



A survey on web modeling approaches for ubiquitous web applications

Wieland Schwinger, Werner Retschitzegger, Andrea Schauerhuber, Gerti Kappel, Manuel Wimmer, Birgit Pröll, Cristina Cachero Castro, Sven Casteleyn, Olga De Troyer, Piero Fraternali, Irene Garrigos, Franca Garzotto, Athula Ginige, Geert-Jan Houben, Nora Koch, Nathalie Moreno, Oscar Pastor, Paolo Paolini, Vicente Pelechano Ferragud, Gustavo Rossi, Daniel Schwabe, Massimo Tisi, Antonio Vallecillo, Kees van der Sluijs and Gefei Zhang
(Affiliations are shown at the end of the paper)

Abstract

Purpose – Ubiquitous web applications (UWA) are a new type of web applications which are accessed in various contexts, i.e. through different devices, by users with various interests, at anytime from anywhere around the globe. For such full-fledged, complex software systems, a methodologically sound engineering approach in terms of model-driven engineering (MDE) is crucial. Several modeling approaches have already been proposed that capture the ubiquitous nature of web applications, each of them having different origins, pursuing different goals and providing a pantheon of concepts. This paper aims to give an in-depth comparison of seven modeling approaches supporting the development of UWAs.

Design/methodology/approach – This methodology is conducted by applying a detailed set of evaluation criteria and by demonstrating its applicability on basis of an exemplary tourism web application. In particular, five commonly found ubiquitous scenarios are investigated, thus providing initial insight into the modeling concepts of each approach as well as to facilitate their comparability.

Findings – The results gained indicate that many modeling approaches lack a proper MDE foundation in terms of meta-models and tool support. The proposed modeling mechanisms for ubiquity are often limited, since they neither cover all relevant context factors in an explicit, self-contained, and extensible way, nor allow for a wide spectrum of extensible adaptation operations. The provided modeling concepts frequently do not allow dealing with all different parts of a web application in terms of its content, hypertext, and presentation levels as well as their structural and behavioral features. Finally, current modeling approaches do not reflect the crosscutting nature of ubiquity but rather intermingle context and adaptation issues with the core parts of a web application, thus hampering maintainability and extensibility.

Originality/value – Different from other surveys in the area of modeling web applications, this paper specifically considers modeling concepts for their ubiquitous nature, together with an investigation of available support for MDD in a comprehensive way, using a well-defined as well as fine-grained catalogue of more than 30 evaluation criteria.

Keywords Internet, Customization, Computer applications, User interfaces

Paper type General review

This work has been partly funded by the Austrian Federal Ministry for Education, Science, and Culture, and the European Social Fund (ESF) under grant 31.963/46-VIII/9/2002.

The authors would like to thank Andr as Sch onbeck and Alexander Knapp for their fruitful comments and contributions to prior versions of this work.

1. Introduction

Today's web applications are full-fledged, complex software systems for which a methodologically sound engineering approach is crucial. Web engineering has emerged as an independent branch of software engineering and "comprises the use of systematic and quantifiable approaches in order to accomplish the specification, implementation, operation, and maintenance of high quality web applications" (Kappel *et al.*, 2006). During the past ten years, academia has provided various web modeling approaches, each aiming at counteracting a technology-driven and *ad hoc* development of web applications. Most of these web modeling approaches originally have emerged as proprietary languages rather focused on notational aspects. As the types of web applications have evolved over time, the web modeling approaches come up with new appropriate concepts. Thus, increasingly more web modeling approaches are supporting the development of so-called Ubiquitous web Applications (UWAs), i.e. web applications that adhere to the anytime/anywhere/anymedia paradigm, taking into account that services in the web are nowadays not exclusively accessed through traditional desktop PCs but through mobile devices with different capabilities and constraints, by users with various interests and goals at anytime from anywhere around the globe. Services provided by UWAs, consequently, need to be adapted to the actual context of use in order to preserve or even enhance their semantic value for users. Thus, on the one hand, it is important that they capture the context, e.g. user, location, time, and device to understand the situation of use. On the other hand, and probably even more important is to provide appropriate adaptation operations to adjust the web applications with respect to their different kinds of contents, e.g. text, images, and links, as well as the offered navigation structure and presentation. Thus, context and adaptation in combination are the main prerequisites for customization of web applications towards ubiquity respecting the mapping of the required adaptation of an application's services with respect to its context.

Several modeling approaches have recognized this new requirement by providing new modeling concepts that capture customization functionality. Additionally, the rise of Model-Driven Engineering (MDE) already has had impact on current web modeling languages. Consequently, some of the approaches are supported with a modeling tool and possibly code generation facilities and have also aimed at providing for a model-driven development in the sense of MDE, i.e. on the basis of MDE techniques and technologies including metamodels and model transformations.

Based on previous work (Schauerhuber *et al.*, 2007a), this paper is dedicated to present the state-of-the-art in model-driven development of UWAs and throw some light on the current strengths and limitations of some of the most prominent modeling approaches. More specifically, an in-depth comparison of seven web modeling approaches currently supporting the development of UWAs is provided. This is conducted by applying, on the one hand, a detailed set of evaluation criteria and, on the other hand, by demonstrating their applicability with respect to a web application example in terms of a tourism web application. This includes five commonly found customization scenarios, thus providing initial insight into the concepts for modeling customization of each approach as well as to facilitate their comparability.

For this, in Section 2, the evaluation set-up is presented, i.e. the selection of web modeling approaches as well as a detailed and well-defined catalogue of evaluation

criteria used for the structured comparison of the approaches, including the description of the example and customization scenarios used in the comparison. Based on this, in Section 3, a per-approach evaluation is presented including a detailed description of the realization of the customization examples in the respective approaches. An extensive report on lessons learned is given in Section 4, pointing out the approaches' strengths and shortcomings. Thereafter, in Section 5, existing related work and the contributions of this survey are discussed. Finally, in Section 6, the paper is closed with a brief summary.

2. Evaluation set-up

This survey's goal is primarily to provide a study of existing web modeling languages having as specific focus the model-driven development of UWAs. In particular, the focus is on design level concepts for modeling customization as well as on tool support for the model-driven development of UWAs. As a consequence, other possible strengths of the approaches under investigation that lie beyond customization, e.g. support for workflow-based web applications, will not be considered herein.

2.1 Selection of approaches

With respect to the selection of evaluation candidates, in literature, several well-established and well-published web modeling languages having different origins and pursuing different goals can be found. In Schwinger and Koch (2006), a categorization of fourteen approaches into data-oriented, hypertext-oriented, object-oriented, and software-oriented web modeling approaches has been proposed (Figure 1).

Still, since focused on customization modeling, in this survey, only web modeling approaches that also provide concepts for customization modeling and thus enable to appropriately model UWAs will be considered. In the following, seven out of the fourteen approaches depicted in Figure 1 will be investigated being selected on basis of their relevance and their topicality. From the data-oriented category, the WebML (Ceri et al., 2003a) and the Hera (Frasincar et al., 2006) approaches, from the hypertext-oriented category, the WSDM (De Troyer and Leune, 1998) approach, and from the object-oriented category the OOHDM (Rossi and Schwabe, 2006), the UWE (Koch and Kraus, 2002), the OO-H (Gomez et al., 2001), and the OOWS (Pastor et al., 2006) approaches are evaluated. All of them can be considered to be well-established since each approach has been published in more than 25 articles including reviewed papers, books, and manuals.

2.2 Catalogue of evaluation criteria

In the following, a catalogue of criteria for the structured evaluation of web modeling approaches is proposed, having a particular focus on criteria for evaluating the support for customization modeling and for model-driven development. The criteria are, on the one hand, the result of a top-down approach considering the four dimensions of web application development (Schwinger and Koch, 2006) and on the other hand the result of a bottom-up approach taking into account interesting issues from related surveys as well as from previous work (Kappel et al., 2001; Schwinger and Koch, 2006).

The overall emphasis of the catalogue of criteria is on functional criteria. Since this survey is based on an in-depth study of literature of the approaches' documentation,

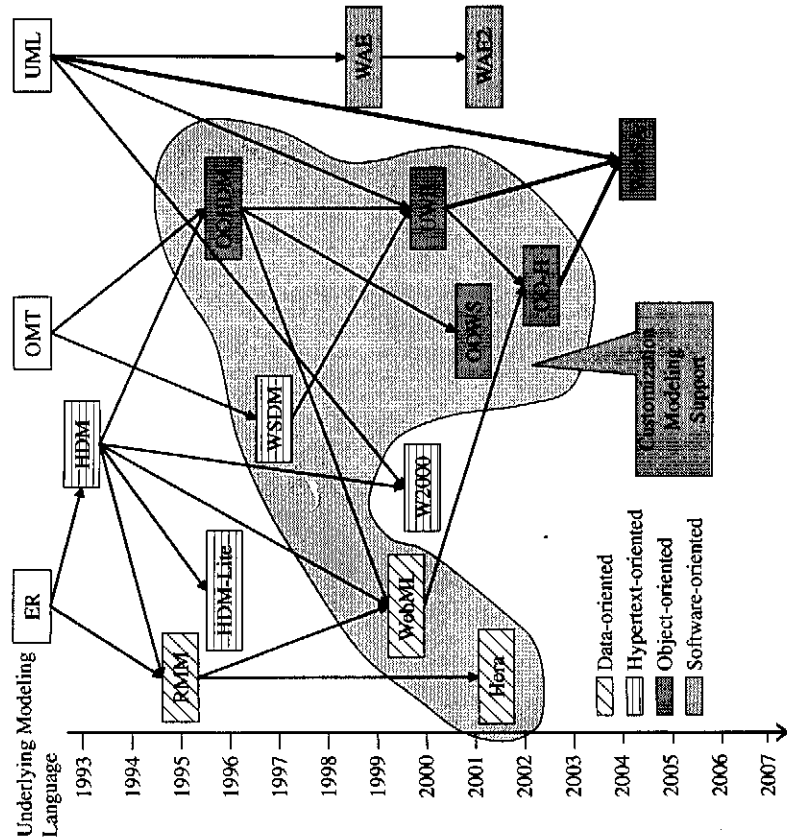


Figure 1. Overview on web modeling approaches

the inclusion of non-functional criteria in terms of several "-ilities" such as evolvability, scalability, traceability, reusability, understandability, maintainability, or flexibility is not considered within the scope of this evaluation. Still, this survey paves the ground for a later evaluation in real-world projects allowing to investigate the web modeling approaches with respect to the aforementioned "-ilities". Nevertheless, such an evaluation in real-world projects would also raise currently unaddressed questions associated with empirical evaluations in web engineering, e.g. how to get an unbiased set-up for the evaluation including control groups.

Aiming at a solid definition of criteria, for each criterion applied in this survey its name and a definition along with the appropriate measures are given which is somewhat similar to ISO/IEC 9126-1 (www.iso.org) for the evaluation of software quality characteristics. The abbreviation of the criterion allows for referencing it during evaluation of the approaches in Section 3.

Furthermore, these criteria are grouped into five categories with three of them, namely the criteria *web Modeling*, *Customization Modeling*, and *Model-Driven Engineering*, being inferred from corresponding functional requirements and two additional categories, one providing general criteria on the *Maturity* of an approach and the other one providing criteria related to the *Tool Support* of an approach. The catalogue of criteria is presented along with its categories in the following.

2.2.1 Maturity. The maturity of an approach is characterized by the following general criteria.

Topicality (G.T). This criterion provides for each approach the year of introduction as well as when the most recent piece of work has been published in order to indicate whether the approach is still under development or not.

Modeling Examples (G.ME). Another indication for the maturity of an approach is the number of different modeling examples discussed. Admittedly, besides evaluating the number of existing examples, their depth would also be of interest. Such a depth measure could be composed of the number of modeling concepts used, i.e. the number of content classes, nodes, links, etc. Still, this is not feasible, since often only parts of the examples are shown in available literature, or the examples have been simplified for readability purposes.

Application in Real-World Projects (G.A). Another indication for a high level of maturity of an approach is its employment in designing real-world applications. This criterion evaluates whether real-world applications exist or do not exist.

2.2.2 web Modeling. The *web Modeling* category covers criteria for evaluating the dimensions of web application development, namely levels, features, and phases. Note that, this category does not focus on the specific support for ubiquity terms of customization, for which criteria can be found in a separate category.

Web Application Levels (W.L). This criterion indicates which web application levels (i.e., content, hypertext, and presentation) are considered by an approach and which formalisms/types of diagrams are employed.

Interfaces (W.I). How the interrelationships between the web application levels are modeled, is indicated by the *Interfaces* criterion. In case that the interface specification is defined separately from the levels, this criterion names the mechanism used for content-hypertext and hypertext-presentation interfaces in terms of notations, natural language, or text-based query languages. If an interface specification is not defined separately but as part of one of the two levels in question, this criterion additionally evaluates to "intermingle".

Feature Modeling (W.F). For each web application level, this criterion investigates if modeling of structural and/or behavioral features of web applications are supported by the web modeling approach or not.

Development Phases (W.Ph). This criterion checks which phases of web application development, i.e. requirements elicitation, analysis, design, and implementation, are supported by the evaluated approaches.

Development Process (W.Py). The extent to which a developer is supported by a development process is covered by the *Development Process* criterion. Specifically, it distinguishes whether a well-defined development process is a proprietary one or is based on a standard development process, e.g. the Rational Unified Process (RUP) (Kruchten, 2000). Furthermore, it lists the detailed steps, output artifacts, and involved actors.

2.2.3 Customization modeling. The Customization Modeling category explicitly deals with characteristics of the customization dimension in web application development. This includes criteria investigating support for modeling context information as well as the necessary adaptations. The following criteria are based on previous work (Kappel et al., 2003) but specifically focus on the modeling level.

Context Properties (C.P). Although the relevant kind of context is specific to each UWA, a web modeling approach should support a set of common context properties including user, location, device, time, and network. Consequently, this criterion evaluates if the approach supports explicit concepts for modeling context and context properties, and what context properties have been used in modeling examples illustrating the approach.

Context Extensibility (C.CE). It is required that built-in modeling concepts for context properties can be easily extended by additional ones in case a UWA needs further context information (e.g. temperature). This criterion, thus evaluates to supported or not supported.

Chronology (C.C). This criterion tells if the approach offers (or does not offer) concepts that allow modeling how contextual information changes over time. For example, considering video streaming, information on how the bandwidth changed in the past can be used to infer how the bandwidth will develop or how stable it can be considered in the future in order to be able to tune the resolution of the video accordingly.

Complex Context (C.CC). Complex context information is aggregated using different context properties, e.g. "Vienna at night" which comprises the context properties of location being "Vienna" in composition with the context property time being "night". In this respect, this criterion evaluates if an approach supports or does not support appropriate modeling concepts to specify complex context.

Separation of Context (C.SC). Customization modeling should ensure the separation of context information and not just intermingle context with adaptation or the web application itself, i.e. usually the content model. This criterion tests whether an explicit representation, e.g. in terms of a separate context model, would allow for reusability of already defined context information across several UWAs.

Adaptation Operations (C.O). This criterion evaluates if the approach supports predefined modeling concepts for adaptation operations, e.g. filter some content, add links, change resolution of an image, change hypertext, etc.

Adaptation Extensibility (C.AE). Similar to the required extensibility of modeling concepts for context properties, this criterion evaluates whether adaptation operations can be extended by user-defined adaptation operations. The criterion evaluates to supported or not supported.

Complex Adaptations (C.CA). Complex adaptations define multiple adaptation operations performed on the same or on different subjects, e.g. adapting appearance for visually impaired persons may require resizing and color adaptation in combination. This criterion evaluates if an approach provides or does not provide appropriate means for modeling complex adaptations.

Adaptation of Levels (C.L). Customization influences potentially all levels of web applications. Content adaptation changes the information that is provided to the user by, e.g. adding or removing/filtering content, hypertext adaptation changes the navigation structure by, e.g. disabling a link or foreseeing different access structures, and presentation adaptation changes the way information is presented to the user by, e.g. changing colors or modality. Thus, this criterion investigates for which web application levels concepts to model adaptations are offered by an approach.

Adaptation of Interfaces (C.I). Customization may also influence the interfaces between levels, e.g. queries to underlying web application levels may need to change

due to a certain context. Consequently, this criterion evaluates if modeling of adaptations with respect to interfaces is supported or not supported.

