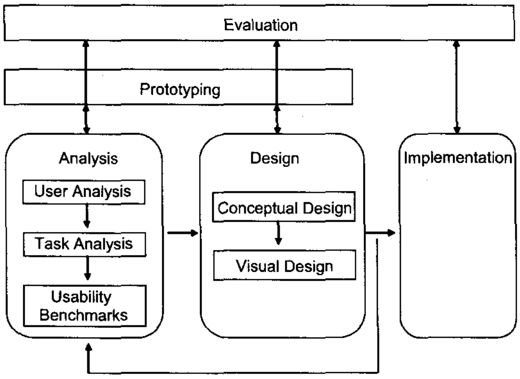
3.2 User Centered Design

Mayhew (2003) maintains that the Web is a software, really just a different platform, and the field of software usability engineering (UE) can be directly applied to Web design and development. Usability and the user experience are emerging as critical determinants of success in web applications (Constantine and Lockwood, 2002). UCD is frequently prescribed and adopted as the key to user-friendly web design (Mao *et al.*, 2005).

In a broad sense UCD is a philosophy and a process (Knobel, 2002). It is a philosophy which is based on two fundamental premises: (1) design should focus on users and (2) a scientific or engineering approach to design must be taken. UCD is a process that incorporates a set of techniques used by the designers to design a “usable” system. UCD is participatory, iterative and multi-disciplinary. In a narrow sense UCD is a loose collection of human-factor techniques united under a philosophy of understanding users and involving them in design (Constantine and Lockwood, 2002).

To define how UCD techniques are applied during the WIS development process, several UE life cycle approaches have been proposed. Mayhew (1999) proposes a UE life cycle which documents a structured and systematic approach to addressing usability within the product development process. The overall life cycle consists of three phases: requirements analysis, design/testing/development, and installation. Ferre *et al.* (2001) developed a generic usability process from different approaches to usability. They summarize different UE approaches into two phases: usability analysis phase and usability design phase. Jokela (2003) proposes that a technique independent model of UE which is analogous to the life cycle models used in software engineering. The process model suggests a sequence of UE actions: user group identification, analysis of context of use, user task redesign, usability requirements determination, interaction design, and usability evaluation. Constantine and Lockwood (2002) propose the usage-centered engineering for web application which is a model-driven approach to software engineering, focusing on user interface design and usability.

The usability engineering approach in the proposals mentioned above essentially focuses on the analysis phase and design phase. A generic UE process on which we will map different UCD techniques can be represented as <Figure 5>, which is similar to the one in Mayhew (1999), and Ferre *et al.* (2001), and Ferre and Morento (2003). It is again noted that some of the design activities in UCD, for example conceptual design, may be considered as analysis activities in a traditional ISD life cycle. This is because we followed the UE literature.



<Figure 5> UCD Process

The UCD field has numerous techniques to be used for different project characteristics and for different usability purposes (Seffah and Metzker, 2004). UCDs are point techniques for addressing a particular need during product development (Notess and Blevis, 2004). Even articulations of HCI design processes found in Preece *et al.* (2002) merely lay out broad stages of design activity and suggest a broad range of HCI techniques that may be used within each stage. The key to the general applicability and flexibility of the UE lies in the choice of which techniques to apply to each task, not in the choice of which tasks to carry out (Mayhew, 2003). The approach to any given project can be adapted by a careful selection of techniques based on project constraints.

In the following we look at UCD techniques which are more or less suitable at different phases in the WIS development. We have chosen the ones more valuable for a broad variety of systems, considering specifically their applicability from a WIS developer point of view.

3.2.1 Analysis Techniques

Constantine and Lockwood (2002) point out that web projects often suffer from vague objectives and unrealistic ambitions. At the start of a WIS development project, system scope definition activity can be performed using techniques such as needs analysis (Hix and Hartson, 1993) and product vision (Anderson *et al.*, 2001). A product vision is a model envisioning the final products' strategy for meeting the requirements. The product vision provides the structural blueprint for the product and how the end user will interact with and navigate through the system. Anderson *et al.* (2001) view that this model, a roadmap, is a key component of the UCD process.

3.2.2 User analysis

An abstract representation of users is often used to guide product design. Contextual inquiry (Beyer and Holtzblatt, 1997) provides an effective, detailed, and structured requirements gathering technique. Interview and questionnaire (Hackos and Redish, 1998) are ways of collecting data related to the needs of and expectations of users. A good reference for user surveys is Dillman (1999). A focus group (Hackos and Redish, 1998) is an organized discussion with a selected group of users.

Personas, user profiles, and scenarios can be derived from contextual analysis and use classes. A persona is a hypothetical user with a name and a face, carefully described in terms of needs, goals and tasks for guiding decisions about visual design, functionality, navigation, and content (Cooper, 1999; Pruitt and Grudin, 2003). Personas are developed through research and interviews with client team members and actual users (e.g., customers, prospects, etc.). Personas complement other approaches or are used where another approach is impractical. For example, a context-driven, rhetorical role-playing approach can be used by combining the persona theory and situated action framework (Loeber and Cristea, 2003).

User role models or user profiles (Constantine and Lockwood, 2002) are good techniques in understanding and interpreting the project requirements. Sawasdichal and Poggenpohl (2002) propose a user-centered approach to information design on websites which considers users' purposes, users' modes of searching and their search behaviors. Using existing marketing models Albert *et al.* (2004) propose a model for design and management of content and interactivity of customer-centric web sites. Their model, called GIST (Gather-Infer-Segment-Track), analyzes users' needs and motives to help design and redesign the websites continuously. To analyze the navigational behavior of the website visitor and user characteristics, Loeber and Cristea (2003) combine information seeking processes inside a motivation, ability and opportunity (MAO) model by Hoyer and McInnis (2001).

3.2.3 Task analysis

User analysis is taken as input for task analysis (Hackos and Redish, 1998), and both are sometimes performed jointly using the same analysis techniques such as personas and contextual inquiry. Field study is a way of collecting information concerning the environment in which the artifact will be used. Rapid ethnography is an observational technique that involves going to the prospective users of a system and observing the activities they perform, their interactions and their work environment (Salvendy, 1997). Alt *et al.* (2002) uses a technique called customer process analysis to analyze and model the customer processes, the required services, and the existing IS to design a portal. Contextual analysis (Beyer and Holtzblatt, 1997) involves observing and interviewing users in their workplace to identify the task requirements, system specifications, environmental factors, and both direct and indirect user characteristics that must be supported by the system.

