Modeling Ubiquitous Web Applications - A Comparison of Approaches

G. Kappel[1] B. Pröll[2] W. Retschitzegger[1] W. Schwinger[3] T. Hofer[3]

[1]Department of Information Systems
Johannes Kepler University Linz
{gerti | werner}@ifs.uni-linz.ac.at

[2]Institute for
Applied Knowledge Processing
Johannes Kepler University Linz
bproell@faw.uni-linz.ac.at

[3]Software Competence Center
Hagenberg
{wieland.schwinger | thomas.hofer}@scch.at

Abstract

E-commerce and m-commerce demand for full-fledged, increasingly complex applications which need to offer ubiquitous access in terms of the anytime/anywhere/anymedia paradigma. From a software engineering point of view, the development of such ubiquitous web applications requires proper modeling methods in order to ensure architectural soundness and maintainability. Recently, web modeling methods started recognizing this fact by providing first concepts for dealing with ubiquity. In this paper, two of these modeling methods are compared, identifying their strengths and shortcomings. As a prerequisite, an evaluation framework is introduced, using the notion of customization as the uniform mechanism to enable ubiquity. Customization adapts a web application towards a particular context which reflects the environment the application is running in. To enable a holistic view on the development process of a ubiquitous web application, customization is regarded as a new modeling dimension, influencing all other tasks of ubiquitous web application development.

1 Introduction

The Internet and in particular the World Wide Web have introduced a new era of computing, providing the basis for promising application areas like e-commerce [26], [41] and m-commerce [9]. These application areas have dramatically boosted the demand for services which enable *ubiquitous access,* thus adhering to the *anytime/anywhere/anymedia paradigm* [1], [47]. Consequently, ubiquity offers new opportunities for web applications in terms of time-aware, location-aware, device-aware and personalized services.

Considering ubiquitous web applications from a software engineering point of view, as their complexity increases, so does the importance of modeling techniques [6], [21], [22], [23], [40]. Models of a ubiquitous web application prior to its construction are essential for comprehension in its entirety, for communication among project teams, and to assure architectural soundness and maintainability.

There is already a couple of methods especially dedicated to the modeling of traditional web applications (for an overview see [6], [11], [19], [42]). These methods focus on unique characteristics of web applications comprising among others the usage of the hypermedia paradigm in terms of hypertext and multimedia in combination with application logic. Up to now, however, only few of them, especially the *Web Modeling Language (WebML)* [7], [8] and the *Object-Oriented Hypermedia Design Method (OOHDM)* [43], [44] provide first modeling concepts dealing with the ubiquitous nature of today's web applications.

The goal of this paper is twofold: First, an evaluation framework is provided defining the design space for modeling methods of ubiquitous web applications. Second, by comparing the above mentioned approaches on the basis of this framework, strengths and weaknesses are identified, pointing to future directions in modeling ubiquitous web applications. According to these goals, the paper is structured as follows. Section 2 introduces the modeling dimensions of ubiquitous web applications, introducing the notion of customization as the uniform mechanism to capture ubiquity. Section 3 complements our evaluation framework by outlining the design space of customization. Section 4 compares the capabilities of the two modeling methods on the basis of our evaluation framework. The major findings are summarized in Section 5, emphasizing on lessons learned and potential improvements of existing approaches.

2 Modeling Dimensions of Ubiquitous Web Applications

2.1 Modeling Traditional Web Applications

Modeling methods for ubiquitous web applications should first of all consider general requirements which need to be obeyed by any web application modeling method. According to [42], these requirements can be described along three orthogonal dimensions, namely *levels*, *aspects* and *phases* (cf. Figure 1).