Granularity (C.G). The granularity of adaptation indicates the number of modeling concepts affected by a certain adaptation. While micro adaptation is concerned with fine-grained adaptations by affecting a single application element only (e.g. disabling a specific link on a certain page), macro adaptation means that rather large parts of a model are adapted, thus affecting multiple modeling concepts (e.g. changing the language or changing the prevailing access structures).

Separation of Adaptation (C.SA). Customization modeling should ensure the separation of adaptation modeling and thus, prevent a representation which intermingles adaptations within the content, hypertext, and presentation levels. This criterion tests whether separation of adaptation is supported and thus, allows for reusability of already defined adaptations within the same or across several UWAs, or if it is not supported.

Customization Phases (C.CP). This criterion evaluates during which development phases an approach considers customization modeling. Ideally, customization is considered during all development phases.

2.2.4 Model-driven engineering. The *Model-Driven Engineering* criteria focus on modeling language definitions, model transformations, and platform descriptions as a prerequisite for successfully employing MDE in the style of OMG's Model-Driven Architecture (MDA) (OMG, 2003).

Language Definition (M.L). This criterion evaluates if a web modeling language has been defined explicitly in terms of a metamodel (including UML profiles), a grammar, a semantic description in terms of semantic web technologies, or if such a definition is absent.

Model Transformation Types (M.T). The investigated approaches might support or not support various types of model transformations such as transformations between platform-independent models (PIM2PIM), transformations between platform-independent and platform-specific models (PIM2PSM), transformations between platform-specific models and code (PSM2Code), and transformation from model transformation models to code directly (PIM2Code). For this, different kinds of model transformation languages, such as imperative, declarative, or hybrid ones, can be used.

Platform Description Model (M.P). This criterion evaluates if the information about a platform is represented separately within a platform description model, or if this information is implicitly captured within the transformation rules.

2.2.5 Tool support. For those approaches offering tool support, this category of criteria provides detailed information about the tool.

Tool Base (T.B). This criterion checks if the tool is developed as a stand-alone application or as a plug-in/extension to an existing tool.

Tool Openness (T.O). Whether the offered tool support can be tailored to the developer's needs or not, is tested by this criterion. Either the tool explicitly foresees extension possibilities or the tool's source code is publicly available under an open source license and thus can be changed.

Version (T.V). This criterion records the current version of the tool at the time of evaluation in order to give a rough estimation about the developed status of the tool in terms of revision cycles.

Costs (T.C). A tool might be free of charge or require a license fee to be paid. This criterion evaluates to either freeware or commercial.

Modeling Support (T.M). A tool supporting a specific approach can be evaluated if web modeling and/or customization modeling is supported or not supported.

Model Pre-Generation Support (T.MG). A tool might pre-generate (parts of) models to be refined later on. This criterion evaluates to true or false.

Consistency Check (T.CC). The criterion tests if a tool has a built-in consistency check to verify the correctness of the models or not.

Code Generation (T.CG). Tools may generate code from the models the developer has defined. The criterion evaluates to true, if a code generation facility is available, otherwise it evaluates to false.

Process Support (T.P). The kind of guidance through the development process within the tool is evaluated by this criterion. Firstly, it indicates whether the process supported by the tool is realized for the approach as described in literature. Secondly, this criterion details if the developer is bound to a step-wise procedure, or has the possibility to go back and forth between the development steps without loss of information (i.e., wizard-like), or can freely choose where to start and what to model as long as all phases are processed.

Collaboration (T.Co). The evaluated tool might support or might not support version control and thus, allow collaboratively working on a project in a team.

2.3 Modeling example: a tourism information web application

In order to support the textual comparison on the basis of a structured set of criteria, an example is provided, which is modeled by means of the concepts of each web modeling approach. The example further enhances the evaluation in that it first, provides an initial insight into each approach and second, facilitates the comparison of the approaches' modeling means in terms of their notation, especially with respect to modeling customization. It has to be emphasized, however, that the role of the modeling example in this survey is of a supportive nature, only. It is not intended to give a comprehensive introduction to all of a web modeling language's features. In particular, the example shall provide insight to the approaches' means for modeling customization functionality.

2.3.1 Running example. The web modeling field lacks commonly accepted reference modeling examples, which can be used to "assess" individual approaches, and in particular lacks modeling examples including comprehensive customization. Nevertheless, a first time this was done and with great success (in terms of impact on the methods) was at the International Workshop on Web-Oriented Software Technologies (IWOST 2001) (www.dsic.upv.es/~west/iwost01/) by introducing a Conference Management System. The second time, initiated at the Workshop on Model-driven web Engineering (MDWE 2005) (www.lcc.uma.es/~av/mdwe2005), proposed a Travel Agency. Most recently, (Rossi *et al.*, 2007) applies the example of a movie database to allow a comparison of different modeling approaches but not specifically focusing on customization issues. Furthermore, in the course of surveying the different web modeling approaches, it became obvious that some modeling examples have been used particularly often across several web modeling approaches, amongst them library systems, e-stores, and art galleries.

The main reason for not adopting one of these examples for this survey is that the modeled web applications typically are limited with respect to their ubiquitous nature. If at all, customization functionality is very simple, e.g. encompassing content adaptations with respect to an isolated context factor such as the user, the device, or the location, only. Instead, the aim of this survey was to choose an example from a domain with sufficient customization potential while possibly not creating any biases, e.g. due to reusing existing examples in some of the investigated approaches.

The domain of tourism information systems is able to provide such examples requiring customization, especially, when considering accessibility through mobile devices as is the case for mobile tourism guides (Grün *et al.*, 2006). Consequently, a hypothetical Tourism Information web Application (TIWA), which has been inspired by Tiscover (www.tiscover.at), the official Austrian tourism platform, has been chosen as the running example in this survey.

2.3.2 Basic functionalities. In the following, a short description of the requirements for our TIWA is given, briefly laying out the basic functionalities of a typical tourism information system which is necessary to be able to understand the modeling of the customization scenarios throughout this survey. The essential use cases are illustrated in Figure 2.

Hotels have Rooms, which in turn can have multiple Bookings, whereby each Booking belongs to a User. Hotels have additional Features such as a swimming pool, an animation team or the like. Every Hotel is situated in a Region which in turn has neighboring regions. In every Region there will be an offer of Activities, e.g. in terms of events.

In the course of this example, guest users, i.e. unauthenticated users, shall be enabled to browse and search for hotels, regions, activities as well as information about the weather. Registered users shall be allowed to book rooms and to browse their prior bookings. Administrators shall be responsible for the web application's content and therefore have to execute typical CRUD operations for hotels, regions, users, activities, and bookings.

How such a web application could look like, is indicated with the screenshot mock-up in Figure 3, showing a possible home page of the TIWA – based on the Tiscover Web-site – with dedicated areas for login, (search, and weather information. Furthermore, the menu allows navigating to, e.g. hotels as well as more detailed weather information.

Thus, the TIWA shall allow users to browse the offer of hotels either for special regions or as a result of the user's search. These lists of hotels will provide essential information, e.g. the hotel's name, a picture, a short description, a quick link for booking, etc. Whichever hotel the user picks leads to an according detail page, allowing the obtaining of all the information available for the chosen hotel as well as the possibility to start a booking process. Likewise, the user will be able to browse regions as well as activities and request more information on either subject by following links to the according detail pages. Furthermore, if already logged into the TIWA, the user is able to book a room and will be able to browse all prior bookings. Administrators have a special interface to create, edit or delete hotels, regions, activities, as well as bookings and users. Consequently, the content of the web application encompasses information on users, bookings, hotels and their special features, rooms, regions as well as activities.

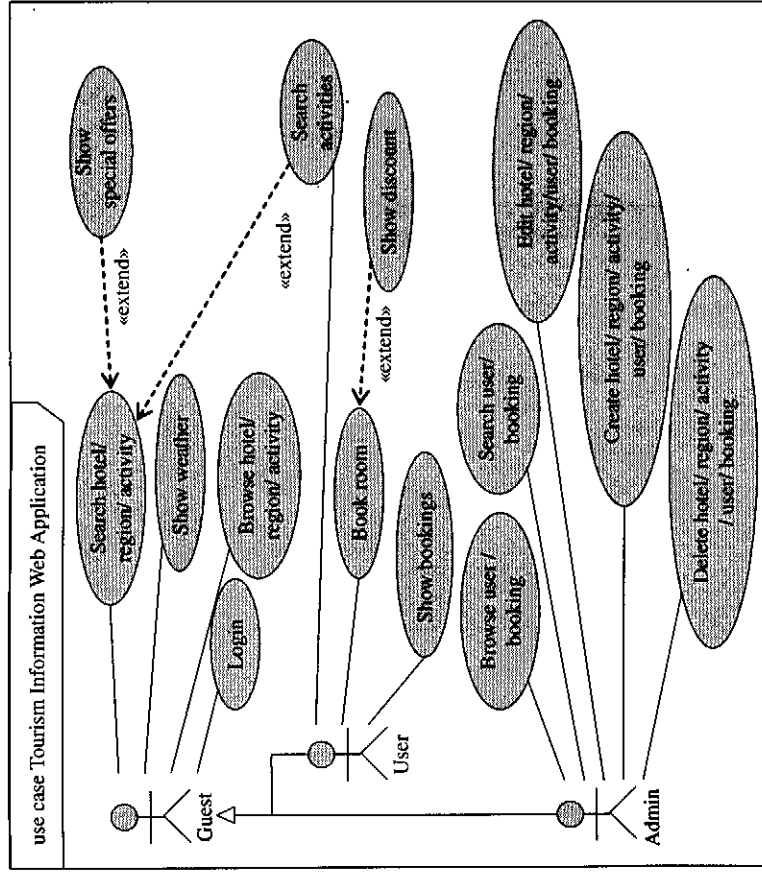


Figure 2.
Use cases of the tourism
information web
application

2.3.3 Customization scenarios. TIWAs naturally give space for a series of customization possibilities to better support user experience. In the realm of the TIWA, five customization scenarios which have been designed to support the catalogue of criteria, are laid out in the following as the supposed system's customization functionality.

- (1) **Customized Activities:** Users will have the possibility to view activities that are particularly relevant to their needs. The activities (C.L) made available to the user shall be filtered according to the user's age, location, the current time, and the weather at the user's location (C.P), (C.CE). Similar scenarios, representing personalization, could filter content according to the user's preferences, only, e.g. list all hotels according to the user's interest.
- (2) **Special Offers:** Users should enjoy special offers based on their navigational behavior (C.P). They shall be able to navigate to their special offers, if they have visited several pages of a specific region (C.CC). Thus, this scenario describes adaptation of the hypertext level (C.L). Alternatively, a user that has already booked three times will get 3 per cent off the price for the next reservation, which represents a content level adaptation (C.L).
- (3) **Administrator Links:** Administrators (C.P) shall be provided with "edit" links for every concept of the content they are allowed to edit (C.L), enabling for an



Figure 3.
The tiscover start page

(5) *Current Season's Style*: The look and feel of the web application shall be different according to the season. Different styles shall be chosen during summer and winter, respectively. The look & feel might also be subject to customization for supporting users having special needs, e.g. visually impaired users (C.L).

3. Comparison of approaches

In the following, the seven selected web modeling approaches will be compared applying the catalogue of criteria presented in Section 2.2. The description of each approach will be exemplified using the modeling example described above. Each of the following sections is dedicated to one approach. The approaches are ordered along the categories they belong to, starting with the category of data-oriented web modeling approaches. For a summary of the evaluation results, also in form of tables it is referred to Section 4.

3.1 The web modeling language (WebML), Ceri et al.

3.1.1 *Maturity*. WebML (Ceri et al., 2003a) is one of the well elaborated web modeling languages stemming from academia and is supported already over several years by the commercial tool WebRatio (www.webratio.com). Furthermore, WebML and WebRatio have already been successfully employed in several real-world projects (G.A). WebML is explained by a comprehensive number of different modeling examples including various forms of e-Stores selling different kinds of products, a conference management system, an online travel agency, a loan brokering web application, a museum guided tour, a campus tour, and an e-learning application (G.ME). Since its introduction in 1999, WebML has continuously evolved and recently has been extended by additional concepts addressing context-aware (Ceri et al., 2007a, b), service-enabled (Manolescu et al., 2005), workflow-based (Brambilla et al., 2006a), and semantic (Brambilla et al., 2006b) web applications as well as rich internet applications (Bozzon et al., 2006) (G.T).

3.1.2 *web modeling*. The WebML language as presented in Ceri et al. (2003a) provides modeling concepts for content modeling and hypertext modeling, only. While the content level resembles the well-known ER-model (Chen, 1976) (Figure 4(a)), at the hypertext level a proprietary graphical notation is provided (Figure 4(b)) (W.L).

Within the WebRatio tool, additional means for configuring the presentation level are available (e.g. for defining the web page's stylesheet and the positioning of information on a web page) which, however, are not part of the WebML language (Ceri et al., 2003a). The interface between content and hypertext level is specified graphically by denoting the entity of the content level (e.g. *Hotel* in Figure 4(a)) to be displayed in a so-called content unit (e.g. *Hotels* in Figure 4(b)) as well as in a textual representation to specify additional properties. In Figure 5, the textual representation of the *IndexUnit HotelList* of the *Hotels* page in Figure 4(b) is given.

Since specified at the hypertext level, however, the interface cannot be modeled separately (W.I). Additionally, the approach allows behavioral modeling to a limited extent, only. While UML activity diagrams are used during the requirements specification phase, some behavioral features are represented in the hypertext model by the control-flow-like semantics of WebML's operation units (*ModifyUnit* and *ConnectUnit* in Figure 6). Recently, the introduction of process modeling concepts into WebML (Brambilla et al., 2006), as well as the introduction of customization modeling

additional and possibly easier way of content management. For normal users or visitors, these links will not be available requiring fine-grained hypertext adaptation (C.G).

(4) *Multi-Delivery*: In order to consider users equipped with small devices (C.P), an appropriate adaptation of the system shall be chosen. If a small-screen device is used, pictures and detailed descriptions shall not be shown. This can be achieved by creating a dedicated hypertext model for each device type which represents a coarse-grained hypertext adaptation (C.G) or by omitting content for devices with small displays, which represents a fine-grained content adaptation (C.L).

concepts (Ceri et al., 2007a), allows further modeling of some sort of behavioral aspects at the hypertext level (W.F). The WebML approach includes its own seven-phase, iterative, and incremental development process based on Boehm's Spiral model (Ceri et al., 2003b) (W.Pr) comprising all development phases from requirements to implementation (W.Ph).

3.1.3 Customization modeling. According to available literature, the WebML approach provides two proposals for modeling customization. The first approach introduces context-aware pages (Ceri et al., 2007a), whereas the second approach is based on event-condition-action-rules (Ceri et al., 2006) for which it is, however, not clear how they are related and how they can be used together. Both approaches have in common that they do not provide an explicit context model, but represent context information within the content model.