Scenario (McGraw and Harbison, 1997; Caroll, 2000) and storyboarding (Hackos and Redish, 1998) can serve as effective shared artifacts between the users and designers. In the object-oriented software engineering, use cases are often used to capture and document external requirements of the system. Use case comprises a single case of use of a system that is complete, well-defined, and meaningful from the perspective of an external user (Constantine and Lockwood, 1999). They can also be used for prototyping and requirements validation. Affinity diagram is often used to develop logical categories based on user-perceived relationships and conceptual framework (Beyer and Holtzblatt, 1998).

3.2.4 Usability benchmarks

Usability benchmarks are set as quantitative usability goals, which are defined before system design begins. Usability benchmarks are established by defining a set of benchmarks for each usability attribute to evaluate. To define minimal acceptable user performance and satisfaction criteria based on

〈Table 5〉 UCD Analysis Techniques

	Technique	Source
	Productive Vision	Anderson *et al.* (2001)
	Needs Analysis	Hix and Hartson (1993)
User Analysis	Focus group	Hackos and Redish (1998)
	Contextual inquiry	Beyer and Holtzblatt (1998)
	Personas	Cooper (1999)
Task Analysis	Contextual inquiry	Beyer and Holtzblatt (1998)
	Personas	Cooper (1999)
	Affinity diagram	Beyer and Holtzblatt (1998)
	Storyboarding	Hackos and Redish (1998)
	Scenario writing	Carroll (2000)
	Paper Prototypes	Nielsen (1993)
	Use Case	Constantine and Lockwood (1999)
	Task analysis	Hackos and Redish (1998), Preece *et al.* (1994)
Usability Benchmark	Usability Specification	Hix and Hartson (1993)
	Benchmark task	Hix and Hartson (1993), Preece *et al.* (1994)

a subset of high-priority qualitative goals, specific, qualitative goals reflecting usability requirements are also extracted from the user profile and contextual task analysis (Hix and Hartson, 1993). Role playing, walkthrough, streamlined cognitive walkthrough (Spencer, 2000), and simulations are used to evaluate information about user needs and expectations, prototype evaluation.

3.3 Design Techniques

3.3.1 Conceptual design

The findings of the user and task analysis are the basis for the conceptual design. Every task (or use case) identified in the task analysis is specified in detail in order to design precisely the interaction between the user and the system. Techniques described above for task analysis can apply. Holistic design (Preece *et al.*, 1994) can let development team to define an innovative system.

Contextual design (Beyer and Holtzblatt, 1997) is an approach to designing products directly from an understanding of how the customer works and provides a systematic method whereby interdisciplinary design teams can use data gathered through field observations to arrive at a shared understanding of work process, participants, and environments. Various models capture this understanding and drive a work redesign and validation process, resulting in an implementable system design. Contextual design is adapted to the agile software development method with the focus of customer and called as Rapid Contextual Design (Beyer *et al.*, 2004). In the participatory design the users of the system become co-designers. Participatory design (Schuler and Namioka, 1993) is usually used within a mini project to generate prototypes that feed into an overall project's design process. To better understand the users' concept of how the information on the web site should be organized, card sorting is used as an input to design (Fuccella, 1997).

UML is often used to represent the structure of the detailed tasks. In the UML, one of the key tools for behavior modeling is the use case construct, originating from OO software engineering. Use Cases, related to scenarios, are used to model tasks by capturing the system's functional requirements from the users' perspective (Constantine and Lockwood, 1999). Unlike a single use scenario, use case is used in multiple capacities as a thread to tie together activities throughout projects. Use case map (Constantine and Lockwood, 1999) captures functional requirements in terms of causal scenarios bound to underlying abstract components. State-transition diagrams (Hix and Hartson, 1993) are used to describe the behavior of individual objects over the full set of use cases that affect those objects.

3.3.2 Visual design

During the visual design or interaction design (Preece *et al.*, 2002) phase, the user interface (UI)'s appearance is designed. Interaction conceptual design is a genuine creative activity which can be helped by valuable design guidelines (Constantine and Lockwood, 1999; Nielsen, 1993; Shneiderman, 1998). During the conceptual design the basic user interaction and the objects in the UI and the contexts in which interaction takes place are defined using techniques such as the context navigation map (Constantine and Lockwood, 1999).

Detailed design of the complete product user interface is carried out based on the refined and validated conceptual model and screen design standards. Design tips for Visual Interaction Design can be found in (Shneiderman, 1998). The deliverables from this phase are typically paper prototypes (Beyer and Holtzblatt, 1998), such as pencil drawings or screen mockups (Constantine and Lockwood, 1999), and a specification, which describes the UI's behavior.

〈Table 6〉 UCD Design Techniques

Phase	Techniques	Source
Conceptual Design	Holistic Design	Preece *et al.* (1994)
	Contextual Design	Beyer and Holtzblatt (1998)
	Participatory Design	Schuler and Namioka (1993)
	Card Sorting	Fuccella (1997)
	State Transition Diagram	Hix and Hartson (1993)
	Use Case Map	Constantine and Lockwood (1999)
Visual Design	Context Navigation Map	Constantine and Lockwood (1999)
	Software Mock-up	Constantine and Lockwood (1999)
	Guidelines for Visual Interaction Design	Hix and Hartson (1993), Shneiderman (1998), Hackos and Redish (1998)
Evaluation	Formulative Evaluation	Hix and Hartson(1993)
	Heuristic Evaluation	Nielsen (1993), Shneiderman (1998)
	Collaborative Usability Inspection	Constantine and Lockwood (1999)
	User-Performance Data Logging	Nielsen (1993)
	User Interface Satisfaction Questionnaire	Shneiderman (1998)
	Usability Testing	Hackos and Redish (1998), Dumas and Redish (1999)

3.3.3 Prototyping and Evaluation

Prototypes are valuable for performing usability testing in early development phases. Prototyping techniques include paper mockups, scenarios, storyboards, and snapshots. Role playing, walkthrough, and simulations are techniques for the evaluation of alternative designs. Interviews and questionnaires are ways of evaluating design alternatives, prototypes and the final artifact. Usability evaluation can be performed by paper prototypes (or paper mockups) (Nielsen, 1993), thinking aloud (Nielsen, 1992), formative evaluation (Hix and Hartson, 1993), heuristic evaluation (Nielsen, 1993), collaborative usability inspection (Constantine and Lockwood, 1999), or informal usability testing with users. Usability testing focuses on user needs and collects quantitative data related to measurable usability criteria.