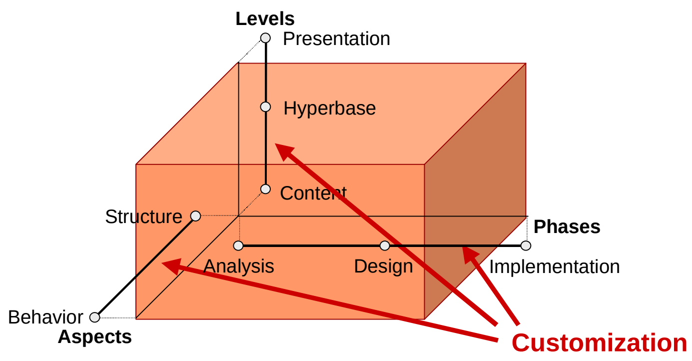

Figure 1. Modeling Dimensions of Ubiquitous Web Applications

The first dimension of web application modeling comprises three different *levels* in terms of content level, hyperbase level, and presentation level. The *content level* refers to domain-dependent data used by the web application and is often managed by means of a database system. The *hyperbase level* denotes the logical composition of web pages and the navigation structure. The *presentation level*, finally, is concerned with the presentation of the hyperbase level, e.g., the layout of each page [17]. The second dimension called *aspects* is orthogonal to the first one, requiring that both *structural aspects* in terms of abstraction mechanisms such as classification, aggregation and generalization, as well as *behavioral aspects* like business logic, activation of navigational nodes, and user interaction need to be considered. Finally, structure and behavior of content, hyperbase and presentation has to be addressed in each phase of the development process, ranging from *analysis* via *design* to *implementation* [31], [35].

2.2 Customization as Additional Modeling Dimension

Besides considering the modeling dimensions of traditional web applications, a ubiquitous web application should be designed from the start to take into account also its ubiquitous nature. For this, we propose the notion of *customization* as the uniform mechanism to enable ubiquity by *adapting* a web application towards a particular *context* which reflects the environment the application is running in. This viewpoint is similar to [36] who differentiates between an afferential component (i.e., the context), an efferential component (i.e., the adaptation) and an inferential component (i.e., the customization itself which maps between context and adaptation)[1]. In general, we suggest a *holistic view on the development process* of a ubiquitous web application by introducing *customization as an additional design dimension*, affecting all dimensions of traditional web application modeling (cf. Figure 1).

3 Design Space of Customization

3.1 History on Customization

For defining the design space of customization, it is useful to consider customization from a historical point of view [28]. Customization represents a major challenge not least since the end user has been put in the middle of concern when developing interactive applications. Consequently, the user interface community is dealing with customization issues already for a long time, which brought up the notion of *adaptive user interfaces,* cf., e.g., [24]. Adaptive user interfaces are designed to tailor a system's interactive behavior considering both individual needs of human users and changing conditions within an application environment. The broader approach of *intelligent or advisory user interfaces* includes adaptive characteristics as a major source of its intelligent behavior, cf., e.g., [5]. Another area dealing with customization but emphasizing more on adapting the content of an application are *information filtering* and *recommender system* [2], [30]. The goal of these systems is to go through large volumes of dynamically generated textual information and present to the user those which are likely to satisfy his/her information requirements. With the emerge of hypertext [13] the need for alternative access paths to information in terms of, e.g., different navigation structures became prevalent leading to another research direction called *adaptive hypertext and hypermedia* [3]. Last but not least, the proliferation of *mobile computing* and *mobile web applications,* in particular, makes it necessary to consider not only user preferences but also the environment in terms of, e.g., location in order to adapt the application [36].

3.2 Customization Dimensions

Although the approaches within the different areas described above use the notion of customization merely in an implementation-oriented sense, they represent a proper basis to derive the design space considering customization at the modeling level. In our view, the design space of customization can be characterized again by three orthogonal dimensions, comprising the *kind of context*, the *granularity of adaptation* and the *degree of customizability* (cf. Figure 2).

[1] Note that, in literature the terms *customization* and *adaptation* are often used interchangeably.