In the first proposal for dealing with UWAs (Ceri et al., 2007a), however, the modeler is supported in designing the context information with guidelines proposing to imagine the content level as a set of overlapping sub-schemas (C.SC). For illustration purposes, these sub-schemas can be indicated in the content model (Figure 4(a)). The entities of the, so-called, *Basic User Sub-Schema* are always available in a WebML model and comprise User, Group, and Module entities. The so-called *Personalization Sub-Schema* associates the User entity with other entities to denote, e.g. user preferences, whereas the so-called *Context Model Sub-Schema* associates the User entity with other context information, e.g. the user's location, the device used, etc. (C.CE). As a form of static adaptation, WebML siteviews, i.e. several hypertext models defined upon the content level, on the one hand, may be used to personalize a web application according to users

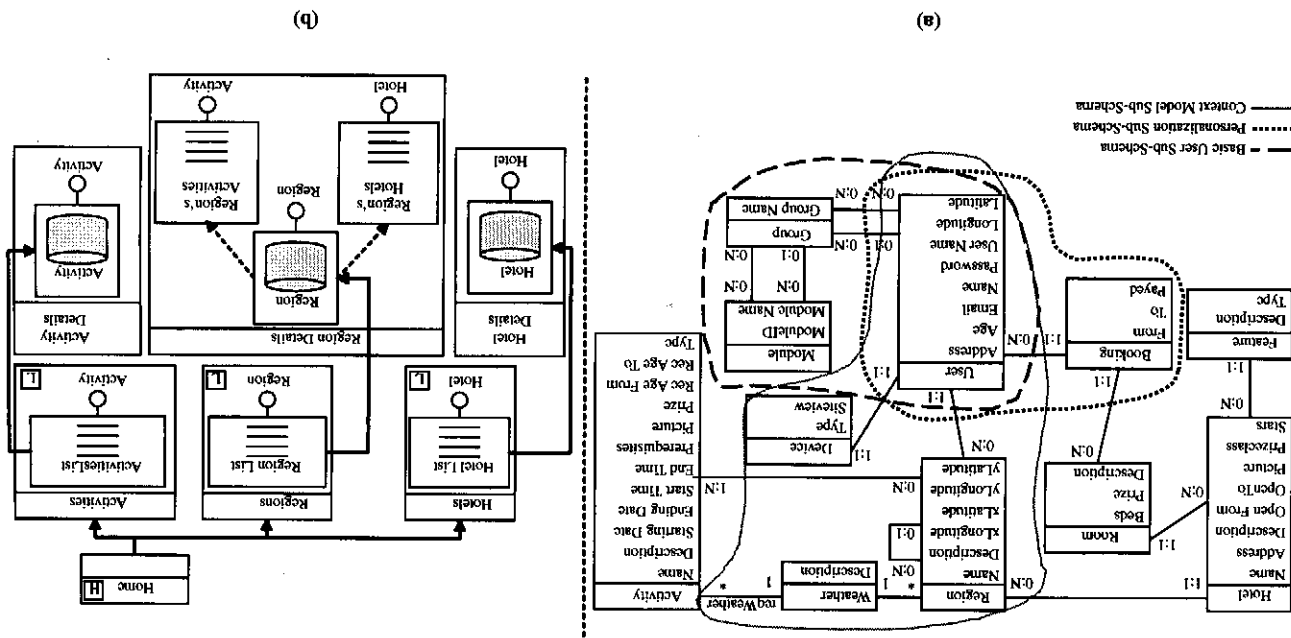


Figure 4. WebML: (a) content level, (b) hypertext level

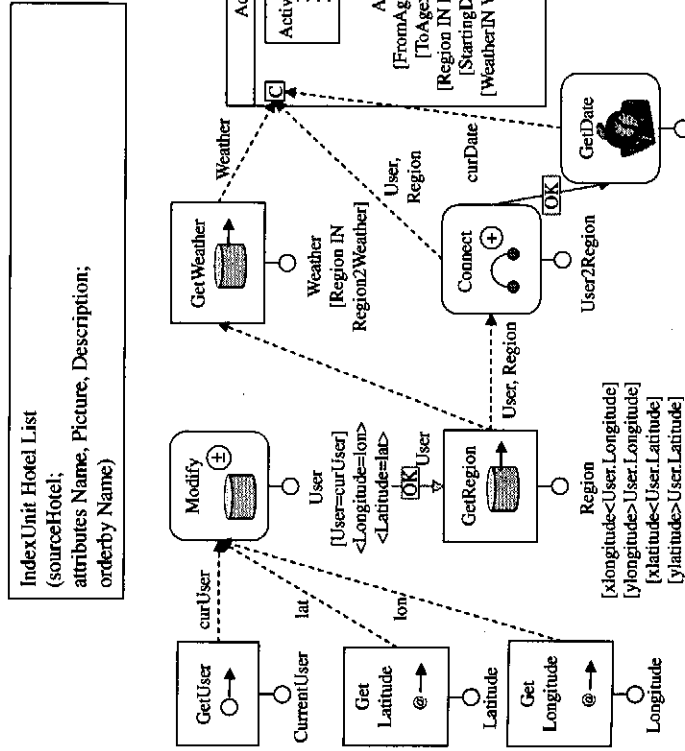


Figure 5. Listing 1: WebML textual representation of the IndexUnit HotelList

Figure 6. WebML: customized activities scenario

and user groups and, on the other hand, serve the purpose of expressing alternative forms of content presentation for different devices (Ceri *et al.*, 2003b) (C.P). Modeling complex context is not explicitly supported within WebML (C.CC). For modeling context-awareness, the idea of a context-aware web page (Activities Page in Figure 6) and concepts for retrieving context information have been introduced. For example, for retrieving context information from the content level, the *GetDataUnit* concept is employed (*GetRegion* *GetDataUnit* Figure 6), while the *GetClientParUnit* concept is used to obtain context information from the client (*GetLatitude* *GetClientParUnit* Figure 6), e.g. information on the user's location or on wireless connectivity (Ceri *et al.*, 2007a) (C.P). For modeling adaptation, two predefined adaptation operations are available, namely *ChangeSiteview* (Figure 8(a)) and *ChangeStyle* (Figure 8(b)) (C.O), which allow for a coarse-grained adaptation of the hypertext and the presentation level, respectively, (C.L), (C.G). The possibility of changing the navigation flow, e.g. through WebML's *IfUnit* (Figure 7(b)), enables more fine-grained adaptations. There is no way to explicitly adapt the interface between the content and the hypertext level (C.I), since adaptations are defined for a page rather than for a content unit. It is not possible to extend the predefined set of adaptation operations (C.AE), neither is there a concept for defining complex adaptations (C.CA). In WebML, UWAs are modeled as refinements of the models of a non-ubiquitous web application, i.e. the development process has not yet been adapted to guide developers in considering customization functionality throughout the development lifecycle (C.CP). Since adaptations are modeled as an extension of the hypertext model, WebML does not provide a separation of adaptations from the rest of the web application model (C.SA).

In the second proposal for modeling UWAs with WebML, the focus is on personalization purposes and allows defining adaptive behaviour of a web application based on event-condition-action (ECA) rules (Ceri *et al.*, 2006). The approach relies on adaptive pages that specify the action part of the rule and are similar to context-aware pages. Conditions are described on the basis of a so-called web behavior model (WBM) script, which is a timed state-transition automaton for representing classes of user behaviors on the Web, i.e. navigation patterns. Thus, some form of context chronology is supported (C.C) for this approach.

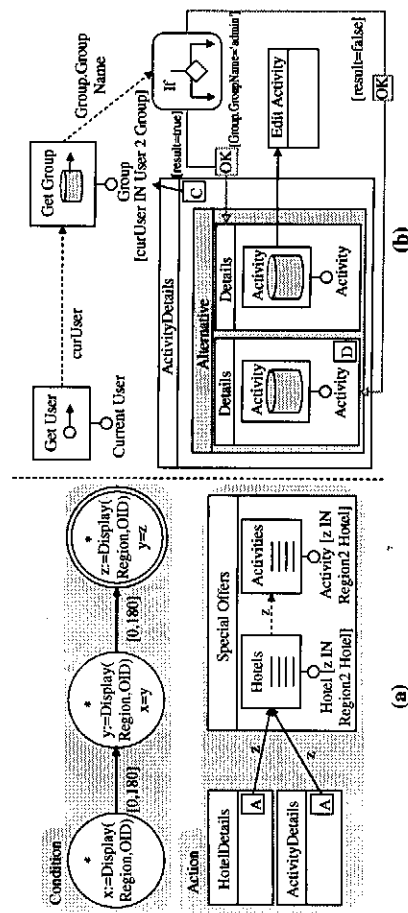


Figure 7.
WebML: (a) special offers scenario, (b) administrator links scenario

3.1.4 *Model-driven engineering.* WebML's language concepts are partly defined in terms of XML document type definitions (DTDs) (ML) and partly hard-coded within the approach's accompanying tool, WebRatio. Consequently, the WebML approach currently cannot profit from the benefits of MDE as defined herein. Nevertheless, two proposals have recently invested some efforts to port WebML to MDE. One of them provides a metamodel for WebML based on the Meta Object Facility (MOF) (Schauerhuber *et al.*, 2007b) and another effort provides a UML profile for WebML (Moreno *et al.*, 2007a). While the WebRatio tool provides code generation support (using J2EE and Jakarta Struts) from the platform-independent WebML models, the WebML approach does not provide for model transformations to different platforms prior to generating code (M.T). As a consequence, the approach does not provide platform description models for the above mentioned platforms (M.P).

3.1.5 *Tool support.* WebRatio is a commercial, proprietary tool developed as a standalone application (T.C), (T.O), (T.B). WebRatio 4.3 has been evaluated with an academic license (T.V) which is free of charge (T.C). Currently, WebRatio does not include the concepts for modeling customization, except for providing the User, Group, and Module entities in every web application's content model. Still, it is reported that the concepts proposed in Ceri *et al.* (2007a, b) have been realized in prototype implementations (T.M). The tool allows extending the WebML language via specialized content units and operation units (T.O). This way, the WebML language and tool support have already been extended, e.g. to support service-enabled Web applications. WebRatio allows generating a running application out of a Web application model, with no additional coding. The generated web application can be automatically deployed to an integrated Tomcat Servlet Container (T.CG). Moreover, the tool offers a so-called pattern wizard that allows building common parts of the hypertext model automatically, like a login mechanism or content management functionality for selected entities from the content level (T.MG). The tool also allows consistency checks that can be executed on-demand and an additional approach for a formal verification has been proposed (Brambilla *et al.*, 2007) (T.CC). The process of the method is only partly supported by the tool, e.g. requirements engineering is not considered, and the user is allowed to start wherever s/he chooses (T.P). Cooperative work is supported in that the WebRatio tool is also capable of shared editing via a concurrent versioning system (CVS) (T.Co).

3.1.6 *Modeling example.* In Figure 4(a) the content level, i.e. an ER-diagram, of the TIWA example is shown for WebML. In WebRatio, the predefined entities User, Group, and Module enable basic personalization of the hypertext level towards user and user groups (the Basic Sub-Schema indicated in Figure 4(a)). The other entities are representing the web application's data, whereby each Booking belongs to a User (the Personalization Sub-Schema indicated in Figure 4(a)).

Further context information has been incorporated to the content level as follows: The User is directly related to the device s/he is using. Additionally, the User is also directly associated with the Region s/he is currently located as well as indirectly to the Weather entity that has been introduced to allow for customization according to the current weather situation (the Context Model Sub-Schema indicated in Figure 4(a)).

Part of the hypertext level of the TIWA is shown in Figure 4(b). This particular view shows that users can browse hotels, regions, and activities via three dedicated landmark pages ("L" label). The specific subjects can then be selected via WebML's

IndexUnits to be displayed in detail on separate pages, i.e. *HotelDetails*, *RegionDetails*, and *ActivityDetails*. Each of these pages uses a *DataUnit* for presenting the selected item. In addition, the *RegionDetails* page presents an index of the region's activities and hotels, respectively.

Customization Scenario Customized Activities. The *Activities* page can be made context-aware in order to show only activities of the region the user is currently located in and are employed to filter them according to the user's age and the current date (cf. first approach of WebML to address UWAs). A *GetUnit GetUser* acquires the current user and two *GetClientParUnits* provide the user's longitude and latitude (cf. Figure 6). The location information then can be stored for the current user in the content model with a *ModifyUnit*. The location information is also used to select the region the user is currently located with the *GetRegion GetDataUnit*. The *ConnectUnit* is used to relate the user with the current region, i.e. the user's location. A *TimeUnit* which is available also in WebRatio, can be used to obtain the current date so that only those activities are shown that have not yet started. The current weather situation for the user's region is assumed to be updated in the data source by some external service so that it can be queried by the *GetDataUnit GetWeather*. In order to correctly filter the information on activities for the user, all this context information is then used in a set of so-called *SelectorConditions* (square brackets) for the *ActivitiesList IndexUnit* in the page *Activities*.

Customization Scenario Special Offers. For modeling the Special Offers customization scenario, WebML's rule-based approach (second approach for modeling UWAs with WebML) is employed, which allows reacting to user navigation. In the following example, the TIWA will monitor the user navigation to find out what region a user is particularly interested in. As soon as this information is available, the user shall be redirected from the *HotelDetails* as well as from the *ActivityDetails* page to a page presenting special offers for hotels and activities. As depicted in Figure 7(a), the condition of this rule is specified by a WBM script, i.e. a timed finite state automaton, stating that a user needs to visit three arbitrary pages (denoted with "x"), having in common that each displays information about the same region. In a WBM script, a state (denoted with a circle) represents the user's inspection of a page. The three pages do not necessarily have to be visited in sequence, i.e. the user can navigate to other pages in between. Still, the transition constraint states that after 180 seconds, the next page displaying the same region information has to be navigated otherwise the script will be discarded. In case the third page is visited, the automaton reaches its accepting state, triggering the action of the rule. In the action part of the rule, the *HotelDetails* and the *ActivityDetails* pages become adaptive pages (A), having a link to the Special Offers page, which will be navigated in case the rule's condition is true.

Customization Scenario Administrator Links. In general, WebML's siteviews can be used for modeling different hypertexts for different user groups, e.g. external users and administrators. To enable editing from the public siteview, however, fine-grained adaptation shall allow/disallow dedicated links in the *ActivityDetails* page of the public siteview, in the following. In Figure 7(b), depending on the current user's group, the *ActivityDetails* page will present different navigation possibilities. The *Alternative* concept is employed to indicate that the normal *Activity DataUnit* is displayed in the default case ("D" label). Alternatively, an *Activity DataUnit* that has a link to an

EditActivity page can be displayed. An *IfUnit* is employed to redirect navigation to the respective alternative on the basis of the context retrieved by the *GetUser GetUnit* and the *GetGroup GetDataUnit*.

Customization Scenario Multi-Delivery. WebML also suggests modeling dedicated siteviews for special navigation purposes as is the case for adapting the hypertext level according to the user's device. The *ChangeSite ViewUnit*, used in this scenario, has been recently introduced to the WebML language to allow for changing the hypertext model to better suit the current context of the user. In Figure 8(a), the *PublicSiteview* has been made context-aware. Thus, if a user requests any page located in this siteview, the *GetClientParUnit GetDevice* will provide the necessary information to find a device specification in the database via the *GetDataUnit GetDeviceSpec*. The user will then be redirected to an appropriate siteview that has been associated to the device in the database by the *ChangeSiteviewUnit*. In case no such specification can be found, the user will be redirected to a page where the device can be chosen from a predefined list.

Customization Scenario Season's Style. Finally, WebML does also allow customizing the presentation level with the recently introduced *ChangeStyleUnit*. This coarse-grained adaptation operation changes the look&feel of the web application by changing the Cascading Style Sheet (www.w3.org/Style/CSS) (CSS) used according to the current context. Figure 8(b), shows how the time context is obtained with a *TimeUnit GetDate*. An *IfUnit* is used to decide if the presentation of the home page shall be based on the summer style or the winter style by using the respective stylesheets.