Ⅳ. WIS SDLC Methodology

4.1 Web Engineering

To address the growing need for disciplined approaches and new methodologies for the development, deployment, and evaluation of WIS, academics argued for the establishment of Web engineering as a discipline in its own right (Murugesan *et al.*, 2001). The field of Web engineering was widely heralded in workshops and research papers in the late 1990s and early 2000s. Now we have web engineering conferences, journals, and books. Web Engineering is both a pro-active approach and a growing collection of theoretical and empirical research in Web application development (Deshpande *et al.*, 2002). Web engineering is compelled to become a multidisciplinary field as it encompasses diverse disciplines (Ginige and Murugesan, 2001).

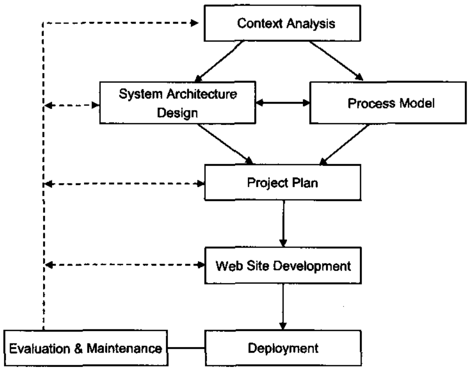
Web engineering is defined as the establishment and use of sound scientific, engineering, and management principles and disciplined and systematic approaches to the successful development, deployment and maintenance of high quality Web-based systems and applications (Murugesan *et al.*, 2001). Web engineering is basically a life cycle process model and encompasses all phases of WIS development process including the analysis, design, implementation, and performance evaluation and continual maintenance stages (Deshpande and Hansen 2001). Web engineering borrows many of software engineering's fundamental concepts and principles, emphasizing the same technical and management activities (Seffah and Metzker, 2004; Pressman, 2004).

A number of researchers proposed general Web Engineering methodologies so that WIS developers can design appropriate process models to suit different types of development and development issues, based on the type of application. In general, Web engineering methodology is a loose fitting framework for developing WIS. This allows the organization to adapt the methodology to the specific situation of the project. Pressman (2004) maintains that web engineers will choose a process model based on the attributes of web applications. If immediacy and continuous evolution are primary attributes of a web application, an agile process model will produce web application releases in rapid-fire sequence. If a web application is to be developed over a longer time period, an incremental process model might be chosen. <Table 7> shows some of web engineering methodologies.

Murugesan and Ginige (2001) propose a web engineering process shown in <Figure 6>. Among the steps to be performed in <Figure 6>, the process model leaves the development team to design appropriate process models. A process model specifies a set of sub-processes that need to be carried out to develop and implement the overall system. To carry out some of these sub-processes, the development team can adopt various development models such as the waterfall model, the spiral model or the rapid prototyping model from the software engineering discipline. Murugesan and Ginige (2005) recommend an evolutionary process for Web development as the structure and functionality will change constantly over time.

〈Table 7〉 Web Engineering Methodologies

Source	Development View
Ginige and Murugesan (2001)	Evolutionary
Pressman (2004)	Flexible, Evolutionary
Norton (2001)	Cross-Functional, Evolutionary
Standing (2000)	Evolutionary
Vidgen *et al.* (2002)	Soft Methodology

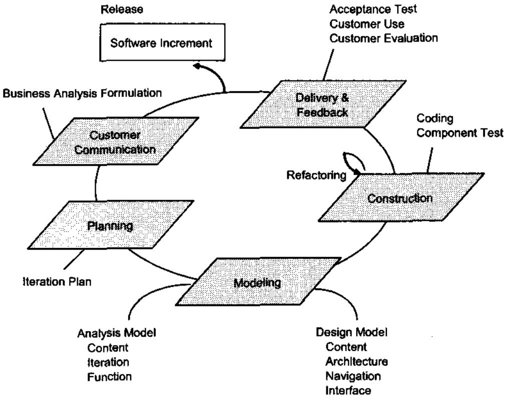


〈Figure 6〉 Web Engineering Process (Murugesan and Ginige, 2001)

Vidgen *et al.* (2002) propose the WIS development methodology (WISDM). WISDM uses the multiview framework as the basis for constructing a situation-specific methodology. The methodology balances the engineering approach and sociotechnical approach. Although WISDM is put under the IS development life cycle perspective, WISDM covers the analysis and design activities of system development. WISDM focuses on the information system definition and leaves the rest of IS development life cycle out of the scope.

Web Engineering is intentionally made to be generic, adaptable and easy for the developers to follow in order to transfer good practices and research results of Web engineering to real WIS projects. This is especially so in the methodology of Ginige and Murugesan (2001) and Vidgen *et al.* (2002).

Unlike Murugesan and Ginige (2001), Pressman (2004) proposes a specific web engineering methodology based on the characteristics of WIS development. It assumes that (1) web applications are often delivered incrementally, (2) changes will occur frequently, and (3) timelines are short. Pressman's web engineering process is based on the agile process model. It shows a web engineering framework in which web engineering activities are applied using an incremental process flow. One can consider both customer communication and planning activities as the planning phase of the traditional life cycle process.



〈Figure 7〉 Web Engineering Process (Pressman, 2004)

Norton (2001) introduces a methodology from established industry best practices. The methodology is a mix of rapid application development, evolutionary delivery and joint application development adapted to the unique climate of the Web. It focuses on the requirements definition and early stages of Web projects. The collaborative product definition consists of five activities: assembling preliminary requirements, determining participants, holding feature workshop session (JAD), prototyping, performing feature document drafting and review, and developing product design. This methodology emphasizes cross-functional feature teams, collaborative product conception and design and evolutionary delivery.

Standing (2000) proposes a framework, called Internet Commerce Development Methodology (ICDM), in which a development team can adapt the methodology to the specialized conditions of the organization. ICDM is a business analysis methodology, as well as a system development methodology. ICDM encompasses business strategy, the development of Internet systems, and the appropriate management structures. ICDM has a number of component phases to guide the development of strategy and steer the processes related to Web site development. The component development phases follow strategy, analysis, design, implementation, and evolution.