Kind of Context. The first dimension covers the *kind of context*, reflecting the environment of the application which is considered by customization. The majority of the approaches like [8], [14], [15], [16], [25], [32], [33] and [46] focus on the issue of *personalization* in terms of a *user context*. This is done by making assumptions about relevant user characteristics and preferences to get personalized services from a certain resource, or even to personalize already the discovery of resources [12], [39]. Also a considerable number of approaches take device and network properties in terms of *device context* and *network context* into account [32], [34]. Device context and network context are often considered together which is reasonable since mobile devices also imply a wireless connection carrying certain network constraints. [18] for example distinguishes between three aspects of "client variation" comprising network variations such as bandwidth and latency, hardware variations like screen size, and memory and software variations including the data types the client is able to handle. [29] additionally introduces a *location context* comprising physical locations and logical locations (e.g., at home as opposed to at work). It has been encountered, however, that only few of the surveyed approaches explicitly regard location context [36], [38]. This is due to both technical deficiencies and lack of legal regulations. Similarly, *time context* which makes certain customizations dependent on the point in time when a certain service is accessed is rarely considered in literature. Independent of its kind, a context can be separated into an *unchangeable part* which is captured by monitoring the environment (e.g., the kind of device used or a cell identifier indicating the current location when actually accessing an application) and a *changeable part* which is voluntarily entered by a user (e.g., user preferences), a device vendor (e.g., device characteristics) or a designer (e.g., street maps). The former is furtheron called *current context*, the latter is, conforming to literature, referred to as *profile* [48].

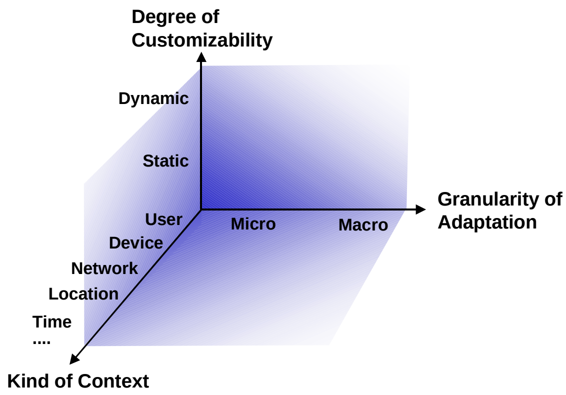

Figure 2. Design Space of Customization

Granularity of Adaptation. The second dimension indicates the *granularity of adaptation* ranging from *micro adaptation* to *macro adaptation*. Whereas micro customization is concerned with fine-grained adaptations (e.g., disabling a link on a page, determining the membership of an object in a class), macro adaptation means that depending on the context, rather large parts of an application are adapted (e.g., instead of an indexed guided tour, use a simple bullet list). Subject of adaptation can be each of the levels of a web application as introduced in Section 2.1. Note that, there is no exact border between micro and macro adaptation. In its most extreme form, macro adaptation simply means that depending on the context, the whole application realizing a certain service is substituted by another one, thus better fitting in a certain context. This extreme form of macro customization often occurs together with the combination of static context and static adaptation (cf. below).

Degree of Customizability. The third dimension, the *degree of customizability*, expresses that both, context and adaptation can be either *static*, i.e., pre-defined or *dynamic*, i.e., determined during run-time. An example for static context and static adaptation could be to select a pre-defined version of a certain web application for either access through a PC or through a WAP enabled device. An example for the fully dynamic case would be to adapt the resolution of an image on the fly, due to changes in bandwidth. Applications supporting only static contexts and/or adaptation are often called *adaptable* whereas those supporting also the dynamic case are considered to be *adaptive* [33].

4 Comparison of WebML and OOHDM

In the following the customization capabilities of WebML [7], [8] and OOHDM [43], [44] are discussed on the basis of the evaluation framework given in the previous sections, aiming at the elicitation of their strengths and shortcomings along. These two are, to the best of our knowledge, two representative approaches out of the few existing methods dealing with ubiquity and customization mechanisms in turn at a modeling level [28].