3.2 The Hera approach, Houben et al.

3.2.1 Maturity. The Hera (www.wis.win.tue.nl/hera/) approach (Frasincar et al., 2006), first introduced in 2000, has been inspired by the hypermedia design method Relationship Management Methodology (RMM) (Isakowitz et al., 1995) and originally concentrated on providing web access to data by automatically generating static hypermedia presentations (Barna et al., 2002). Later, revisions of the Hera approach have been introduced, for example the Java-based version HPG-Java (Barna et al., 2006) and the latest one Hera-S (van der Sluijs et al., 2006, Rossi et al., 2007 (Chapter 10)), which consider dynamic web applications (G.T). The prevailing modeling example used in almost all publications is a virtual art gallery. Besides that, a conference management system, a digital photo library, a movie library, and e-store applications are used as modeling examples (G.ME). While it seems that real-world applications of

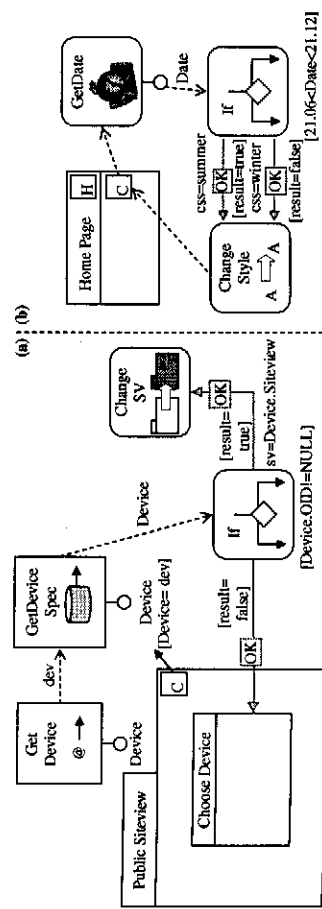
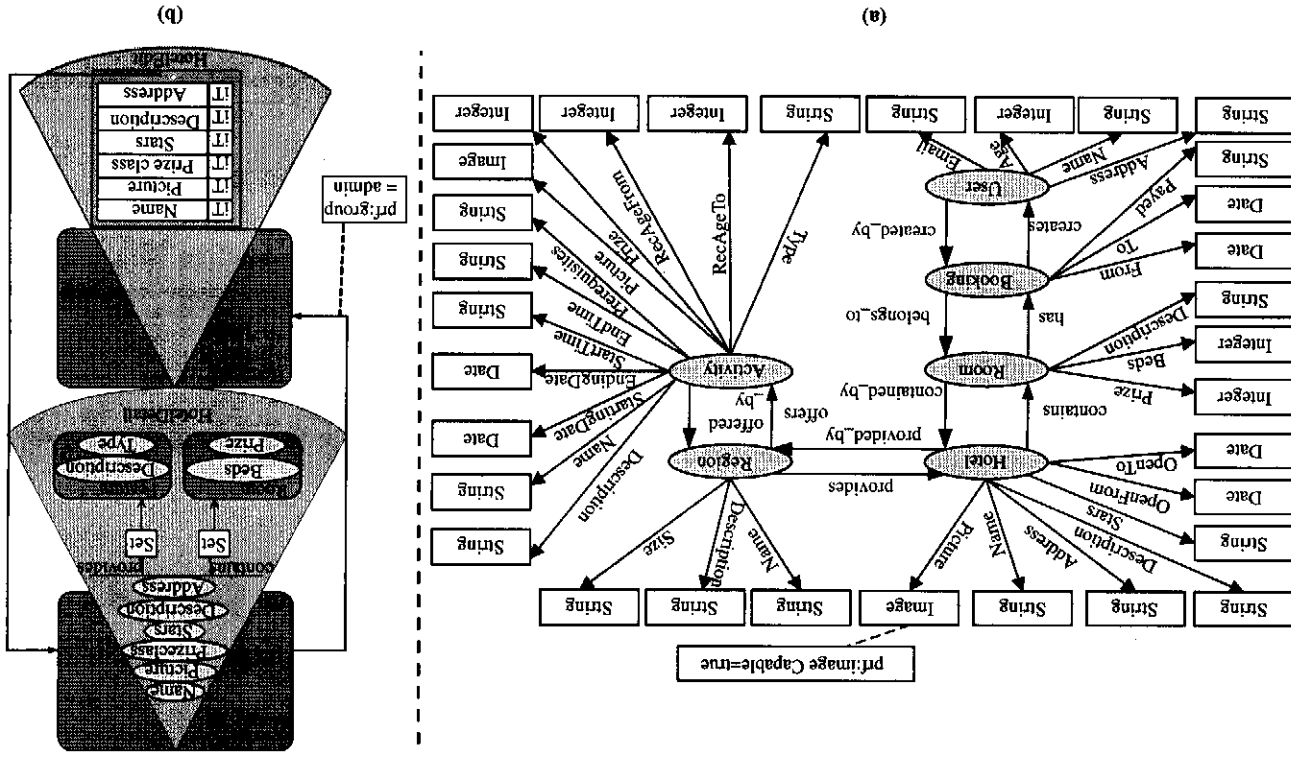


Figure 8.
WebML: (a) Multi-delivery scenario, (b) season's style scenario

Hera do not exist yet, four demo web applications have been developed including the museum application where real-world content has been reused (G.A). We do note that the majority of this survey considers the first version of Hera (and its tool HPG (HPG-XSLT)).

3.2.2 web modeling. The Hera approach, in principle, proposes models for the content, hypertext, and presentation levels (WL). Since RMM has been a major influence to Hera (Frasincar *et al.*, 2001), the content level (i.e. Hera's conceptual model or alternatively called domain model) initially has been based on the ER-model (Chen, 1976). Today, Hera is based on the Resource Description Framework (Schema) - RDF(S) (W3C, 2004b), thus supporting the engineering of semantic web information systems. For modeling the content level, it uses plain RDF(S) which means that all available tools for (textual) editing of RDF(S) are applicable; next to that the custom editing tools use a proprietary graphical notation (Frasincar *et al.*, 2006). At content level, as depicted in Figure 9(a), the domain model of Hera is based on concepts (e.g. Hotel and Region), attributes (e.g. Name and Picture), concept relationships, (e.g. provided by and contains) and media types (e.g. String and Image). The hypertext level (i.e., the application model) and the presentation level (i.e., the presentation model) still resemble RMM's graphical notation as is shown in Figure 9(b) and 11, respectively. More specifically, the Hera application model addressing the hypertext level is mainly based on slices and slice relationships, which can be either compositional or navigational. In the upper part of Figure 9(b), the *HotelDetail* slice has *Hotel* as the owning concept from the content level (the rounded rectangle) and presents the details of a hotel, including its name, picture, priceclass, stars, description and address. Furthermore, the set of rooms it contains and the features it provides are part of the slice. From the *HotelDetail* slice, a navigational relationship leads to the *HotelEdit* slice. In the different versions of Hera there are different ways to consider the presentation. In one of them, the presentation level is based on a hierarchy of regions allowing the positioning of slices from the hypertext level (Fiala *et al.*, 2004b). A region represents a rectangular part of the display area and has associated with it a layout manager for positioning elements and a specific style. Figure 11 describes how the *HotelDetail* slice and its parts are placed within different regions. For example, the *HotelDetail* slice is placed within a *Box Layout*. The figures show that the interfaces between levels are specified graphically. For example, at hypertext level, a slice always denotes its owning concept from the content level in terms of a rounded rectangular (Figure 9(b)). Moreover, the interfaces between the levels are as well specified textually on the basis of queries (van der Sluijs *et al.*, 2006). Still, the interface is always intermingled with the upper level (W.I). The later introduction of concepts for modeling workflow-based web applications (e.g. a task model and a workflow model) (Barna *et al.*, 2006) allow for modeling behavioral aspects at the hypertext level (W.F). The Hera approach assumes a development process that includes development phases from requirements engineering to implementation (Vdovjak and Houben, 2002) (W.Pr), however, this process has not been detailed beyond a set of guidelines. Recent elaborations on the approach focus particularly on design and implementation, but do not consider the requirements engineering phase (W.Ph).

3.2.3 Customization modeling. In the Hera approach, customization can be supported as long as it is expressible in RDF(S) and its querying mechanism (which allows for a large variety to be supported). Most of the examples focus, as illustration



of that, on user and device context (C.P). The approach strives for encapsulating context information separately (C.SC): On the one hand, (in the early versions of Hera) static adaptations are based on context information obtained from a user profile (Frasincar *et al.*, 2004), (Fiala *et al.*, 2004a). To illustrate, Hera can use a CC/PP vocabulary (W3C, 2004a) to model user preferences and device capabilities. For example, in the profile instance presented in Listing 2, the hardware platform is characterized by the property client, which identifies the device used, while the group property characterizes the user preferences in terms of providing information on the user's group. On the other hand, dynamic adaptations, that are based on a user model storing the users' navigational behavior, have been proposed for the early versions in Frasincar and Houben (2002) and have been detailed for Hera-S in van der Sluijs *et al.* (2006). Moreover, further dedicated concepts are absent for supporting extensibility of context explicitly (C.CE), for capturing context history (C.C), and for modeling complex context information (C.CC). Adaptation (in the original HPG) is realized by means of so-called appearance conditions attached to different design artifacts, i.e. to the content, hypertext, and presentation models (C.L), (C.D). If such conditions evaluate to true or false, the presence of their associated artifacts, e.g. content information or links (C.C), is enabled or inhibited, respectively, (Frasincar *et al.*, 2004). In Hera-S, the extensive use of SeRQL expressions allows for extensibility and complexity in the adaptation operations. In this respect, appearance conditions can be seen as a form of adaptation operations (C.O). As a consequence, the approach does neither allow for extensibility of adaptation operations nor for complex adaptations (C.AE), (C.CA). Since adaptations are only annotations to different design models (C.CP), they cannot be modeled separately either (C.SA). Still, in the Hera-S approach, the recent introduction of concepts from the aspect-orientation paradigm is intended to provide for such a separation (Casteleyn *et al.*, 2007). This way, appearance conditions can be modeled separately and "woven" into the models before generating the web application. Another proposal for customization presents how a high-level personalization rule language (Garrigos *et al.*, 2005b), can be mapped onto modeling concepts of Hera and OO-H (Section 3.6).

3.2.4 Model-driven engineering. As already mentioned before, the Hera modeling language relies on RDF(S) and CC/PP, respectively, (M.L). The original approach focused on the generation of static hypermedia presentations, i.e. static web sites, in a specific output format, e.g. in terms of HTML and WML, which are deployable on any web server. For this, the approach suggests the subsequent transformation or rather the integration of the different design model artifacts, i.e. Hera's conceptual model, application model, presentation model, and user profile model, whereby in a last step, the output can be generated (M.T). The current Hera-S does not have that pipeline as such. The approach does not support a separate platform model and allows with its RDF(S)-based nature to specify what is needed (M.P).

3.2.5 Tool support. Hera is supported by the Hera Presentation Generator (HPG) tool, which has been evaluated in version 1.3 (T.V). HPG is a proprietary tool, not extensible, and available as freeware (T.B), (T.O), (T.C). Support for a concrete notation for modeling the content and the hypertext as well as the presentation level has been realized by three Microsoft Visio (<http://office.microsoft.com/en-us/visio/>) templates called *Conceptual Model Builder Application Model Builder*, and *Presentation Model Builder*, respectively, (Frasincar *et al.*, 2006). A load/export feature provides the RDF(S)

serialization of the models for the HPG tool. Apart from that, modeling of the user profile, i.e. the CC/PP code, is supported through appropriate "textual" wizards of HPG (T.M). Since HPG 1.3 generates static hypermedia presentations only, it does not allow for modeling dynamic adaptations that consider, e.g. the user's inputs or the user's browsing history (Frasincar and Houben, 2002). Today, transforming the models into a suitable static presentation (HTML, WML, etc.) is possible on the basis of XSL transformations using either HPG itself or the external AMACONT engine developed at the Dresden University of Technology (Fiala *et al.*, 2004a) (T.MG), (T.CG). The HPG tool follows the guidelines for developing a web application with Hera by starting with the content level, i.e. Hera's domain model. A wizard guides the user through the design and generation process (T.P). In every step, the models can be viewed using a text editor or Microsoft Visio, where the model builders enforce some constraints while the user is designing a model. Moreover, the models' consistency can also be checked by HPG (T.CC). Up to now, there is no shared editing or versioning support (T.Co). Later, modeling support for user input and tool support for generating dynamic web applications, which are able to react to user input, has been proposed in Houben *et al.* (2004) and Frasincar *et al.* (2006), respectively. That tool support, HPG-Java, provides a runtime environment for the generated applications on the basis of the Java-Servlet platform. The latest version of the approach Hera-S and its accompanying tool, called HydraGen, combines Java with the functionality of the Sesame environment for storing and retrieving RDF data.

3.2.6 Modeling example. In the Hera approach, here again considered in its first version, not all customization scenarios can be realized if one is able to express them using RDF(S) and SeRQL, which implies a large expressive power. Since the approach is limited to user and device context information as well as to appearance conditions which do not allow filtering of content, it is possible to present the Multi-Delivery and the Administrator Links scenarios, only.

Customization Scenario Multi-Delivery. The content level as well as the other levels can be customized with annotations, i.e. appearance conditions such as *pr:imageCapable = true* in Figure 9(a).

This appearance condition is specified on the basis of the user profile instance of Listing 2 and means that the conceptual model provides a picture of the hotel only if the device used is capable of presenting images. Since, according to the client property in the profile instance, the *imageCapable* property is set to true, the hotel's picture will remain in the domain model and thus in the generated hypermedia presentation.

With respect to the presentation level, appearance conditions can also be used to choose one of a set of alternative regions according to the current device. In Figure 11, the set of rooms of the *HotelDetail* slice are placed at the bottom part of the display (cf. the regions *RegionBottomRooms*). More specifically, two alternatives for presenting the hotel's set of rooms are provided depending on the device used.

The appearance conditions associated with the two alternative regions *RegionBottomRooms* state, that in case of a PDA, the rooms also shall be displayed according to a *Box Layout*, while when using a PC, users shall be presented with a *Table Layout*.

Customization Scenario Administrator Links. Similarly, appearance conditions can be used to customize the hypertext level, e.g. a link in the Hera application model can be adapted according to the user profile. Figure 9(b) presents a small excerpt from the

application model showing how the Administrator Links scenario can be realized. The link from the *HotelDetail* slice to the *HotelEdit* slice is annotated with the appearance condition *prf:group = admin* stating that only users of the group "admin" will be able to see this link. According to the group property in the profile instance presented in Listing 2 (Figure 10), the user is part of the admin group and thus, the edit link will be available. Another way of hypertext adaptation is the possibility to define multiple application models for the content level. In this respect, the application models or rather the RDF(S) specifications just need to be exchanged and the web application can then be re-generated. Please note that appearance conditions also can be associated with a whole slice or one of its shown attributes, e.g. the hotel's description (Figure 11).

Following, Listing 3 indicates how such an appearance condition can be modeled separately as is proposed in Casteleyn *et al.* (2007) using the textual Semantics-based Aspect-oriented Adaptation Language (SEAL) of Hera-S (Figure 12). In the example below, the *ADVICE* specifies that an appearance condition needs to be added to the hypertext level, while the *POINTCUT* specifies the exact place(s). Please note, that in Hera-S the modeling concepts of the language partly have been renamed, e.g. unit corresponds to a slice and the reference to the user profile "prf" has been changed to "cm". Consequently, the *POINTCUT* specifies the link from the *HotelDetail* slice to the *HotelEdit* slice. The advantage of this approach is that within the *POINTCUT*, several links of the hypertext level can be specified in one place and extended with an appearance condition.