4.2 Integration of UCD/HCI and SDLC

As described in section 3.2 several authors put UCD techniques into the perspective of the ISD process and called it "usability engineering (UE)". However, the UE proposals focus on the interaction design of a system as they were developed from the usability perspective. Specifically UE focuses on requirements and evaluations (Gulliksen and Boivie, 2001). Thus, it does not cover the entire life cycle of ISD process. The rise of interactive systems with substantial functionality like large scale WISs mandates the integration of UCD techniques and software engineering process. In this section we look at some works proposing the integration of UCD/ HCI and the traditional SDLC methodology.

In the development of organizational IS, the modern SDLC methodologies focus on system functionalities and data requirements to meet organizational needs. SDLC methodologies have traditionally constructed software systems with the development focus on internals, on processing logic and data organization. On the other hand UCD has focused on the users and the way they interact with the system, in particular requirements and evaluations. HCI field is multidisciplinary and its foundation comes from the disciplines of psychology, sociology, industrial design, graphic design, and so forth. HCI field speaks a different language and it deals with software development using a different perspective. Often, UCD techniques are regarded as being somehow decoupled from the mainstream SDLC (Seffah and Metzker, 2004). The two disciplines are typically separate and are applied independently with little coordination.

Integration of UCD/HCI and SDLC methodology is suggested in order that the accumulated expertise of both communities can be effectively deployed for the benefit of the end users (Faulkner and Culwin, 2000; Granollers, 2003). An increasing number of software development organizations are pursuing the goal of integrating usability practices into their software engineering processes (Ferre and Moreno, 2003). The UCD and SDLC methodologies together help develop "usable" and "functionally satisfactory" software systems. Recognizing

〈Table 8〉 Integration of UCD and SDLC

Source	SDLC Methodology	Development View
Ferre (2001)	Use-case method (OO)	Joint development
Pyla *et al.* (2003)	Generic SE process	Parallel development with coordination
Sousa and Furtado (2003)	Rational Unified Process (RUP)	Joint development
Zhang *et al.* (2005)	Systems Analysis & Design	Joint development

the necessity of a methodology that integrates HCI concepts and techniques with the SDLC methodology, a number of proposals have been introduced as shown in <Table 8>.

These technique independent integration approaches help large complex WIS development. They take either a joint or parallel development view. Ferre (2001), Sousa and Furtado (2003), and Zhang *et al.* (2005) propose a joint development methodology. This approach integrates UCD and SDLC under one common framework and builds a common overall design representation, shared by the two developer roles and processes. UCD techniques reviewed in 3.2 will be used in these integrated methodologies. That is, WIS development team can use these proposals to decide where the UCD techniques may fit in the development process. On the other hand Pyla *et al.* (2003) take a parallel development approach. This approach retains the two development processes as separately identifiable processes, each with its own life cycle structures, development activities, and techniques. Anderson *et al.* (2001) and Radle and Young (2001) present ad-hoc solutions which have been created for particular software development organizations. As these two solutions lack a generic approach to be applied to organizations with different characteristics, they are not included in <Table 8>.

4.2.1 Integration by Joint Development

Ferre (2001) proposes an object-oriented development process (OO-HCI) in which usability techniques are embedded where appropriate. Recognizing that use-case driven development is a UCD technique, the proposal integrates usability techniques with the use-case driven method. Some UCD techniques are chosen to be embedded in the proposal for a joint development process.

Sousa and Furtado (2003) integrate HCI in SE. They adapt Rational Unified Process (RUP) to integrate HCI aspects into its main workflows and generate the RUP for interactive systems. As UML extension is often used to develop WIS, this approach integrates HCI activities into RUP workflow through workers and artifacts.

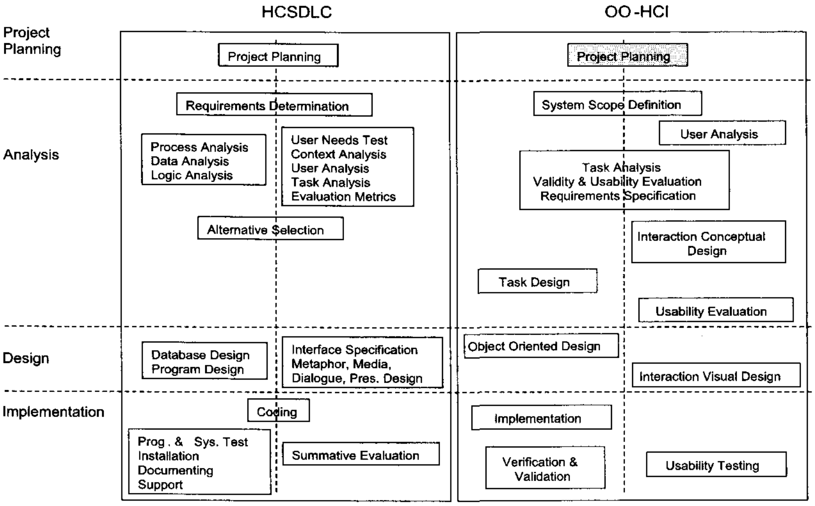
Zhang *et al.* (2005) maintain that incorporating an HCI perspective into the SDLC is critical to IS success and in turn to the success of organizations and businesses. The integrated methodology, human-centered systems development methodology (HCSDLC), provides a step-by-step procedure for transformations between activities at different stages. The methodology performs HCI development with SDLC activities in parallel with an iterative process and conducts evaluation throughout the entire process. The philosophy for HCSDLC is that ISD should meet both organizational and individual needs.