4.1 Kind of Context

User Context. User context is addressed in WebML as well as in OOHDM (cf. Figure 3). Both assume that the user accessing the web application can be identified by means of predefined variables, herewith representing the current context. WebML provides in addition, at the content level, two dedicated *entities* `User` and `User Group` for representing profile information about individual users and user communities, respectively. These entities contain some pre-defined attributes such as name, password and email and are also extensible so that the designer may add supplementary information. In contrast to WebML, OOHDM provides no explicit profile for capturing user and user group information but it is rather left to the designer to define appropriate *classes* for representing user profile information at the content level. Since at the content level OOHDM uses an object-oriented notation in terms of UML [45], the user profile can also capture user-defined methods.

		WebML	OOHDM
User	Current	✔	✔
	Profile	✔	✘
Time	Current	✔	✘
	Profile	✘	✘
Device	Current	✘	✘
	Profile	✘	✘

Figure 3. Supported Kind of Context

Time Context. Time context is considered by WebML only. Two dedicated variables `CurrentDate` and `CurrentTime` provide the current context at the server where the ubiquitous web application is running. Time profiles encompassing, e.g., time zones or time-of-day settings is not supported.

Device Context. Regarding device context, both approaches do not provide explicit concepts, neither for identifying the current device, nor for representing device profile information (therefore the ✘ in Figure 3). They propose, however, to implicitly consider device context by simply defining appropriate adaptations of the presentation level for each device, as will be discussed in Section 4.2 and Section 4.3.

Different to device context, other kinds of context such as network context or location context are not considered at all.

4.2 Granularity of Adaptation

Micro Adaptation in WebML. In WebML, micro adaptation is restricted to the *content level* (cf. Figure 6). Similar to views in database systems [6], WebML offers a so called *derivation model.* A derivation model of the content can be declaratively specified by means of a query language called *WebML-OQL* which is a subset of OQL [37]. It enables queries on the content level including profile information. In general, the derivation mechanism of WebML allows the designer to adapt *entity attribute values*, *entities* themselves and *relationships* between entities. Concerning entities, WebML supports adaptation in two directions. First, the number of its instances can be constrained and second, its structure can be changed by either dropping existing attributes or by adding attributes having a constant value or a derived one. The example illustrated in Figure 4 defines a derived entity `PreferredTourItems` comprising all those instances of the entity `TourItems` taking part in the relationship `UserIsInterestedIn`. Note that, WebML basically uses XML [49] at all three levels of a web application, partly enhanced by some graphical representation such as ER diagrams [10].

```
<ENTITY id="PreferredTourItems"
        super="TourItems" value="TourItems as i
        WHERE i IS IN UserIsInterestedIn">

</ENTITY>
```

Figure 4. Adapting Entities in WebML

Finally, relationships can be adapted in WebML either by specializing an existing relationship or by concatenating two or more existing relationships.

Micro Adaptation in OOHDM. In contrast to WebML, OOHDM restricts micro adaptation to the *hyperbase level*, enabling the designer to adapt *node attribute values*, *node classes* and *link classes* (cf. Figure 6). OOHDM uses plain OQL-queries to support the adaptation of attribute values. Regarding the adaptation of node classes and link classes, OOHDM allows only to restrict their visibility by indicating authorized users or user groups within the node definition.[2] Since there is no possibility to constrain the instances of a node, all instances of that class of the content level, which is mapped to the node are automatically adopted.

The example given in Figure 5 specifies a node `PreferredTourItems` consisting of instances of the content class `TourItems` and being available for users of the content class `RegisteredVistor` only. Note that, the notation used is proprietary to OOHDM.

```
NODE PreferredTourItems
     FROM TourItems: i, user: RegisteredVisitor
```

Figure 5. Adapting Node Classes in OOHDM

Macro Adaptation. WebML as well as OOHDM allows macro adaptation at the *hyperbase level* and at the *presentation level* (cf. Figure 6). This means that on top of a certain content level, several different versions of the hyperbase level and on top of a certain hyperbase level, several different

[2] For the sake of completeness, it has to be noted that OOHDM provides a concept called *context classes*, which allows a node to have different structures. The initiator of such an "adaptation", however, is not a context in the sense of an environmental state, but rather the application state itself, resulting from the user’s navigation path.

presentation levels can be specified, which are tailored towards certain kinds of context.[3] In this way, also the implicit device context can be realized, as mentioned in the previous section.