3.3 The web site design method (WSDM), De Troyer et al.

3.3.1 Maturity. The web Site Design Method (<http://w3dm.vub.ac.be/Research/publications.php>) (WSDM), later re-baptized to web Semantics Design Method (De Troyer *et al.*, 2007), has been first proposed in 1998 (De Troyer and Leune, 1998) and thus, is one of the earliest web modeling approaches. Since its introduction, WSDM has continuously evolved and has been extended by additional concepts addressing localization (De Troyer and Casteleyn, 2004), adaptivity (Casteleyn *et al.*, 2005), accessibility (Plessers *et al.*, 2005a), and semantic annotations (Casteleyn *et al.*, 2006). Recently, WSDM is evolving towards the semantic web (Casteleyn *et al.*, 2006), (De Troyer *et al.*, 2007) (G.T). The method is used extensively in student projects (<http://wise.vub.ac.be/courses/designmethods.htm>) but there is no report on applications of

```
<Description rdf:about="Profile" >
<ccpp:component>
  <HardwarePlatform>
    <imageCapable>true</imageCapable>
    <client>Desktop</client>
  </HardwarePlatform>
</ccpp:component>
<ccpp:component>
  <UserPreferences>
    <group>admin</group>
    ...
  </UserPreferences>
</ccpp:component>
</Description>
```

Figure 10.
Listing 2 – Hera: user profile instance

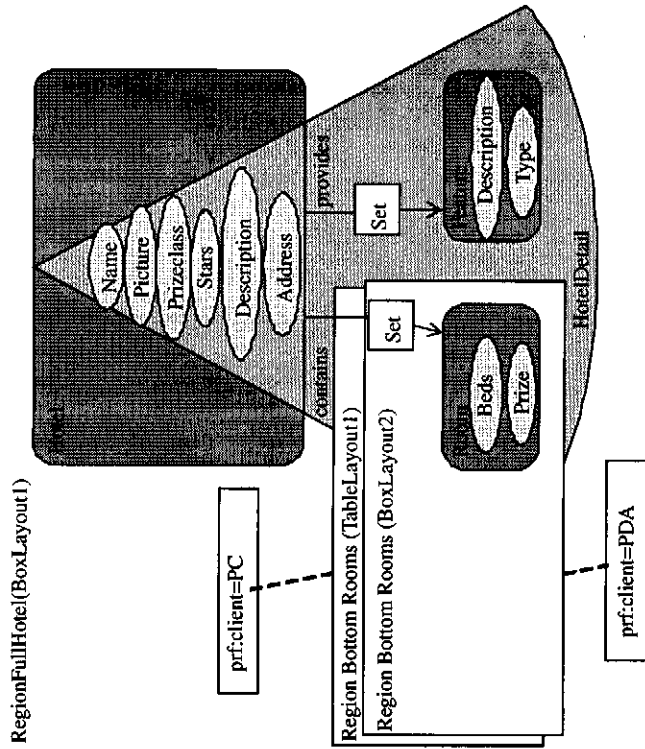


Figure 11.
Hera: presentation model

```
POINTCUT type relationship and
from (type unit and hasName "HotelDetail") and
to (type unit and hasName "HotelEdit")
ADVICE ADD CONDITION cm: group = "admin"
```

Figure 12.
Listing 3 – Hera: separating appearance conditions with aspects

the approach in real-world projects (G.A). Modeling examples used are a conference management system, an e-store, a movie library and a department website (G.ME). **3.3.2 web modeling.** WSDM provides its own five-phase, audience-driven development process (W.Pr) and specifies all its modeling concepts within the WSDM Ontology (<http://wise.vub.ac.be/ontologies/WSDMOntology.owl>) which is based on the web Ontology Language (OWL) (W3C, 2004c). The WSDM ontology is thus a meta-model for WSDM (ML). The process starts from a mission statement, specifying purpose, subject and targeted users, followed by a user-driven requirements engineering and analysis phase, which results in the audience model, i.e. a set of audience classes, each describing a particular group of users and their specific requirements and characteristics. During the (conceptual) design phase, structural modeling of all web application levels is supported (W.L). The content level is represented as the integration of object chunks to the so-called business information model. Each object chunk represents a tiny conceptual schema, which is derived from the tasks each audience class needs to be able to perform. In turn, these tasks are identified in a task modeling activity based on the Concurrent Task Tree (CTT) technique (Paterno, 2000), whereby a task is decomposed into elementary tasks arranged in a temporal order (W.F) (Figure 13). Then, for each elementary task an object chunk is created, which

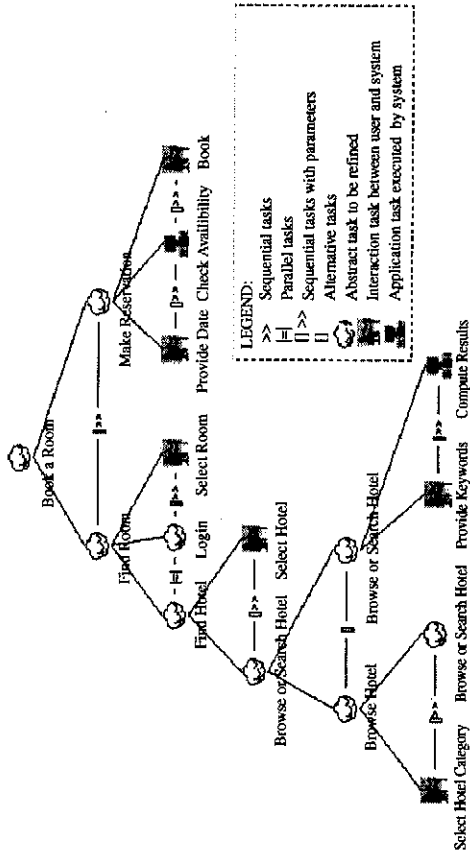


Figure 13.
WSDM: the booking task
CTT

(formally) models the necessary information and functionality needed to complete the task (Figure 14). Moreover, an object chunk can have associated object chunk functions, which allow to model system functionality (e.g. instance creation and selection functions) (De Troyer *et al.*, 2005), (Casteleyn *et al.*, 2006), (Plessers *et al.*, 2005b) (WF). Formerly, object chunks have been based on the Object Role Modeling (ORM) method (www.orm.net) (Halpin, 2001). Today, they are specified using OWL within the WSDM Ontology, while the graphical notation still follows ORM (Figure 14). At the hypertext level, WSDM's navigational model (W.L) consists of navigation tracks, one for each audience class, and models the conceptual structure of the web application in terms of nodes and links (Figure 15). A navigation track can be considered as a "sub-site" dedicated to the information and navigation needs of a particular audience class. Nodes can comprise several object chunks from the content level, which is specified in the WSDM Ontology by the OWL object property *hasChunk*. This relationship is also graphically represented in the navigation track, by connections between object chunks (rounded rectangle) and nodes (rectangles) (Figure 15) (W.L). At presentation level, the site structure model (W.L) maps the concepts modeled at the hypertext level onto pages, i.e. the OWL object property *hasNode* denotes which nodes and links will be grouped onto web pages. Nodes to be presented on one page are surrounded with a rectangle shape with a dog-ear (indicated for the nodes *Select Room*, *Provide Data* and *Book* in Figure 15) (W.L). In addition, WSDM proposes page models defined for each separate page in the site structure model, which allows selection of UI components and positioning of page elements. The mapping of object chunks onto the actual data source is performed in a data source mapping step. Finally, the development process ends with the implementation (generation) of the web application (W.Ph).

3.3.3 Customization modeling. In WSDM's history, two independent proposals for dealing with customization modeling have been published. The first one (Casteleyn *et al.*, 2003), (Casteleyn, 2005), (Casteleyn *et al.*, 2005), focuses on modeling adaptive navigation based on rules specified at design time and triggered due to the browsing behavior of users (C.P). The second one (De Troyer and Casteleyn, 2004), basically

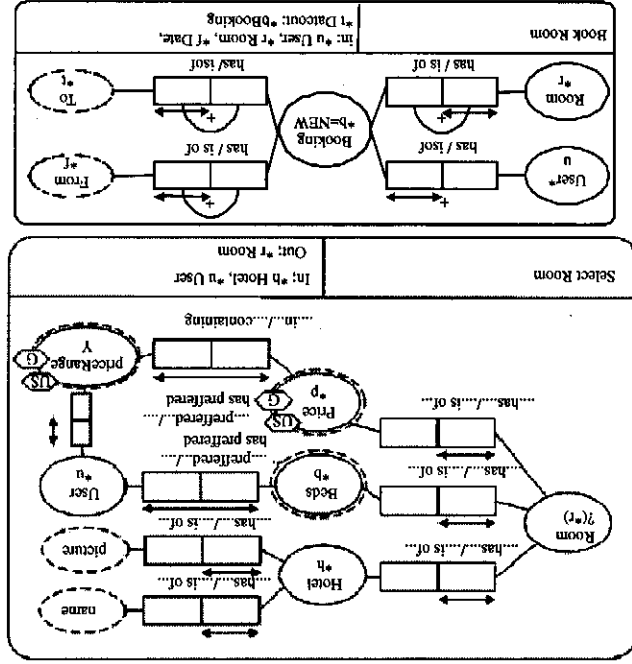
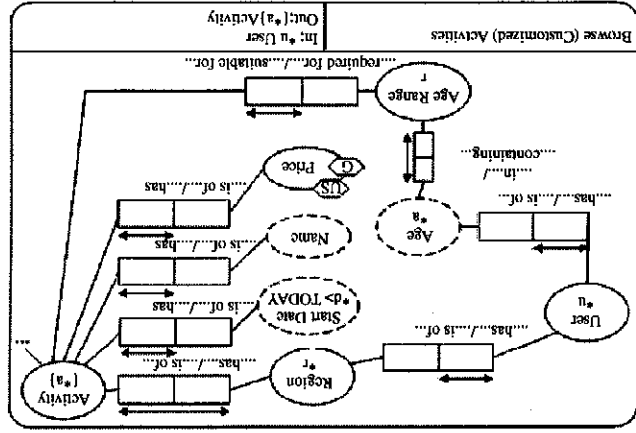
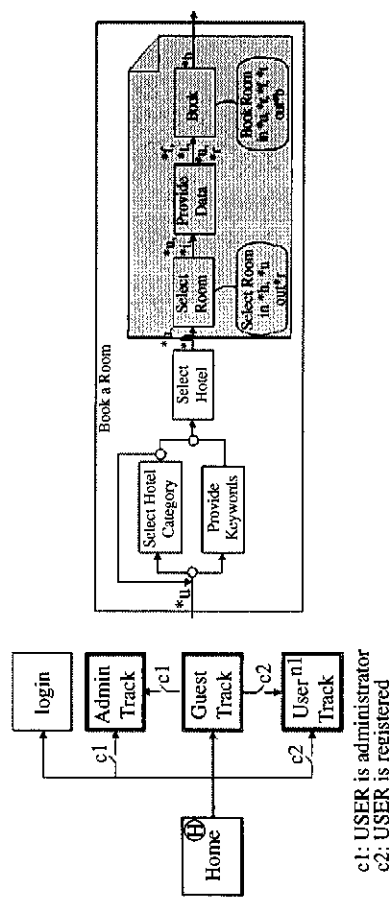


Figure 14.
WSDM: object chunks at
content level

Figure 15.
WSDM: the navigation
model and the user
navigation track



extends each of WSDM's development phases (except implementation) in order to model localization. To do so, the concept of "locality" has been introduced describing a particular place, situation, or location (C.P). In particular, the audience modeling phase has been extended to model localities, i.e. their specifications, characteristics, and their mappings to audience classes. In further design models, localization is considered by annotation-like extensions to the models, which implies no separation of adaptations (C.SA). Context information is actually modeled at the content level but locality labels are used to indicate context-dependent information. Consequently, there is no complete separation of context from the rest of the application (C.SC). Context cannot be extended to support further context properties (C.CE), and neither chronology (C.C) nor complex context can be modeled (C.CC). Casteleyn *et al.* (2003), a first version of the Adaptation Specification Language (ASL) was presented, and an extended, more complete version in Casteleyn (2005), allowing designers (C.CP) to express rules specifying when and how the hypertext level (C.L), needs to be adapted according to monitored user behavior. In this respect, the approach offers an implicit context model, as ASL allows on the basis of events (page load, click, etc.) to specify and track certain user behavior, e.g. *numberOfVisits* of a node and *numberOfTraversals* of a link. It needs to be mentioned that although ASL can be used to specify customization, it was developed for *optimization* (i.e. provide adaptation for *all* users, instead of for the individual user). This way, some form of context chronology can be realized (C.C). The approach offers a set of primitive adaptation operations (C.O), (CAE) on nodes (i.e., *deleteNode* and *addNode*), connections between nodes and object chunks (i.e., *connectChunk* and *disconnectChunk*) (C.I), and links (i.e., *deleteLink* and *addLink*), which can be composed by the designer (or re-used) to form more complex operations. Examples of such more complex adaptations based on the primitive ones are *promoteNode / demoteNode*, which moves a node closer to / further away from the root of a web site, thus making it easier / harder to find (C.CA). Rules in ASL can be applied to single elements but also to groups of elements (C.G) and represent a mechanism to specify adaptations separately from the rest of the web application (C.SA). The approach suggests that these design time rules can be used to generate rules for an adaptation engine within the web application. It does not include support for presentation level adaptation, nor goes into detail on the effect of (navigation) adaptation rules on the presentation level.

3.3.4 *Model-driven engineering*. As already mentioned before, all modeling concepts of WSDM have been defined within the WSDM Ontology. When a designer designs a web application using WSDM models s/he thus populates the WSDM Ontology (models) with concrete instances (M.L). The WSDM approach foresees an implementation phase with a four-step transformation process, respectively, a five-step process when considering semantic annotations (Casteleyn *et al.*, 2006). In this process, the previously defined models are subsequently transformed to the selected output platform, e.g. (X)HTML and WML (PIM2Code). During this process, the different models are integrated into one, thus realizing a PIM2PIM transformation (M.T). The approach, however, does not provide platform description models for the above mentioned platforms (M.P). A proof-of-concept prototype for generating code and particularly semantic annotations from WSDM's design model has recently been described in Casteleyn *et al.* (2006). This prototype currently is able to generate HTML code based on XSLT.

3.3.5 *Modeling example*. On the basis of WSDM's current customization mechanisms, it is possible to model the Customized Activities scenario, the Administrator Links scenario, and a variant of the Special Offers scenario. The customization scenarios Multi-Delivery and Season's Style could possibly be realized by defining different variants of the hypertext and presentation levels for supporting different devices as well as for different seasons. Still, how the different variants are selected in the web application at runtime is assumedly left to implementation. Consequently, these customization scenarios are only partly supported and thus are not considered.

Customization Scenario Customized Activities. In WSDM, activities can be filtered according to the current context at the content level. The content level, i.e. the object chunks, is derived from a CTT, specifying the details of an audience class's task. The task of browsing activities is rather simple. Consequently, for illustration purposes, an example for the more complex booking task of the user audience class is specified by the CTT in Figure 13. To book a room, a room has to be found and after that, a reservation has to be made. To find a room, a hotel must be found first. Either before or after that, the user has to login and can then select a room that s/he likes, etc. Some of the object chunks capturing the information and functional requirements of an elementary task are depicted in Figure 14. The *Select Room* object chunk shows the information requirements for the *Select Room* task of the CTT shown in Figure 13. The input to the object chunk is the current user, represented by the instance "*" of the standard object type User and an instance "*" of the Hotel object type for which name and picture are shown as the only value types. The instance "*" is input from a previous task (e.g. Select Hotel). From the set of rooms of the hotel that match the preferences of the user, and denoted with brackets "{r}", one can be selected by the user as is indicated with "!". The fact that only rooms that match the user's preferences should be offered, is modeled in the object chunk by indicating that the instance "*" of the object type *Beds* of each of the rooms of "{r}" must be preferred by the user, as well as that the price "*" of each of these rooms must be in the price range preferred by the user. The Price and the Price Range value type are tagged with locality labels to indicate information that is dependent on the locality, i.e. the price is different for localities "G" for Germany and "US" for United States. The object chunk *Book Room* creates a new instance (annotated by the keyword "NEW") of booking for a specified

room and date, which are inputs from prior tasks, namely Select Room and Provide Date. The relationships are established with the "+" notation. The object chunk *Browse (Customized) Activities* of Figure 14, shall present customization scenario Customized Activities. In the example, the location of the user is assumed to be resolved to a region and to be stored in the data source. Thus, having the user "*" as input to the object chunk, the region for the user as well as the region's activities can be selected. Furthermore, the activities are selected according to the current date with the keyword "TODAY" and the user's age. The name, the start date as well as the price (localized) of the selected activities will be presented.

Customization Scenario Administrator Links. In Figure 15, the left part depicts the structure of the hypertext level in terms of the specific tracks for the audience classes. The hypertext level of the task *Book a Room* (elaborated in Figure 13) of the *User Track* is presented in detail: nodes are connected with the necessary object chunks and links have attached parameters to be transported. Note that, only the object chunks previously presented in Figure 14 are depicted. With respect to the customization scenario *Administrator Links*, the WSDM notation allows specifying conditional links (link to *Admin Track* and link to *User Track*), which is only available if the associated condition, formulated in a conceptual query language defined on top of ORM, turns out to be true (the bottom-right corner in Figure 15).

Customization Scenario Special Offers. The *Special Offers* scenario can be realized on the basis of ASL. The users shall be provided with a link to the *Region Details* node (n12; not shown in Figure 15) for a particular region, if their browsing behavior indicates a certain interest in that region. The ASL specification of Listing 4 prescribes to add a direct link from the *User track node* (n1; shown in Figure 15) to the *Region Details* for his favorite region node (n12), i.e. the region he visited most (Figure 16).