<Figure 8> shows HCSDLC and OO-HCI methodologies. They cover both SDLC or OO and HCI concerns and activities. The vertical line in the middle roughly divides the different emphasis of

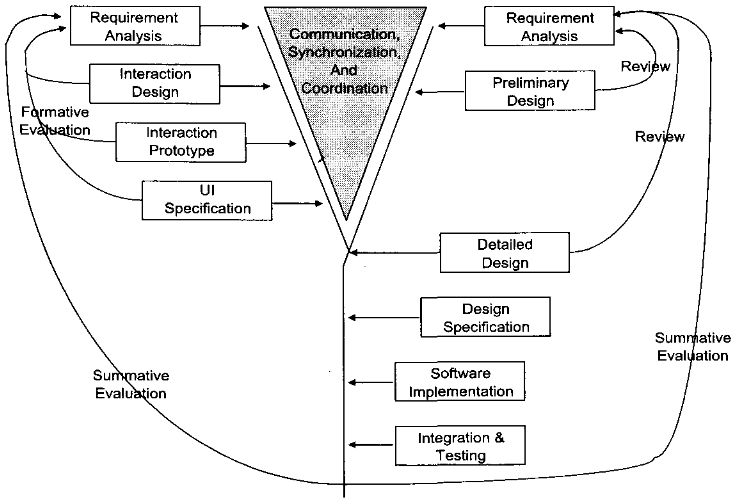
SDLC or OO and HCI development. Activities at the same horizontal level can be performed in parallel, and activities which are placed higher inside a phase are performed before than activities placed lower.

4.2.2 Integration by Parallel Development

Pyla *et al.* (2003) propose a software development process model (Y-model) with parallel and independent UE and SE processes. In this framework the design document containing UE and SE specification is given to the developers when the project reaches the implementation stage. Coordination between the UE and SE teams during the requirements generation and design stages is required to understand and integrate the dependencies and constraints between the interface specifications and the functional core.



〈Figure 8〉 Integration of UCD/HCI and SDLC by Joint Development



〈Figure 9〉 Integration of UCD and SDLC by Parallel Development (Pyla *et al.*, 2003)

V. Practical and Research Implications

In the previous two sections a number of WIS methodologies and techniques were reviewed and compared following a review framework. This study obviously has its limits. First, this work does not include all WIS methodologies and techniques. As mentioned in the review framework section this review does not cover some ISD paradigms and approaches such as Software-Oriented Architecture which is based on the Web service standards. Second, sometimes the direct comparison of a particular phase in different methodologies is not feasible since the depth with which a methodology describes a phase and the guidelines given for a phase in the WIS development process varies enormously. However, the review framework and the survey in general show a number of implications, and these are discussed here.

5.1 Implications for Practitioners

This survey of the state of the art of WIS development methodologies shows that no methodology offers a wholly suitable framework for the development of WIS from the WIS development process perspective. For example, few hypermedia/web design methodologies handle the requirement specification and implementation phases in detail. UCD also focuses on the analysis and design phase. Web Engineering and Integrated SDLC approaches can be described as incomplete since they do not provide details. Thus, WIS developers need to select and customize methodologies to the particular needs of their development environment. And, they need to use "best of practice" techniques known to have been successful in solving a project's problems.

Practitioners have been somewhat slow in adopting ISD methodologies (Fitzgerald, 1998). Lack of awareness is one of the possible explanations for why WIS methodologies are not being used to any significant extent by practitioners (Barry and Lang, 2001; Lang, 2002). Therefore, WIS developers need to exhaustively study the current research results in the field of WIS and others affecting it in order to gain the knowledge, experience and skills. WIS developers have to continually be in the process of studying, understanding and experiencing (use and test) emerging tools and technologies.

To help WIS developers to study recent research results and to gain knowledge, experience and skills, Christodoulou and Papatheodorou (2004) suggest to run a portal, called WEP (Web Engineering Resources Portal). A number of industry best practices can provide a solid foundation. Different developers will not interpret and apply the same methodology in the same way; nor will the same developer the same methodology in the same way in different development situations (Fitzgerald, 1998). More developers' experiences need to be published and shared.

5.2 Research Issues

Our analysis indicates that there is a need for advances in the level of methodologies and techniques of WIS development. In spite of the attention that WIS is currently attracting, there is no standardized and complete methodology to serve as a development reference. A WIS methodology is supposed to cover all phases of the WIS life cycle. That is, it should specify the activities to follow in each phase, the techniques to apply, and the products resulting from them. It should give an adequate treatment to the information and functionality requirements, as well as the new characteristics of WIS inherited from multimedia. The methodologies and techniques from hypermedia/web design, UCD, and traditional SDLC can not just be borrowed and piled up. They must be integrated to create a viable and usable framework. The right process is one that properly marries tried and true engineering with an understanding of what makes WIS unique.

Ⅵ. Conclusion

WIS development methodologies become more important than before as the demand for higher quality WIS is increasing. There has been a rising interest in the development of WIS methodologies. A number of methodologies and techniques have been proposed. However, we have little understanding about what they are and even less idea about their effectiveness. This paper presents the results of an exhaustive survey of the most important proposals currently applicable to the development of WIS.

This survey attempted to provide a better understanding of the nature of WIS methodologies. This review and comparison of different kinds and types of WIS methodologies can be used as a practical tool for selecting methodologies and techniques by practitioners. This work will also enable researchers to develop a better WIS methodology. A field survey of methodologies-in-action for the WIS development will be a good starting point.

References

Albert, T. C., P. B. Goes, and A. Gupta, "GIST: A Model for Design and Management of Content and Interactivity of Customer- Centric Web Sites", *MIS Quarterly*, Vol.28, No.2, June 2004.

Alt, R., C. Reichmayr, M. A. Casar, and R. Zurmuhlen, "Evolution of Electronic Catalogs to Customer Process Portals - A Structured Approach at ETA S. A.", Proc. of 15th Bled Electronic Commerce Conference, June 2002.

Anderson, J., F. Fleek, K. Garrity, and F. Drake, "Integrating Usability Techniques into Software Development", *IEEE Software*, Vol.18, No.1, January/February 2001.

Avison, D. and G. Fitzgerald, *Information Systems Development: Methodologies, Techniques and Tools*, McGraw-Hill, Maidenhead, 2002.

Baskerville, R. and J. Pries-Heje, "Racing the E-Bomb: How the Internet Is Redefining Information Systems Development Methodology", In FitzGerald, B. & Russo, N. and DeGross, J. (Eds.), Realigning Research and Practice in Is Development: The Social and Organisational Perspective, Kluwer, 2001.

Baresi, L., F. Garzotto, and P. Paolini, "Extending UML for Modeling Web Applications", Proceedings of the 34th Hawaii International Conference on System Sciences, 2001.

Barry, C. and M. Lang, "A Survey of Multimedia and Web Development Techniques and Methodology Usage", *IEEE Multimedia*, April- June, 2001.