		WebML	OOHDM
Micro	**Content**	✔	✘
	Hyperbase	✘	✔
	Presentation	✘	✘
Macro	**Content**	✘	✘
	Hyperbase	✔	✔
	Presentation	✔	✔

Figure 6. Supported Granularity of Adaptation

An adapted version of the hyperbase level is called *site view* in WebML and is defined, completely independent of other possibly existing site views. This approach lacks reusability since any common structure within the adapted versions must replicated. In OOHDM the different adapted versions are integrated into one model called *navigational class schema* by using the Decorator pattern [20]. Thus, in contrast to WebML, reusability of common parts of the adapted versions is supported. Regarding the presentation level both approaches lack concepts for reusability.

4.3 Degree of Customizability

Context. Regarding the degree of customizability with respect to the kind of context, one can assess that the user context is *statically and dynamically* supported by both, WebML and OOHDM (cf. Figure 7). Time context is only supported dynamically by OOHDM. Finally, the implicit notion of device context as described in Section 4.1 can be considered to be some kind of static context.

		WebML	OOHDM
Static	**Context**	✔	✔
	Micro Adaptation	✘	✘
	Macro Adaptation	✔	✔
Dynamic	**Context**	✔	✔
	Micro Adaptation	✔	✔
	Macro Adaptation	✘	✘

Figure 7. Supported Degree of Customizability

Micro Adaptation. Considering the degree of customizability with respect to micro adaptation, both, WebML and OOHDM focus on the *dynamic* case since micro adaptation is regarded to be performed at runtime (cf. Figure 7). It has to be noted that OOHDM proposes micro adaptation to occur during node initialization. It is not clear, however, when node initialization itself takes place, e.g., it could be triggered by every modification of the underlying content or not before the node is accessed by a user.

Macro Adaptation. Macro adaptation is considered to be purely static for both approaches, at the *hyperbase level* and at the *presentation level*. This static macro adaptation is considered together with a static context as described above. One exception to this is WebML which supports, at the hyperbase level, dynamic context in terms of user and time, in combination with static macro adaptation using different site views. For this, WebML provides a so called *personalization model* in terms of *rules*. *Rules* are used to specify when, at runtime, a certain site view is to apply. They

[3] Note that, OOHDM supports another level of indirection in that the presentation level is further separated into a logical, i.e., implementation-independent part and a physical, i.e., implementation-dependent part. Consequently, a certain logical presentation can be translated into various physical presentations.

are formulated in XML according to the well-known *event/condition/action paradigm* [27]. *Events* considered include the start/end of a session, page access and data changes. The *condition* evaluates a predicate or issues a WebML-OQL query on the content and on the profile of the user issuing the event. Finally, the *action* allows besides manipulating information at the content level, to select a site view tailored towards the current context. The example given in Figure 8 shows how to specify that, at session startup (`SessionStart`), a user which has indicated expert knowledge (`User.expert`) encounters a customized navigation structure in terms of the site view `ExpertView`.

```
<RULE id="AssignExpertView">
  <EVENT eventType="SessionStart"/>
  <CONDITION predicate="User.expert = 'YES'"/>
  <ACTION action="Assign(SiteView, 'ExpertView')"/>
</RULE>
```

Figure 8. Dynamic Context and Static Macro Adaptation in WebML

Consequently, when considering the degree of customizability, whereas the *context is dynamic* since monitored at runtime by means of the event and the condition part of the rule, the *adaptation* can be regarded as *static* because the customized site view has been specified already at definition time.