3.4 The object-oriented hypermedia design model (OOHDM), Rossi et al.

3.4.1 Maturity. The Object-Oriented Hypermedia Design Method (OOHDM) (Rossi and Schwabe, 2006) is amongst the earliest approaches to web application modeling and – dating back to 1994 (Schwabe and Rossi, 1994) – the oldest approach included in this survey (G.T). The approach has been demonstrated by different modeling examples, such as an online catalogue for architecture, an online magazine, a CD store, a university department site, and a conference management system (G.ME). Several Web applications have been developed using OOHDM, for example, the Portinari project (www.portinari.org.br), a web site presenting information about the painter Candido Portinari [1] (Schwabe et al., 1996), and an enterprise knowledge portal in the Brazilian subsidiary of the Xerox Corporation (Schwabe and Salim, 2002) (G.A).

```

let DetailNodes be ALL, n12;
let maxDetails[currentUser] be
  max(DetailNodes)
  [MAP on element: element[currentUser].amountOfAccesses];
forEach node in (DetailNodes):
  if node[currentUser].amountOfAccesses = maxDetails[currentUser]
  then addLink (n1, node);

```

Figure 16.
Listing 4 – WSDM:
special offers scenario

3.4.2 web modeling. The approach specifies a development process comprising five steps (Rossi and Schwabe, 2006), namely requirements gathering on the basis of user interaction diagrams (UID), a refinement of the well known use cases technique (Vilain et al., 2000), conceptual design, navigational design, abstract interface design, and implementation. While OOHDM does not prescribe a particular implementation approach, some supporting tools have been developed, such as the OOHDM-java2 software architecture (Jacynto et al., 2002) (W.Pr), (W.Ph). OOHDM supports all web application levels. The content level, in terms of the conceptual model, is represented as a UML-like class diagram (Figure 17). It has to be noted, however, that due to the use of multi-typed attributes, the OOHDM approach is not fully conform to the UML (Schwabe and Rossi, 1998) (W.L). At hypertext level, the navigational class schema (Figure 19) is derived from the conceptual model and at the same time integrated from the navigational contexts [2]. Navigational contexts represent sets of similar objects which can be organized in such a way that navigation among them is improved (Figure 18). At the end, the navigational class schema represents a view on the content level in the form of a proprietary text-based query language – similar to view-definition approaches in object-oriented databases. It is used to specify the mapping between content level and hypertext level concepts (Schwabe and Rossi, 1998) (W.D). In the navigational class schema presented in Figure 19, the *Hotel* node encompasses the original *Hotel* class as well as the *Feature* class from the conceptual model. In the node *Person*, the attributes of the *Customer* and *Admin* classes have been integrated. An example of the textual specification for the *Hotel* node is given at bottom of Figure 19. The navigational class schema again is represented as a UML class diagram, while the navigational context schema uses a proprietary notation (W.L). For the presentation level, OOHDM originally proposed the concept of Abstract Data Views (ADV) (Cowan and de Lucena, 1995) which represent abstract interfaces for navigational objects such as nodes, links, and access structures (Rossi et al., 1995) (W.L). More specifically, the approach made use of so-called Configuration Diagrams for specifying interface objects, their structural relationships, and the relationships among interface objects (ADV) and navigational objects, e.g. nodes (Rossi et al., 1995) (W.J). Furthermore, the approach made also use of Abstract Data View-Charts (a statechart derivative) for expressing the behavior of an ADV, thus being the only

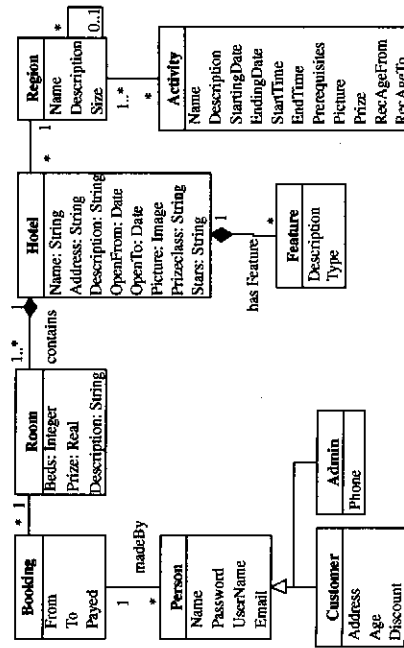


Figure 17.
OOHDM: the conceptual
model

Figure 18.
OOHDM: customized
activities and
administrator links
scenario

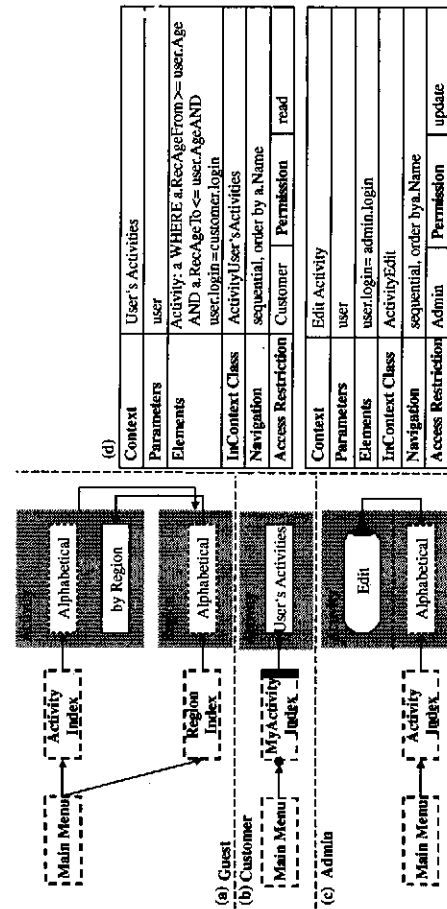
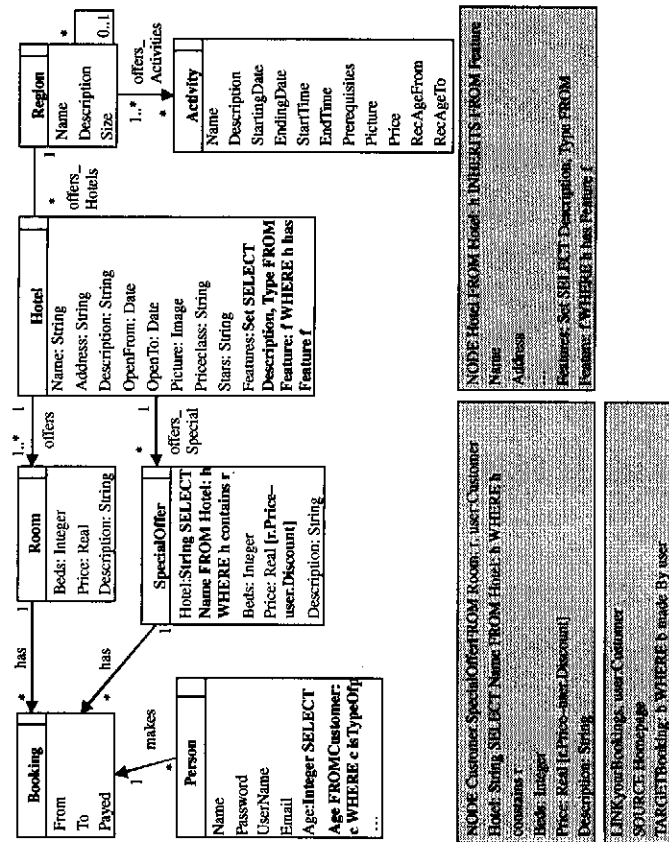


Figure 19.
OOHDM: navigational
class schema



such as mouse clicks, *ElementExhibitors* exhibit a type of content such as text or images, etc. For lack of an explanation, the interface between hypertext and presentation level is assumed to be specified in natural language, only (W.L).

3.4.3 *Customization modeling*. Customization in OOHDM is considered during the design phase, only (Rossi *et al.*, 2001), (Schwabe *et al.*, 2002). OOHDM supposes that different users might have different tasks, access rights, and information needs. Thus, on the one hand, it is suggested to reuse the conceptual model by building different navigational views for different user profiles (C.G). Likewise, it is suggested to build different interfaces for different devices, though this possibility has never been exemplified (C.P). On the other hand, the approach proposes more fine-grained adaptation of content and hypertext levels according to the current user (C.G). In this respect, a "user" variable is introduced to be employed in the text-based query language for defining nodes and links from the navigational class schema, i.e. constraining them, as well as for defining context classes in the navigational context schema. This user variable is then "mapped" to an appropriate class in the conceptual model (C.SC). However, OOHDM neither suggests ways for supporting further, possibly complex context factors, nor context chronology (C.CE), (C.CC), (C.C).

With respect to adaptation, the approach does not define adaptation operations (C.O), (C.AE), (C.CA); as already stated before, adaptations are expressed using OOHDM's query language (C.SA). This language can be used to specify nodes and links (Figure 19) as well as context classes (Figure 18(c)) and thus, supports content adaptation as well as hypertext adaptation (C.L). Since the query language is used for defining the interface between content and hypertext, OOHDM also supports adaptation of the interface (C.I). Though support of presentation level adaptation is claimed (Rossi *et al.*, 2001), (Schwabe *et al.*, 2002), a corresponding demonstration in terms of a modeling example seems to be missing (C.L).

3.4.4 *Model-driven engineering*. OOHDM is described as a set of models, which are built using object-oriented primitives with a syntax close to UML (Rossi *et al.*, 2001). A language specification of OOHDM in terms of a metamodel, a grammar, or a semantic specification, has not been fully developed. Nevertheless, as already explained above, the interface model has recently been replaced by the Abstract Widget Ontology making use of RDF and OWL (M.L). There is no way of specifying model transformations in the OOHDM approach (M.T), neither are platform description models available (M.P). Still, OOHDM-web (Schwabe *et al.*, 1999) is a Web-based environment for OOHDM, which allows the specification of an OOHDM design in XML. Given this specification, and a template page, it generates web pages for the CGI/Lua environment (Hester *et al.*, 1998), thus implementing the specified application. More recently, the HyperDE environment has been developed. HyperDE provides support for rapid prototyping in the context of domain specific languages (Nunes and Schwabe, 2006).

3.4.5 *Modeling example*. Following, the OOHDM solutions to the customization scenarios Customized Activities, Administrator Links, and Special Offers are presented. Although, the approach suggests modeling different hypertext models for enabling device-independence, it is not clear, how the corresponding hypertext can be associated with the current user. Furthermore, the approach offers no mechanism for customizing the presentation level of web applications. Consequently, the customization scenarios Multi-Delivery and Season's Style cannot be realized in OOHDM.

way of behavioral modeling in OOHDM (W.F). Recently, in the context of the SHDM approach for developing Semantic web Applications (Schwabe *et al.*, 2004), a new approach has been developed, namely the Abstract Widget Ontology (AWO) (Rossi and Schwabe, 2006) (W.L). The entire interface is specified by several ontologies using RDF and OWL as a formalism. The AWO proposes a set of concepts whose instances will comprise a given interface: *SimpleActivators* react to external events

The conceptual model of OOHDM is described with a UML class diagram (Figure 17). Although possible in OOHDM, in this example we refrain from using multi-typed attributes.

Customization Scenario Customized Activities. Figure 18 presents some of the navigation context schemata for the different tasks of different user types. For example, a guest user will be able to navigate from a *Main Menu* to both an *Activity Index* and a *Region Index* (Figure 18(a)). From these indices, s/he will be able to access the *Activity* and *Region* node (depicted as grey rectangles) in different contexts, e.g. alphabetically or sorted according to some criteria. Navigating from the *Region Index*, all regions will be displayed in alphabetical order. Having selected a region, the user can navigate to the *by Region* context of the *Activity* node and visit the activities of the region. With respect to the customization scenario Customized Activities, customers of the web application have an additional index, which requires the user to login (see the black bullet at the incoming arrow of *MyActivityIndex* in Figure 18(b)). The *Activity* node then can be visited in the *User's Activities* context, which will display only activities that are appropriate for the current user's age. This filter condition as well as the access restriction is defined within the Elements section of the context specification card *User's Activities* in Figure 18(c), where it is also tested if the user is of type *Customer*. Unfortunately, an explanation of OOHDM's user variable and a possible user model is missing. This example thus has been directly derived from examples given in Rossi *et al.* (2001), Schwabe *et al.* (2002).

Customization Scenario Administrator Links. OOHDM's solution to the Administrator Links scenario is shown in Figure 19(a). The administrator will be able to navigate from the context showing the activities in an alphabetic order to the context for editing. Obviously this time, similar access restrictions to this context are needed, which are specified in the context specification card *Edit Activity* in Figure 19(b). How a user is identified as administrator, however, cannot be expressed in OOHDM.

Customization Scenario Special Offers. For realizing the Special Offers scenario, the node *SpecialOffer* is introduced to the navigational class schema in Figure 19 and realizes a content adaptation, since the price of a room is adapted according to the user's discount. Nevertheless, it is not possible to model that this discount shall be given according to the user's booking history and thus, it is not possible to fully model the customization scenario Special Offers. On the bottom of Figure 19, the full specification of the *SpecialOffer* node is presented in terms of OOHDM's textual query language. This mechanism can also be used to define several navigational class schemata for different types of users with different access rights. For example, for guest users there will be no need for information on booking or special offers. Furthermore, the query language allows for link personalization. For example, on the bottom of Figure 19, the specification of the link *yourBookings* can be found. This link will be displayed on the homepage and will include the bookings of the current user, only. To do so, the user variable is used, which is mapped to the *Customer* class.

3.5 *The UML-based web engineering approach (UWE), Koch et al.*

3.5.1 *Maturity.* The UML-based web Engineering (www.pst.fh.lmu.de/projekte/uwe/) (UWE) approach (Koch, 2001), (Koch and Kraus, 2002) has been continuously developed since 1998. Recent work heads towards model-driven development of web

applications (Koch *et al.*, 2006), (Koch, 2007) based on model transformation techniques and MDA standards (OMG, 2003) (G.T). In addition, aspect-orientation was recently introduced into UWE to address customization (Baumeister *et al.*, 2005), (Zhang, 2005). The approach has been illustrated using several modeling examples, amongst them a music portal, an online library, a conference management system, an online movie database, an adaptive e-learning system and an e-store (G.ME). UWE is currently being used for modeling the web application of a medical engineering company, although its application in real-world projects has not yet been published (G.A).

3.5.2 *web modeling.* The UWE approach supports UML 2.0 models for requirements (use case model), content (content model), hypertext (navigation model), and presentation levels (presentation model) (W.L). These models are supplemented, when necessary with a user model, a context model and a process model. Structural modeling is based on class diagrams at all levels, i.e. special stereotypes for the specific web concepts are introduced, such as «navigation class», and «navigation link». While at hypertext level, state chart diagrams can additionally be used to model navigation scenarios, at presentation level, sequence diagrams can be employed to depict presentation flows (W.F). The interface between content and hypertext levels is described in terms of OCL. This means each attribute of a «navigation class», i.e. a hypertext node, in the navigation model is derived from information stored in the content model (W.I). The navigation model is extended to support navigation by accessingslements such as menus, indexes, queries, guided tour, etc. For each navigational class of the navigation model, at the presentation level a separate diagram is created, describing where and how the navigation primitives will be presented to the user. The UML composition notation for classes is used together with a set of stereotypes for the nested elements (Koch and Kraus, 2002), e.g. «text», «form», «button», and «image» (Baumeister *et al.*, 1999). Furthermore, UWE provides natural language guidelines to infer the navigation model from the content model and to infer the presentation model from the navigation model. These guidelines form the basis for the implementation of the model transformations within the tool ArgoUWE, and the transformations defined in ATL (Kraus *et al.*, 2007). UWE has a well-defined development process based on RUP (Kruchten, 2000) (W.Pr) and provides explicit support for RUP's five phases including for each phase the workflow requirements engineering (use-case diagram), analysis (content model, navigation model), and design (presentation model and refinement of all models) (W.Ph) (Koch, 2001).

3.5.3 *Customization modeling.* Customization modeling support within UWE is mainly considered during analysis and design phases (C.CP) and has its origins in the Munich Reference Model (Koch and Wirsing, 2002). At content level, Koch *et al.*, distinguish between a domain model and a user model capturing contextual information about the user. Relationships between classes from both models allow for partial separation of context only (C.SC).

Besides the user model, an adaptation model is suggested, which is realized by means of a UML communication diagram and describes a set of condition-action rules (i.e., construction rules, adaptation rules, and acquisition rules), the rules' temporal relationships as well as when they are triggered. Still, the rules themselves are described in natural language only. The UWE approach also suggests concepts for modeling context, which are separated into a *User* package, and an *Environment* package in the UWE metamodel presented in Koch and Kraus (2003) (C.P).

UWE proposes in Baumeister *et al.* (2005) the use of model aspects to introduce model elements needed for modeling adaptivity to the navigation model, and the use of link traversal aspects to model the execution of additional actions (e.g. updating the user model) when a link is traversed. Possible reactions to link traversal aspects include link annotation aspects for modeling link annotation and link traversal aspects for adding and deleting links from the navigation model.