Beyer, H. and K. Holtzblatt, Contextual Design: A Customer-Centered Approach to Systems Design, Morgan Kaufmann Publishers, 1997.

Beyer, H., K. Holtzblatt, and L. Baker, "An Agile Customer-Centered Method: Rapid Contextual Design", Agile Universe 2004, LNCS 3134, Zannier *et al.* (Eds.), Springer-Verlag, 2004.

Booch, G., J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language: A User Guide*, Addison-Wesley, 1999.

Carroll, J. M., *Making Use: Scenario-Based Design of Human-Computer Interactions*, MIT Press, 2000.

Carstensen, P. H. and L. Vogelsan, "Design of Web-based Information Systems - New Challenges for Systems Development", Proc. Of ECIS 2001, Kranj, Slovenia, 2001.

Ceri, S., P. Fraternali, A. Bongio, M. Brambilla, S. Comai, and M. Matera, Designing Data-Intensive Web Applications, Morgan Kaufman, 2003.

Ceri, S., P. Fraternali, and M. Matera, "Conceptual Modeling of Data-Intensive Web Applications", *IEEE Internet Computing*, July-August, 2002.

Christodoulou, S. and T. Papatheodorou, "WEP: A Reference Model and the Portal of Web Engineering Resources", Proc. Of ACM Hypertext 2004, 2004.

Christodoulou, S. P., P. A. Zafiris, and T. S. Papatheodorou, "Web Engineering: The Developers' View and a Practitioner's Approach", LNCS 2016, 2001.

Constantine, L. L. and L. A. D. Lockwood, *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*, Addison-Wesley, 1999.

Constantine, L. L. and L. A. D. Lockwood, "Usage-Centered Engineering for Web Applications", *IEEE Software*, March/April 2002.

Constantine, L. L. and L. A. D. Lockwood, "Use Cases in Task Modeling and User Interface Design", CHI 1999, May 1999.

Cooper, A., The Inmates are Running the Asylum: Why High-tech Products Drive Us Crazy and How to Restore the Sanity, Sams, 1999.

Dennis, A. R., "Lessons from Three Years of Web Development", *Communications of ACM*, Vol. 41, No.7, July 1998.

Deshpande, Y. and S. Hansen, "Web Engineering: Creating a Discipline Among Disciplines", *IEEE Multimedia*, Vol.8, No.2, April-June, 2001.

Deshpande, Y., S. Murugesan, A. Ginige, S. Hansen, D. Schwabe, M. Gaedke, and B. White, "Web Engineering", *Journal of Web Engineering*, Vol.1, No.1, 2002.

De Troyer, O. and C. Leune, "WSDM: A User-centered Design Method for Web Sites", Proceedings of 7th International World Wide Web Conference, 1997.

Dillman, D. A., *Mail and Internet Surveys: The Tailored Design Method*, John Wiley & Sons, 1999.

Dumas, J. S. and J. C. *Redish, A Practical Guide to Usability Testing*, Intellect, 1999.

Escalona, M. J. and N. Kock, "Requirements Engineering for Web Application - A Comparative Study", *Journal of Web Engineering*, Vol.2, No.3, 2004.

Escalona, M. J., M. Mejias, and J. Torres, "Methodologies to develop Web Information Systems and Comparative Analysis", *Upgrade*, Vol.3, No.3, June 2002.

Faulkner, X. and F. Culwin, "Enter the Usability Engineer: Integrating HCI and Software Engineering", Proc. Of 5th Annual SIGCSE/SIGCUE ITiCSE, Conf. on Innovation and Technology in Computer Science Education, Helsinki, Finland, 2000.

Ferre, X., "Incorporating Usability into an Object Oriented Development Process", INTERACT '01 Workshop, Tokyo, Japan, July, 2001.

Ferre, X., N. Juristo, H. Windl, and L. Constantine, "Usability Basics for Software Developers", *IEEE Software*, January-February, 2001.

Fitzgerald, B., "Formalised Systems Development

Methodologies: A Critical Perspective", *Information Systems Journal*, January, 1996.

Fitzgerald, B., "An Empirically-Grounded Framework for the Information Systems Development Process", Proc. Of Int'l Conf. on Information Systems, December 1998.

Fitzgerald, B., N. Russo, and T. O'Kane, "Software development method tailoring at Motorola", *Communication of ACM*, Vol.46, No.4, 2003.

Fraternali, P., "Tools and Approaches for Developing Data-Intensive Web Applications: A Survey", *ACM Computing Survey*, Vol.31, No.3, September, 1999.

Fraternali, P. and P. Paolini, "A Conceptual Model and Tool Environment for Developing More Scalable, Dynamic, and Customizable Web Applications", Proc. Of EDBT '98, 1998.

Fraternali, P. and P. Paolini, "Model-Driven Development of Web Applications: The Autoweb System", *ACM Transactions on Information Systems*, Vol.28, No.4, October 2000.

Fuccella, J., "Using User Centered Design Methods to Create and Design Usable Web Sites", SIGDOC 1997, Utah, USA, 1997.

Garzotto, F., P. Paolini, and D. Schwabe, "HDM: A Model-based Approach to Hypertext Application Design", *ACM Transaction of Information Systems*, Vol.11, No.1, 1993.

Genige, A. and S. Murugesan, "Web Engineering: An Introduction", *IEEE Multimedia*, January-March 2001.

Ginsburg, M., "The Catacomb Project: Building a User-Centered Portal the Conversational Way", Proc. Of 4th International Workshop on Web Information and Data Management, November 2002.

Gomez, J., C. Cachero, and O. Pastor, "Extending A Conceptual Modeling Approach to Web Application Design", *First Int'l Workshop on Web-Oriented Software Technology*, Valencia, 2001.

Gould, J. D., S. J. Boies, and C. Lewis, "Making Usable, Useful, Productivity-Enhancing Computer Applications", *Communications of ACM*, Vol.34, No.1, January 1991.

Granollers, T., "User Centered Design Process Model: Integration of Usability Engineering and Software Engineering", Interact 2003, Zurich, 2003.

Gulliksen, J. and I. Boivie, "Usability Throughout the Entire Software Development Lifecycle: A Summary of the INTERACT 2001 Workshop", Technical Report 2001-026, Dept. of Information Technology, Uppsala University, Nov. 2001.