5 Summary and Lessons Learned

This paper presented an evaluation framework and a comparison of modeling methods for ubiquitous web applications. For this, the notion of customization was introduced as a uniform mechanism to enable ubiquity by adapting a web application towards a particular context which reflects the environment the application is running in. To enable a holistic view on the development process of ubiquitous web applications, customization is seen as a new modeling dimension, which influences those of traditional web applications, comprising levels, aspects and phases. On the basis of this evaluation framework, a survey of two existing modeling methods was presented, revealing their strengths and weaknesses with respect to the customization design space.

Concerning the kind of context supported, the focus of both approaches is on personalisation by considering individual users and groups of users. Additional kinds of context are only partly supported, e.g., time context in WebML. Consequently, a comprehensive modeling method for ubiquitous web applications should not only offer a wider variety of pre-defined kinds of context which clearly separate between current context and profile information but also extensibility in that the designer may include any further kinds of context not foreseen by the modeling method.

Regarding the granularity of adaptation, micro adaptation is considered at the content level in WebML and at the hyperbase level in OOHDM only. Whereas WebML offers various ways of micro adaptation, OOHDM is restricted to visibility issues only. Macro adaptation is considered by both approaches at the hyperbase and the presentation level. Whereas they are very similar in their adaptation capabilities, reusability of common parts is supported by OOHDM at the hyperbase level only. To provide the designer with more flexibility, both micro and macro adaptation should be seamlessly integrated at all three levels of a ubiquitous web application and additionally support proper concepts for reusability.

Finally, taking a look at the degree of customizability, one can asses that both approaches consider dynamic context in combination with micro adaptation whereas in WebML, macro adaptation is bound to static context only. In contrast, OOHDM uses a powerful rule-based mechanism in order to support dynamic context for macro adaptation thereby better reflecting the dynamic nature of ubiquitous web applications.

Acknowledgements

This work was partially funded by *UWA (Ubiquitous Web Applications)*, an EU-funded Fifth Framework Programme project (IST-2000-25131) and by *CustWeb (Customizable Web Application Modeling)*, a national project of the Software Competence Center (SCCH) Hagenberg.

References

[1] G. D. Abowd, E. D. Mynatt, "Charting Past, Present, and Future Research in Ubiquitous Computing", *ACM Transactions on Computer-Human Interaction*, Vol. 7, No. 1, March 2000.

[2] C. Avery, and R. Zeckhauser, "Recommender Systems for Evaluating Computer Messages", *CACM*, Vol. 40, No. 3, March 1997.

[3] B. Beck, and M. McGinnis: "IBM WebShere Everyplace Suite v1.1", white paper, IBM Corporation, Sept. 1999.

[4] A. Bonifati, S. Ceri, P. Fraternali, A. Maurino: "Building Multi-device, Content-Centric Applications Using WebML and the W3i3 Tool Suite, ER2000 Workshop, LNCS 1921, 2000.

[5] J. M. Carroll, and A. P. Aaronson, "Learning by Doing With Simulated Intelligent Help", *CACM*, Vol. 31, No. 9, Sept. 1988.

[6] S. Ceri, P. Fraternali, and S. Paraboschi, "Design Principles for Data-Intensive Web Sites", *SIGMOD Record*, Vol. 24, No. 1, March 1999.

[7] S. Ceri, P. Fraternali, and S. Paraboschi, "Data-Driven One-To-One Web Site Generation for Data-Intensive Applications", *Proc. of the Conference on Very Large Databases (VLDB)*, Edinburgh, September 1999.

[8] S. Ceri, P. Fraternali, and A. Bongio, "Web Modeling Language (WebML): a modeling language for designing Web sites", *Proc. of the 9th World Wide Web Conference (WWW9)*, Amsterdam, May 2000.

[9] D. Chakraborty, H. Chen, "Service Discovery in the future for Mobile Commerce", *ACM Crossroads*, 2000.