Syntactically, these aspects may be seen as special cases of Aspect-Oriented Class Diagrams (AOCs, (Zhang, 2005)): each aspect is represented as a UML package and contains a pointcut compartment and an advice compartment. The pointcut is a pattern, optionally containing a special element, the formal parameter, which is annotated by a question mark. It selects all model elements in the base model that match the pattern and thus will be subject to weaving, thereby instantiating the formal parameter. The process of changing the elements is called weaving in aspect-orientation. The advice defines changes to the selected model elements. This general notation is, as UWE, defined as a UML extension. For more details of AOCs, the interested reader is referred to (Zhang, 2005).

Since both UWE and AOC are UML extensions, the adaptivity modeling of UWE benefits from the whole expressive power of the UML. In particular, arbitrary context (C.CE) and adaptation (C.AE) may be defined just as in the UML. Complex content (C.C) and complex adaptation (C.CA) may be defined by combining aspects. This approach may be used to model content, navigation and presentation adaptation (C.L), (C.I). The adaptations can be separated with aspects from the rest of the application (C.SA). The granularity of adaptations – depending on the aspects' pointcuts – ranges from micro to macro (C.G). However, no example of modeling context chronology (C.CC) has been reported yet.

3.5.4 Model-driven engineering. The UWE language is defined as an extension of the UML metamodel (Koch and Kraus, 2003) and additionally provides a UML profile for interoperability purposes (M.L). The guidelines for generating preliminary models, e.g. the navigation model, from models created during earlier development phases have been automated within the tool support. Furthermore, transformation of requirements specified in terms of use case diagrams and activity diagrams into UWE models at content and hypertext level based on MOF (OMG, 2002) and QVT (OMG, 2005) has been recently proposed in Koch *et al.* (2006). This approach, however, has not yet been automated (M.T). In addition to model-to-model transformations, UWE supports some model-to-code transformations. UWE uses model transformation rules written in the transformation language ATL to translate UWE content and presentation models into Java Beans and JSPs, and UWE supports the execution of the business processes integrated in the navigation model with a virtual machine built on top of the controller of the Spring framework (Kraus *et al.*, 2007).

3.5.5 Tool support. ArgoUWE is the free tool support for UWE (T.C). It is a proprietary extension of the UML modeling tool ArgoUML (T.B), (T.O). ArgoUWE 0.16 provides general modeling support (T.V) but customization modeling has not yet been realized within the tool (T.M). During modeling, the tool checks the artifacts for errors as well as offers help to resolve the identified problems with a wizard (T.CC). Moreover, the tool carries out semi-automatic transformations to generate the navigation model from the content model, as well as the presentation model from the navigation model (T.T). With respect to the presentation model, ArgoUWE does not yet support the

composition notation of classes but uses composition associations to connect the attributes of the navigation class to the owning presentation class. UWE's development process, however, is not realized within the tool, since typical iterations within the development of a web application are not supported without losing information, e.g. changing the content model without losing information from the hypertext level is not possible (T.P). Code generation (T.CG) is realized as an Eclipse-plugin based on ATL transformations (Kraus *et al.*, 2007), but not yet integrated in ArgoUWE. Versioning support allowing for shared editing is not supported yet (T.Co).

3.5.6 Modeling example. Requirements modeling is supported by UWE mainly based on the features provided by the UML: use cases and activity diagrams. Two stereotypes «navigation» and «personalized» are defined as extension of the UML metaclass *UseCase*. The use of these stereotypes enriches the requirements model with additional information about use cases supporting business processes and customization. In the same way a set of stereotypes are defined to increment the expressiveness of the workflows modeled by activity diagrams (see (Koch *et al.*, 2006) for further details).

UWE uses a UML class diagram to model the content model (see Figure 20). UWE suggests to use as far as possible all the variety of the UML model elements, such as associations, aggregation, composition, multiplicities, association ends (role names), etc. in order to augment the expressive power of the models. The backbone of the content model includes a class *Hotel*, a class *Region* and a class *Activity*. A hotel has a set of *Rooms* that can be booked and may have some special *Features*, such as swimming pool, garage, etc. A *Region* is characterized by a name and a pair of coordinates among other attributes and may have associated subregions and activities. Further details are illustrated in the diagram shown in Figure 20.

Following the principle of separation of concerns, UWE proposes to model context information, in particular user information in a separate model (Figure 21 for user and context model). This model is also represented as a UML class diagram. It always contains a class *Session*. User modeling is performed through a class *User*. Similarly, context modeling requires a class *Context*, which has different types of contexts. It is assumed that the application will be able to detect the user's location, which will then be resolved to a certain region, providing the necessary data for customization. Furthermore, the user model captures the necessary data for browsing that enables adaptation at presentation level making browsing faster by elimination of images.

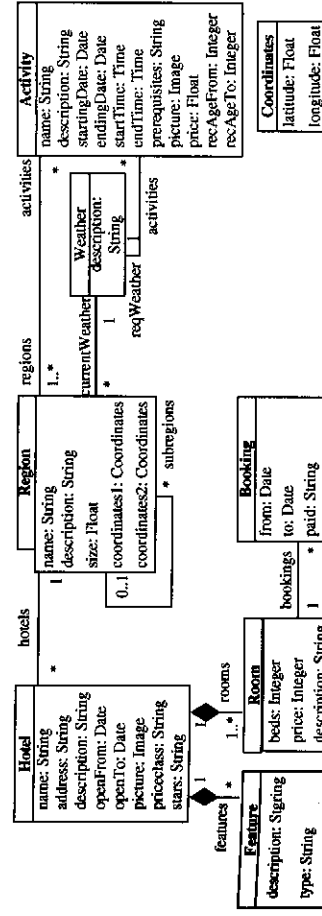


Figure 20.
UWE: content model

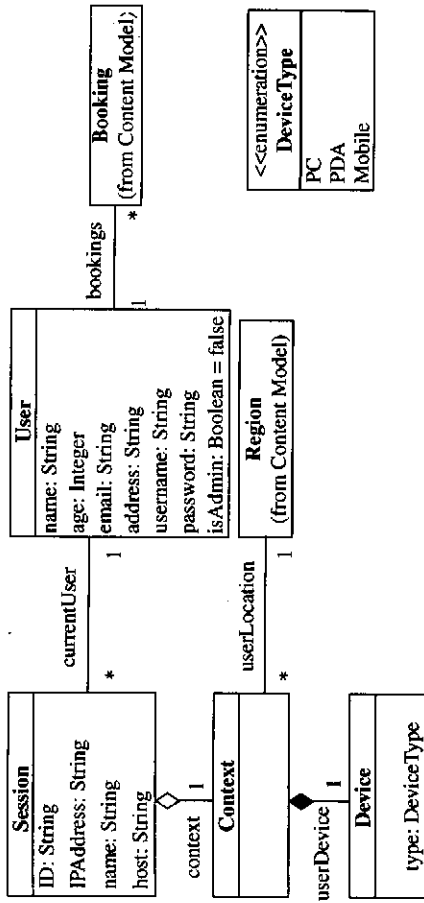


Figure 21.
UWE: user and context model

As shown in Figure 21, the user and context model is connected to the content model, i.e. an association connects *User* to *Booking* and an association connects *Context* to *Region*.

Part of the navigation diagram is depicted in Figure 22, which shows that indexes of hotels, regions, and activities can be reached from the main menu of the TIWA. From the indexes, the hotels, regions and activities can be viewed in detail. In particular, the *Region* navigation class offers to browse the region's hotels and activities via two further dedicated indexes *HotelsByRegion* and *ActivitiesByRegion*, respectively. UWE proposes the use of the stereotypes «navigation class» (visualized as), «menu» 0, «index» 0, «query» 0, «guided tour» 0, «process class» 0 for the representation of the hypertext node concepts and «navigation link» and «process link» for the hypertext link concepts. These stereotypes are defined as extension of the UML metaclass *Class* and *Association*, respectively. Examples of these stereotypes are shown in Figure 22. To avoid overloading the stereotype «navigation link» is omitted in this navigation diagram. In addition, this diagram depicts the tagged values {isHome} and {isLandmark} which are reachable from every other node. Details of the *BookRoom* business process are shown in Figure 23. Login is a further activity, which also can be detailed with another UML activity diagram.

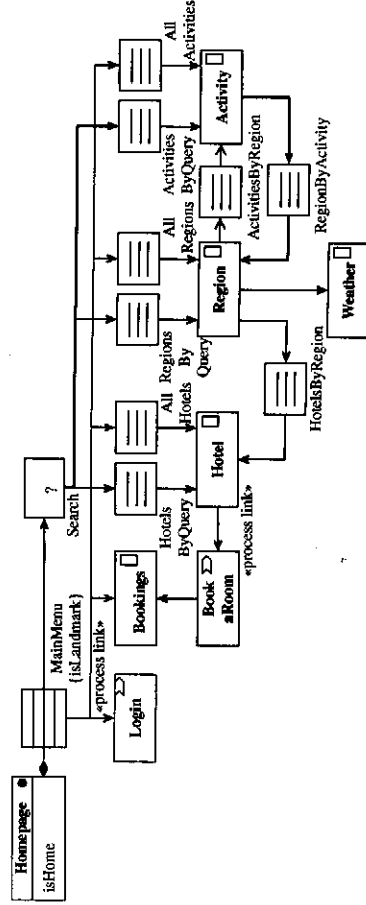


Figure 22.
UWE: navigation model

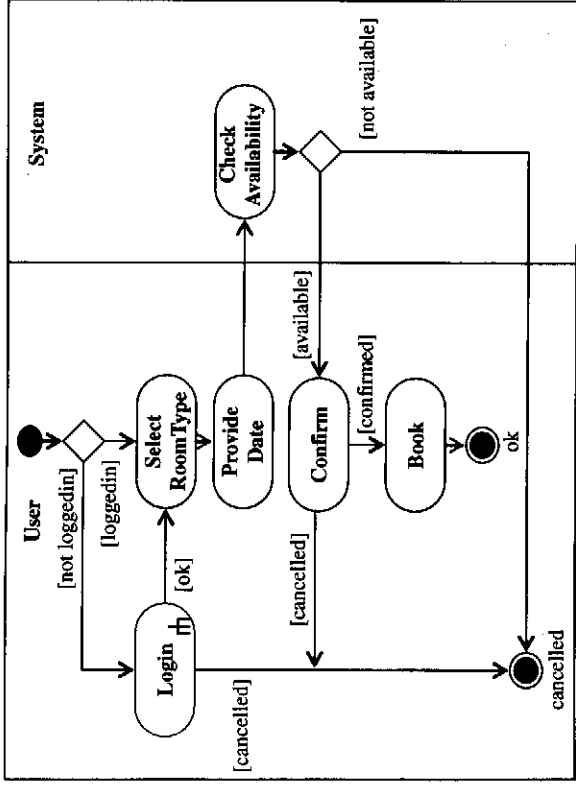


Figure 23.
UWE: simplified business process of room booking

In the following, UWE's customization approach is applied to the scenarios of the TIWA example as described in Baumeister *et al.* (2005), Zhang (2005) and Koch *et al.* (2006). Currently, the approach covers content and hypertext level adaptations distinguishing between model and runtime level. A «model aspect» is woven statically at model level. For the hypertext level in particular, UWE distinguishes different types of dynamically woven aspects («runtime aspects»): «link transformation aspect», «link annotation aspect» and «link traversal aspect». For the presentation level, UWE focus on the abstract presentation modelling. Thus, look & feel modelling and adaptation as required in the current season's style scenario has not been addressed by UWE yet.

Customization Scenario Customized Activities. The TIWA web application can be made context-aware offering an activities list, which presents, e.g. activities filtered by the user's age and the current date. The aspect-oriented approach of UWE uses an aspect as shown in Figure 24. Its pointcut includes the *MainMenu* and the navigation class *Activity*. The advice of this «model aspect» defines the result of the

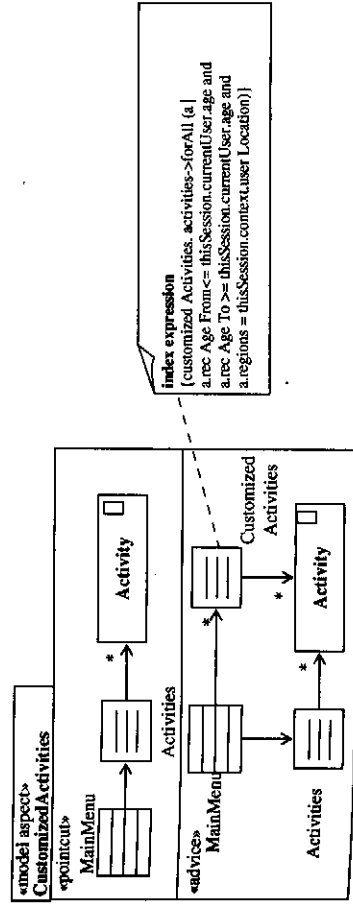


Figure 24.
UWE: customized activities scenario

weaving: an index *CustomizedActivities* and two navigation links enabling navigation from the main menu to the customized activities index and from the index to an activity instance which are added to the menu. Activities are selected according to the OCL expression given in the comment, as shown in Figure 24.

Customization Scenario Special Offers. For modelling the scenario special offers, UWE also uses a graphical notation based on Aspect-Oriented Class Diagrams (Zhang, 2005). For this scenario, two aspects are defined (see Figure 25), which will be weaved into the TIWA application. First, the «model aspect» *Special Offers* selects in its pointcut the class with the name *User* (there may be at most one such class in a consistent model) and enriches it (question mark) by an attribute *discount*. Second, the «link traversal aspect» *Discount* defines a post condition of traversing the link from any arbitrary navigation class (wildcard *) that the attribute *discount* is set to *true* for exactly every fourth booking of the current user.

Customization Scenario Administrator Links. In the user model, context information about the user is captured. In particular the *isAdmin* attribute of the class *User* is set to *true* after each successful login of an administrator. Customization functionality for the administrator links can then be introduced to the hypertext level also using aspects (see Figure 25). In the runtime aspect *AdministratorCRUD* the pointcut selects all links from an index to a navigation class (applied to our navigation model), selected are all links leading to the classes *Hotel*, *Region*, and *Activity*, and adds in the advice two process classes *Edit* and *Delete*, reachable from the (selected) navigation class, and a process class *Create*, reachable from the (selected) index. In the runtime aspect *AdministratorBrowsing*, navigation paths that are only visible to an administrator are defined in the advice (Figure 26).

Customization Scenario Multi-Delivery. The customization of the application eliminating images from the pages if a small screen device is used is quite simple to model. It implies adaptation at presentation level. To do this, UWE applies AOOD to the presentation model, as shown in Figure 27. The pointcut of the *SmallScreen* identifies all the classes with the stereotype «image» of the application; the advice eliminates such elements from the model at runtime. On the left hand of Figure 27 the presentation classes of *Hotel* and *Activity* are shown.