Hackos, J. T. and J. C. Redish, *User and Task Analysis for Interface Design*, John Wiley & Sons, 1998.

Halasz, F. and M. Schwartz, "The Dexter Hypertext Reference Model", *Communications of ACM*, February, 1994.

Hennicker, R. and N. Kock, "A UML-based Methodology for Hypermedia Design, LNCS 1939, Springer-Verlag, 2000.

Hix, D. and H. R. Hartson, *Developing User Interfaces: Ensuring Usability Through Product and Process*, John Wiley & Sons, 1993.

Holck, J., "4 Perspectives on Web Information Systems", Proc. Of 36th Hawaii Int'l Conf. on Systems Science (HICSS '03), 2002.

Hoyer, W. D. and D. McInnis, *Consumer Behaviour*, Houghton Mifflin Co., 2003.

Isakowitz, T., M. Bieber, and F. Vitali, "Web Information Systems", *Communications of ACM*, Vol.41, No.7, 1998.

Isakowitz, T., E. Stohr, and P. Balasubramanian, "A

Methodology for the Design of Structured Hypermedia Applications", *Communications of ACM*, Vol.8, No.38, 1995.

Iivari, J., R. Hirschheim, and H. K. Klein, "A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches", *Journal of Management Information Systems*, Vol.17, No.3, Winter 2000-2001.

Iivari, J., R. Hirschheim, and H. K. Klein, "A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies", *Information Systems Research*, Vol. 9, No.2, June 1998.

Jacobson, I., G. Booch, and J. Rumbaugh, *The Unified Software Development Process*, Addison-Wesley, 1999.

Jokela, T., "Beyond Usability Methods: Usability Engineering Through Processes and Outcomes", *Cutter IT Journal*, Vol.16, No.10, October 2003.

Kappel, G., E. Michlmayr, and B. Proll, "Web Engineering - Old Wine in New Bottles?," ICWE 2004, LNCS 3140, Koch, N., P. Fraternali, and M. Wirsing, (Eds.), 2004.

Knobel, C., "Leveraging Usability to Maximize Your Web Site", *AICPA InfoTech Update*, January-February 2002.

Kock, N., Software Engineering for Adaptive Hypermedia Applications, Ph.D. Thesis, FAST Reihe Softwaretechnik Vol.12, Uni-Druck, Munich, Germany, 2001.

Hennicker, R. and N. Koch, "A UML-based Methodology for Hypermedia Design", LNCS 1939, 2000.

Lang, M., "Hypermedia Systems Development: Do We Really Need New Methods?", *Informing Science*, June 2002.

Lang, M. and B. Fitzgerald, "Hypermedia Systems Development Practice: A Survey", *IEEE Software*, Vol.20, No.2, March-April 2005.

Lange, D., "An Object-Oriented Design Approach for Developing Hypermedia Information Systems", *Journal of Organizational Computing and Electronic Commerce*, Vol.6, No.3, 1996.

Lee, H., C. Lee, and C. Yoo, "A Scenario-based Object-Oriented Hypermedia Design Methodology", *Information & Management*, Vol.36, 1999.

Loeber, S. and A. Cristea, "A WWW Information Seeking Process Model", *Educational Technology & Society*, Vol.6, No.3, 2003.

Lowe, D. B., A. J. Bucknell, and R. G. Webby, "Improving Hypermedia Development: A Reference Model-based Process Assessment Method," Proc. Of Hypertext 1999, Darmstadt, Germany, Hall, 1994.

Lowe, D. and W. Hall, *Hypermedia and the Web: An Engineering Approach*, John Wiley, 1999.

Lowe. D. and R. Webby, "Utilisation of Process Modeling in Improving the Hypermedia Development Process", *The New Review of Hypermedia and Multimedia*, Vol.5, 1999.

Mao, J., K. Vredenburg, P. W. Smith, and T. Carey, "The State of User-Centered Design Practice", *Communications of ACM*, Vol.48, No.3, March 2005.

Mayhew, D., *The Usability Engineering Lifecycle*, Morgan Kaufmann Publishers, 1999.

Mayhew, D., "Introduction," in *Human Factors and Web Development*, Ed. Julie Ratner, Lawrence Erlbaum Associates, 2003.

McDonald, A. and R. Welland, "Web Engineering in Practice", Proc. of 10th Int'l WWW Conference, May 2001.

McGraw, K. and K. Harbison, *User-Centered Re-*

quirements: The Scenario-based Engineering Process, Lawrence Erlbaum Associates Publishers, 1997.

Murugesan S., Y. Deshpande, S. Hansen, and A. Ginige, "Web Engineering: A New Discipline for Development of Web-based System", Web Engineering 2000, LNCS 2016, Murugesan, S. and Deshpande, Y. (Eds.), 2001.

Nielsen, J., "The Usability Engineering Life Cycle", *IEEE Computer*, March, 1992.

Nielsen, J., *Usability Engineering*, AP Professional, 1993.

Nielsen, J., *Designing Web Usability*, New Riders Publishing, 2000.

Notess, M. and E. Blevis, "Integrating Human-Centered Design Methods from Different Disciplines: Contextual Design and PRInCiPleS," FUTUREGROUND 2004, Melbourne, Australia, Nov. 17-21, 2004.

Norton, K. S., "Applying Cross-Functional Evolutionary Methodologies to Web Development", WebEngineering 2000, LNCS 2016, Murugesan, S. and Y. Deshpande, (Eds.), Springer-Verlag, 2001.

Powel, T. A., *Web Site Engineering*, Prentice-Hall, 1998.

Preece, J., Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey, *Human-Computer Interaction*, Addison Wesley, 1994.

Preece, J., Y. Rogers, and H. Sharp, *Interaction Design: Beyond Human-Computer Interaction*, New York, Wiley 2002.

Pressman, R. S., "Can Internet-Based Applications Be Engineered", *IEEE Software*, Vol.15, No.5, September/October 1998.

Pressman, R. S., "What a Tangled Web We Have", *IEEE Software*, Vol.17, No.1, January/February 2000.

Pressman, R. S., *Software Engineering: A Practitioner's Approach*, McGraw Hill, 2004.

Pruitt, J. and J. Grudin, "Personas: Practice and Theory", ACM 2003.