[10] P. Chen, "The Entity-Relationship Model - Toward a Unified View of Data", *ACM Transactions on Database Systems*, Vol. 1, No. 1., March 1976.

[11] S. Christodoulou, G. Styliaras, and T. Papatheodourou, "Evaluation of Hypermedia Application Development and Management Systems", *Proc. of the 9th ACM Conference on Hypertext and Hypermedia*, Pittsburgh, 1998.

[12] I. Cingil, A. Dogac, A. Azgin, "A Broader Approach to Personalization", *C ACM*, Vol. 43, No. 8, Aug. 2000.

[13] E. J. Conklin, "Hypertext: An Introduction and Survey", *IEEE Computer*, Vol. 2, No. 9, September 1987.

[14] P. Cotter, and B. Smyth, "Wapping the Web - A Case Study in Content Personalisation for WAP-Enabled Devices", *Proc. of the Int. Conf on Adaptive Hypermedia and Adaptive Web-based Systems*, Trento, Italy, 2000.

[15] P. De Bra, "Design Issues in Adaptive Web-Site Development", Proc. of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW of the 8th International Word Wide Web Conference, Toronto, Canada, 1999.

[16] J. Fink, A. Kobsa, J. Schreck: "Personalized Hypermedia Information Provision through Adaptive and Adaptable System Features: User Modeling, Privacy and Security Issues", *Proc. of the Workshop Adaptive Systems and User Modeling on the World Wide Web of the 6th Int. Conf. on User Model*ing, Chia Laguna, Sardinia, June 1997.

[17] D. Florescu, A. Levy, A. Mendelzon, "Database Techniques for the World Wide Web: A Survey", *ACM SIGMOD Record*, Vol. 27, No. 3, September 1998.

[18] A. Fox, E. Brewer, S. Gribble, E. Amir "Adapting to Network and Client Variability via On-Demand Dynamic Transcoding", Proc. of the 7th Int. ACM Conf. on Architectural Support for Programming Languages and Operating Systems, 1996.

[19] P. Fraternali, "Tools and approaches for data-intensive Web applications: A survey", *ACM Computing Surveys*, Vol. 31, No. 3, September 1999.

[20] E. Gamma, R. Helm, R. Johnson, J. Vlissides, "Design Patterns - Elements of Reusable Object-Oriented Software", Addison Wesley, 1994.

[21] F. Garzotto, P. Paolini, and D. Schwabe, "HDM - A Model-Based Approach to Hypertext Application Design", *ACM Transactions on Information Systems*, Vol. 11, No. 1, January 1995.

[22] D. German, and D. Cowan, "Towards a Unified Catalog of Hypermedia Design Patterns", *Proc. of the 33rd Hawaii International Conference on System Sciences (HICSS),* Maui, Hawaii, January 2000.

[23] A. Ginige, D. B. Lowe, and J. Robertson, "Hypermedia Authoring", *IEEE Multimedia*, Vol. 2, No. 4, 1995.

[24] M. D. Good, J. A. Whiteside, D. R. Wixon, S. J. Jones, "Building a User-Derived Interface", *CACM*, Vol. 27, No. 10, Oct. 1984.

[25] D. Hicks, K. Tochtermann, Th. Rose, and St. Eich, "Using Meta-Data to Support Customization", *Proc. of the Third IEEE Meta-Data Conference*, Bethesda, Maryland, USA, April 1999.

[26] G. Kappel, W. Retschitzegger, and B. Schröder, "Enabling Technologies for Electronic Commerce", *Proc. of the XV. IFIP World Computer Congress*, Vienna/Austria and Budapest/Hungary, August/September 1998.

[27] G. Kappel, W. Retschitzegger, "The TriGS Active Object-Oriented Database System - An Overview", *ACM SIGMOD Record*, Vol. 27, No. 3, September 1998

[28] G. Kappel, W. Retschitzegger, W. Schwinger, "Modeling Customizable Web Applications - A Requirement's Perspective", *Int. Conf. on Digital Libraries: Research and Practice (ICDL)*, Koyoto, Japan, November 2000.

[29] Th. Kunz, and J. P. Black, "An architecture for adaptive mobile applications", *Proceedings of Wireless 99, the 11th International Conference on Wireless Communications*, Calgary, Alberta, Canada, July 1999.

[30] S. Loeb, D. Terry, "Information Filtering", *CACM*, Vol. 35, No. 12, December 1992.

[31] D. Lowe, and R. Webby, "Web Development Process Modelling and Project Scoping: Work in Progress", *Proc. of the Workshop on Web Engineering*, Brisbane, 1998.

[32] M. McIlhagga, A. Light, and I. Wakeman: "Towards a Design Methodology for Adaptive Applications", *Proc. of the 4th annual ACM/IEEE Int. Conference on Mobile Computing and Networking (MOBICOM)*, Dallas, USA, Oct. 1998.

[33] B. Mobasher, R. Cooley, and J. Srivastava, "Creating Adaptive Web Sites Through Usage-Based Clustering of URLs", *Proc. of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX)*, November 1999.

[34] R. Mohan, John R. Smith, Ch.-Sh. Li, "Adapting Multimedia Internet Content for Universal Access". *IEEE Transactions on Multimedia*, Vol. 1, No. 1, 1999.

[35] J. Nanard, and M. Nanard, "Hypertext Design Environments and Hypertext Design Process". *Communication of the ACM (CACM)*, Vol. 38, No. 8, August 1995.

[36] R. Oppermann, and M. Specht, "A Nomadic Information System for Adaptive Exhibition Guidance", *Proc. of the Int. Conf. on Hypermedia and Interactivity in Museums (ICHIM)*, D. Bearman and J. Trant (eds.), Washington, September 1999.

[37] Object Query Language (OQL), Object Data Management Group, http://www.odmg.org/, 2001.

[38] Oracle: "Oracle Internet Application Server 8i Wireless Edition", data sheet, August 2000.

[39] Ozen, B. T., Kilic, O., Altinel, M., Dogac, A., "Highly Personalized Information Delivery to Mobile Clients", *Proc. of 2nd ACM Int. Workshop on Data Engineering for Wireless and Mobile Access (MobiDE)*, Santa Barbara, USA, May 2001.

[40] T. Powell, *Web Site Engineering*, Prentice Hall, 1998.

[41] B. Pröll, W. Retschitzegger, R. R. Wagner, A. Ebner, "Beyond Traditional Tourism Information Systems - TIScover", *Journal of Information Technology and Tourism*, Vol. 1, Inaugural Volume, 1998.

[42] W. Retschitzegger, and W. Schwinger, "Towards Modeling of DataWeb Applications - A Requirements' Perspective", Proc. of the Americas Conf on Information Systems (AMCIS), California, Vol. I, Aug. 2000.

[43] G. Rossi, D. Schwabe, D., and A. Garrido, "Designing Computational Hypermedia Applications", *Journal of Digital Information (JODI)*, Vol. 1, No. 4, February 1999.

[44] G. Rossi, D. Schwabe, D., Guimarães, R. M., "Designing Personalized Web Applications", Prof. of the *International World Wide Web Conference (WWW)*, Amsterdam, 2001.

[45] J. Rumbaugh, I. Jacobson, G. Booch, *The Unified Modeling Language Reference Manual*, Addison-Wesley, 1998.

[46] J. R. Smith, R. Mohan, and C.-S. Li. "Content-based Transcoding of Images in the Internet". *Proc. of the IEEE International Conference on Image Processing (ICIP),* October 1998.

[47] M. Weiser, "Some computer science issues in ubiquitous computing", *CACM*, Vol. 36, No. 7, July 1993.

[48] World Wide Web Consortium (W3C), Composite Capabilities/Preference Profiles, http://www.w3.org/Mobile, 2000.

[49] World Wide Web Consortium (W3C), Extensible Markup Language (XML), http://www.w3.org/XML/