Pyla, P. S., M. A. Perez-quinones, J. D. Arthur, and H.R. Hartson, "Towards a Model-Based Framework for Integrating Usability and Software Engineering Life Cycles", Proc. Of Bridging the SE & HCI Communities Workshop in UNTERACT, 2003.

Radle, K. and S. Young, "Partnering Usability with Development: How Three Organizations Succeeded", *IEEE Software*, Vol.18, No.1, January/February 2001.

Ramler, R., K. Wolfmaier, and E. Weippl, "From Maintenance to Evolutionary Development of Web Applications: A Pragmatic Approach," ICWE 2004, LNCS 3140, Koch, N., P. Fraternali, and M. Wirsing, (Eds.), Springer-Verlag, 2004.

Ratner, J. (Ed), *Human factors and Web development*, 2nd Ed., Mahwah, N.J.: Lawrence Erlbaum Associates, 2003.

Reifer, D. J., "Web Development: Estimating Quick-to-Market Software", *IEEE Software*, Vol.17, No.6, November/December 2000.

Retschitzegger, W. and W. Schwinger, "Towards Modeling of DataWeb Applications - A Requirement's Perspective", *Proc. Of AMCIS*, 2000.

Salvendy, G. (Ed.), *Handbook of Human Factors and Ergonomics*, John Wiley, 1997.

Sawasdichal, N. and S. Poggenpohl, "User Purposes and Information-Seeking Behaviors in Web-Based Media: A User-Centered Approach to Information Design on Websites", DIS 2002, 2002.

Schneiderman, B., *Designing the User Interface:*

Strategies for Effective Human-Computer Interaction, Addison Wesley, 1998.

Schuler, D. and A. Namioka (Ed.), *Participatory Design: Principles and Practices*, Lawrence Erlbaum Associates, 1993.

Schwabe, D. and R. de Almeia Pontes, "OOHDM-Web: Rapid Prototyping of Hypermedia Applications in the WWW", ACM SigWeb Newsletter, June 1999.

Schwabe, D. and G. Rossi, "Object-Oriented Hypermedia Design Model", *Communications of ACM*, Vol.38, No.8, 1995.

Schwabe, D., G. Rossi, and S. Barbosa, "Systematic Hypermedia Design with OOHDM", Proc. Of ACM Int'l Conf. on Hypertext, Washington, March, 1996.

Seffah, A. and E. Metzker, "The Obstacles and Myths of Usability and Software Engineering", *Communications of ACM*, Vol.47, No.12, December 2004.

Selmi, S. S., N. Kraiem, and H. B. Ghezala, "Toward a Comprehensive View of Web Engineering", ICWE 2005, Springer-Verlag LNCS 3579, 2005.

Singh, S. and P. Kotze, "An Overview of Systems Design and Development Methodologies with Regard to the Involvement of Users and Other Stakeholders", Proceedings of SAICSIT, 2003.

Skyrme, D. J., "Multimethodoloies: the Knowledge Perspective", in Multi-methodology, Mingers, J. and A. Gill, eds., John Wiley & Sons, 1997.

Sousa, K. S. and E. Furtado, "RUPi - A Unified Process that Integrates Human-Computer Interaction and Software Engineering", Int'l Conf. on Software Engineering, Portland, OR, USA, 2003.

Spencer, R., "The Streamlined Cognitive Walkthrough Method: Working Around Social Constraints Encountered in Software Development Company", Proc. Of Conference on Human Factors in Computing System (CHI '00), The Hague, 2000.

Standing, C., *Internet Commerce Development*, Artech Houser Publishers, Boston, 2000.

Tai, H., K. Mitsui, T. Nerome, M. Abe, K. Ono, and M. Hori, "Model-driven Development of Large-scale Web Applications", *IBM Research & Development*, Vol.48, No.5/6, September/November 2004.

Taylor, M. J., J. McWilliam, H. Forsyth, and S. Wade, "Methodologies and Website Development: a survey of practice", *Journal of Information and Software Technology*, Vol.44, No.6, 2002.

Vessey, I. and R. Glass, "Strong vs. Weak: Approaches to Systems Development", *Communications of ACM*, April 1998.

Vidgen, R., "Constructing a Web Information System Development Methodology", *Information Systems Journal*, Vol.12, 2002.

Vidgen, R., D. Avison, B. Wood, and T. Wood-Harper, Developing Web Information Systems: From Strategy to Implementation, Elsevier Science, 2002.

Vilain, P., D. Schwabe, and C. Sieckenius, "A Diagrammtic Tool for Representing User Interaction in UML", LNCS, Proc. Of UML 2000, York, England, 2000.

Zhang, P., J. Carey, D. Te'eni, and M. Tremaine, "Integrating Human-Computer Interaction Development into the Systems Development Life Cycle: A Methodology", *Communications of the Association for Information Systems*, Vol. 15, 2005.

Information Systems Review
Volume 8 Number 2
August 2006

Web Information System Development Methodologies: A Survey of the Literature

Jae Hwa Choi[*]

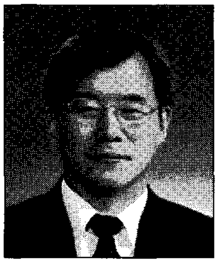
Abstract

The success of a Web information system (WIS) can be achieved only if it is developed in a proper and rigorous manner. As WIS has currently assumed a greater significance in organizations, WIS developers need a better understanding of methodologies supporting the process of developing WIS. Existing literature provides few guidelines to the WIS developers. This work provides a comprehensive review and a comparative study of the current approaches available in the WIS field that use different methodologies and techniques to develop WIS. The goal of this paper is to compare the WIS development methodologies and techniques in literature, provide practical guidelines, and generate new research issues.

Keywords: Web Information System, Information System Development Methodology, Web Engineering

[*] Professor, Dankook University

◐ 저 자 소 개 ◑



Jae Hwa Choi (jchoi@dankook.ac.kr)

Dr. Choi is a professor of Information Systems at the College of Business and Economics of the Dankook University. He received a Ph.D. degree in Information Systems from the Roberts School of Business at the University of Maryland in College Park. He was an assistant professor of Information Systems at the School of Management of the University of Michigan in Dearborn. His research interests include the Web technology, knowledge management, electronic commerce, and information ethics.

논문접수일 : 2006년 4월 17일　　게재확정일 : 2006년 6월 27일
1차 수정일 : 2006년 6월 12일