3.6 The object-oriented hypermedia method (OO-H), Gomez et al.
3.6.1 Maturity. The Object Oriented Hypermedia (OO-H) Method (Gomez et al., 2001), (Gomez et al., 2000) first emerged in 2000, formerly as an extension of the OO-Method

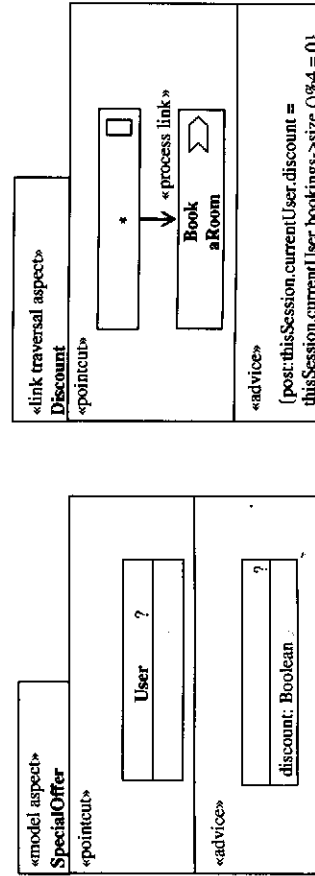


Figure 25.
UWE: special offers scenario

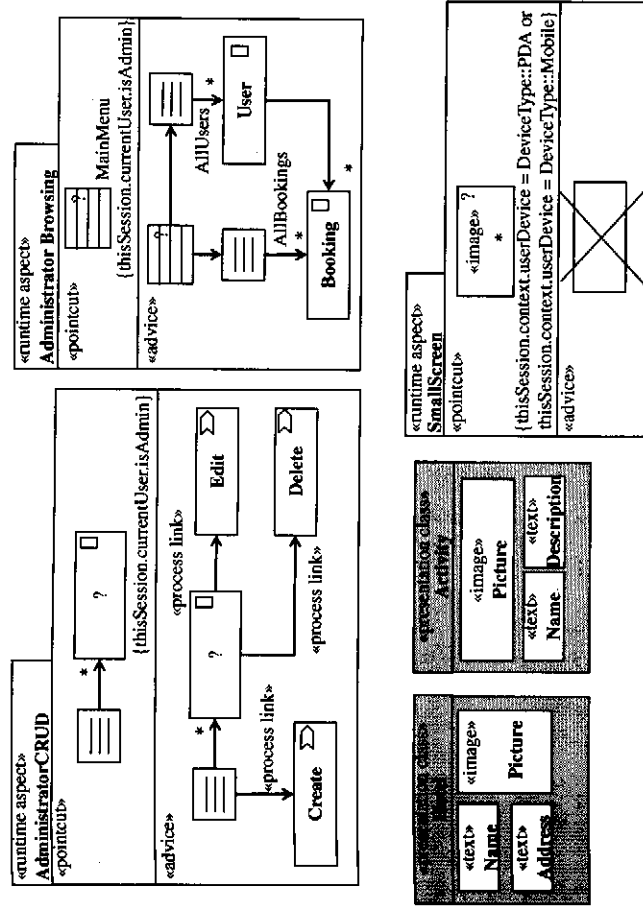


Figure 26.
UWE: administrator links scenario

Figure 27.
UWE: multi-delivery scenario

(Pastor et al., 1997), a design method for object-oriented systems. The most recent publications deal with personalization of web sites (Garrigos et al., 2005a), (Garrigos et al., 2007) (G.T). The approach is demonstrated using multiple examples including amongst others a conference management system, a library system, a forum application, an e-store application, a hotel reservation system, and an e-mail system. Moreover, the OO-H method and its tool support in terms of the VisualWade CASE tool (www.visualwade.com) have been applied in the development of several real-world applications as is reported in Gomez et al. (2005) (G.A).

3.6.2 web modeling. The OO-H method is based on UML. It defines its own process for the design phase, while other phases in the development lifecycle are described with guidelines, only (W.Pr) (Gomez et al., 2001). The approach supports all phases from requirements engineering with use-case diagrams to implementation with the methods tool support (Cachero et al., 2001) (W.Ph). The approach comprises different models for the content, hypertext and presentation level (W.L). Standard UML class diagrams are used for content modeling. With the introduction of UML activity diagrams for modeling processes (Koch et al., 2004), the OO-H method also supports behavioral modeling at the content level (W.F). At hypertext level, so-called navigation access diagrams (NAD) are associated with each user type, capturing the navigation paths and the services (from the content level) the users can activate (Figure 29). They consist of navigation classes which represent views on the conceptual classes from the content level. The interface is specified graphically, i.e. the "label" of a navigation class consists of its name separated with a colon from the name of the underlying conceptual class (W.I). Navigation classes are associated with each other through different types of navigation links, which may have both, a navigation pattern from the OO-H pattern

catalogue (Gomez et al., 2001) (e.g. *Index*, *GuidedTour*) and a set of OCL-like navigation filters, associated. Different types of collections represent (possibly hierarchical) access structures defined on navigation classes or navigation targets and may also have both, a navigation pattern and a set of navigation filters associated. Additionally, these concepts can be grouped within navigation targets in order to cover a certain user navigation requirement. Mapping rules allow inferring default navigation access diagrams from activity diagrams modeling processes (Koch et al., 2004). At presentation level, starting from a NAD, a default abstract presentation diagram (APD) (Cachero et al., 2000) can be generated, by using so-called NAD2APD mapping rules (Gomez et al., 2001) (W.J). An APD can be interpreted as the sitemap of the web application consisting of a set of "abstract pages" associated with links and its modeling concepts are defined in several DTDs. The default APD can be refined through patterns, defined as transformation rules (Gomez et al., 2001), which are implemented in Python. Furthermore, the OO-H CASE tool also includes the composite layout diagram (CDL), which allows further refinement of the user interface (i.e., the XML specification of the APD) in the manner of a WYSIWYG editor (Cachero et al., 2001b).

3.6.3 Customization modeling. The OO-H method suggests a personalization framework in the form of a UML class diagram (Garrigos et al., 2003b), (Garrigos et al., 2005a) comprising a user model and a personalization model. The user model actually extends the content level (C.SC) and allows capturing context information with respect to user, location, device, time, and network (C.P). Further context properties can be introduced through inheritance from a generic *Context* class (Garrigos et al., 2005a) (C.CE). The framework currently does not foresee complex contexts or a concrete mechanism for modeling a change in context over time (C.CC), (C.C). The personalization model consists of a set of ECA rules, which realize different personalization strategies. Firstly, acquisition rules define how context information is acquired. Secondly, personalization rules, which belong to a certain profile or to all users, define the adaptation that has to be made. A profile encompasses a set of personalization rules supporting a group of users with similar needs, e.g. users accessing the web application through small screen devices. And third, profile rules associate a user to a profile. Personalization rules are further distinguished into rules manipulating the content, the hypertext, and the presentation level (C.L), (C.I). The existing set of concrete rules (C.O) includes fine-grained adaptations such as *SelectInstance*, *SelectAttribute*, *HideLink* (Garrigos et al., 2007) (used to show attributes in the Customized Activities scenario in Listing 6 and to show a link in the Special Offers scenario in Listing 7) but also coarse-grained adaptations (*SetCSSTemplate* (Garrigos et al., 2003b) used in the Season's Style scenario in Listing 13) (C.G). The set of rules can be extended again through inheritance from the generic rule classes (i.e., *ContentRule*, *NavigationRule*, and *PresentationRule*) (Garrigos et al., 2003b) (C.AE). The combination of adaptations to form complex adaptations is considered (C.CA) in the latest proposal. The ECA rules are specified separately from the other models (C.SA) following the syntax of the EBNF-based Personalization Rule Modeling Language (PRML), which can be interpreted by a rule engine at runtime (Garrigos et al., 2003a). The use of this rule language, however, means that any extension of existing adaptation operations has to be done to this language (C.AE). Customization is considered in the design and the implementation phase of the development process (C.CP).

3.6.4 Model-driven engineering. The OO-H language definition for the content and hypertext level is realized as an extension of UML and OCL (Cachero et al., 2001b). Still, the concepts of the presentation level, i.e. the APD, are defined as a set of DTDs and a separate language for the specification of customization is provided with the EBNF-based PRML (M.L). The OO-H approach provides mapping rules from activity diagrams to NADs and from NADs to APDs. As mentioned before, there is tool support for the OO-H method, which implements the NAD to APD transformations (PIM2PIM) as well as modeling PIM2Code transformations (M.T). The approach, however, does not describe platform models (M.P).

3.6.5 Tool support. The current version of the VisualWade tool is 1.2 (T.V). VisualWade is a commercial tool, but offers a trial version (T.B), (T.O), (T.C). It allows modeling in general but does not support customization modeling. Apart from VisualWade, the generation of UWAs on the basis of the PRML language is recently provided by a prototype described in Garrigos et al. (2007). In this work, the OO-H approach has evolved to the distinct Adaptive OO-H approach, (T.M). The OO-H method's process is supported only partly within VisualWade, e.g. no use-cases for the requirements definition phase are available. Apart from that, the conceptual model, the NAD as well as the CLD are supported, as mentioned before. The APD actually is not used explicitly but indirectly via the CDL. The user will start from a conceptual model, via the NAD to the CLD, but can go back to previous models for changes. Changing the NAD, however, means losing the presentation model which has to be generated anew (T.P). VisualWade supports PIM2PIM and PIM2Code transformations (T.MG) as well as provides an OCL-based action language for transformations and code generation capabilities (Gomez et al., 2005). With respect to code generation, VisualWade currently supports one model compiler that generates code for the PHP target platform. The application can be generated in several independent steps such that the application logic and the database, for example, can be created in random order (T.CG). The modeled artifacts can be statically checked on demand and the user is provided with hints indicating how to solve the detected problems (T.CC). With respect to collaborative development of web applications, VisualWade currently provides no specific support (T.Co).

3.6.6 Modeling example. OO-H's PRML allows for modeling all customization scenarios. Before presenting the individual customization scenarios in terms of PRML rules, however, the content and hypertext level as well as the context model of the web application are presented. The content level of the TIWA is extended with the context model (Figure 28). More specifically, the given context framework has been extended with a *WeatherContext* class. Furthermore, the conceptual classes of the content level have been extended with operations for creating, updating, and deleting objects of the specific types. Figure 29 presents part of the hypertext level. In this case, one NAD has been designed for guests and registered users of the TIWA. From the homepage, users will have to login to the TIWA and will be redirected to the *Homepage* menu if they belong to the user group "user" (*To Homepage* link). The links supporting the login process have attached OCL-like preconditions and navigation filters, which will be ignored for the remaining links for readability purposes. A separate view for administrators is indicated with the navigational target *AdminView* but not further detailed. If the user belongs to the usergroup "admin", s/he will be redirected to the *AdminView* (*To AdminView* link). In case the input retrieves no dataset, i.e. no user

account can be found, the user is redirected to an error page (cf. *Error* link). From the *Homepage* menu the user can navigate, for example, to the *Hotels* navigation class which will present an index of hotels, as is indicated by the second part of the navigation class label which denotes the corresponding conceptual class from the content level. Likewise, the user can browse regions and activities and visit the navigational classes presenting an object of interest, e.g. the *Hotel* navigation class provides detailed information on one hotel. For those links that are not accessible for guests, e.g. *Create Booking*, *Your Bookings*, and *Edit Activity*, special rules need to be designed in PRML such as has been done for the *Edit Activity* link in the customization scenario Administrator Links. The *Edit Activity* link and the *Create Booking* link are service links that call the *Edit()* and *New()* operations of the *Activity* and *Booking* conceptual classes when the user books a room or the administrator edits an activity, respectively.

As already stated above, all customization scenarios can be realized with PRML. Figure 30 outlines the realization of the customization scenarios on basis of ten PRML rules: one acquisition rule, three profile rules, and six personalization rules. All of them need to be initialized (init section) meaning that all rules will be associated to a profile group and thus, to the users of that group. The default group "OOH:all" applies to all users. In the acquisition section of Figure 30, when a new session starts, the acquire-Context rule will collect the necessary context information to be available in other rules, i.e. the device used to access the TIWA as well as the current date (Figure 31).

Also at session start, the user will be attached to one or more profile groups (cf. profile section). In the personalization section, the actual rules for adapting content, hypertext, and presentation of the TIWA are specified. Following, the profile rules as well as the personalization rules will be explained for the individual customization scenarios.

Customization Scenario Customized Activities. The Rule *customizedActivities* to the (Figure 32 Listing 6) is applied if the user navigates from the *Homepage* to the

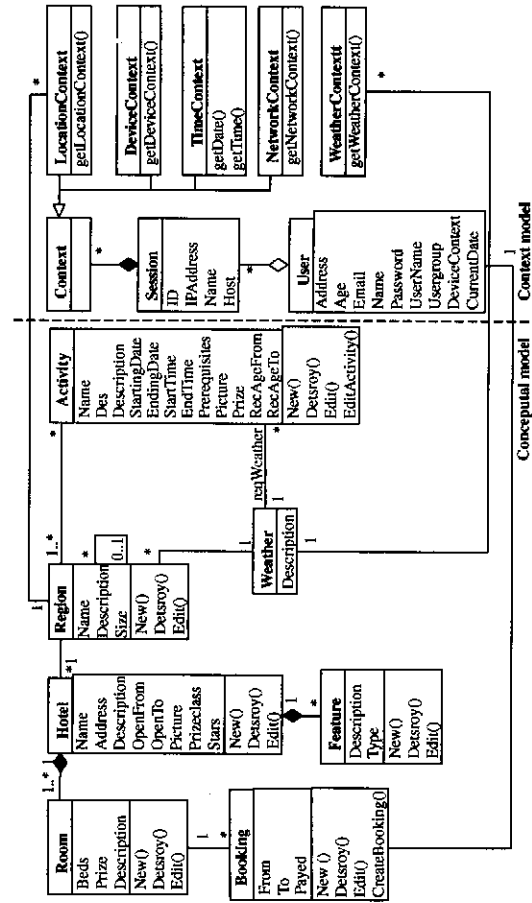


Figure 28.
OO-H: conceptual model
and context model

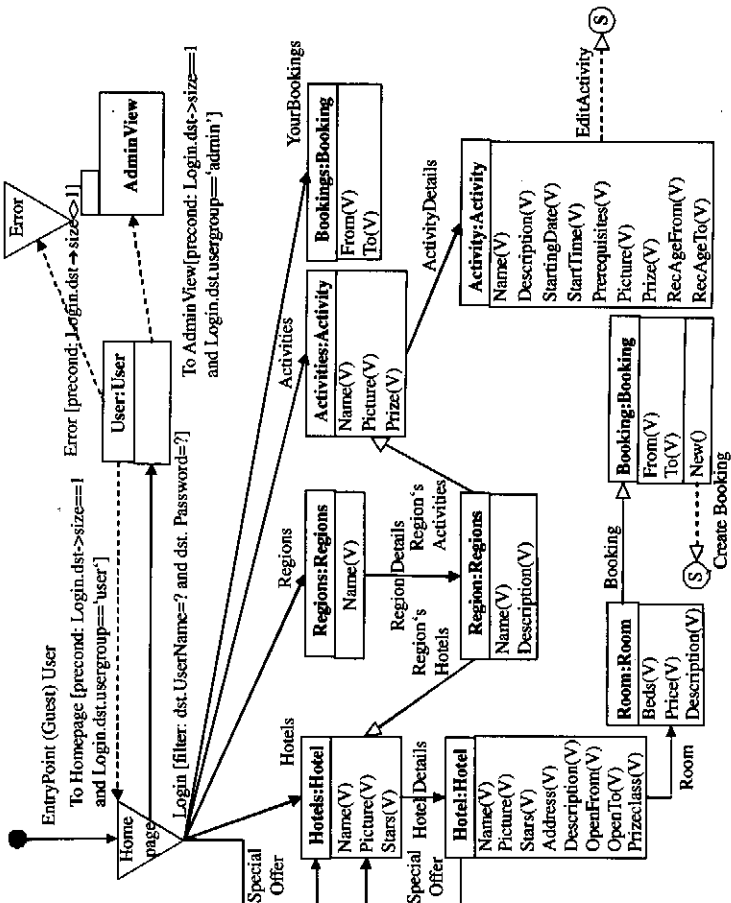


Figure 29.
OO-H: navigation access
diagram

```

#INIT SECTION
When init do
  AttachRuleToGroup("acquireContext", "OOH:all")
  AttachRuleToGroup("defSmallScreen", "OOH:all")
  AttachRuleToGroup("defSeason", "OOH:all")
  AttachRuleToGroup("defUserGroup", "OOH:all")
  AttachRuleToGroup("specialOffers", "user")
  AttachRuleToGroup("editLink", "admin")
  AttachRuleToGroup("ignoreImages", "smallScreen")
  AttachRuleToGroup("ignoreStyle", "summer")
  AttachRuleToGroup("winterStyle", "winter")
endWhen

#ACQUISITION SECTION
#RULE: "acquireContext" priority="high"

#PROFILE SECTION
#RULE: "defSmallScreen" priority="medium" //CustomizationScenario(4)
#RULE: "defSeason" priority="medium" //CustomizationScenario(5)
#RULE: "defUserGroup" priority="medium" //CustomizationScenario(3)

#PERSONALIZATION SECTION
#RULE: "customizedActivities" priority="medium" //CustomizationScenario(1)
#RULE: "specialOffers" priority="medium" //CustomizationScenario(2)
#RULE: "editLink" priority="medium" //CustomizationScenario(3)
#RULE: "ignoreImages" priority="medium" //CustomizationScenario(4)
#RULE: "summerStyle" priority="medium" //CustomizationScenario(5)
#RULE: "winterStyle" priority="medium" //CustomizationScenario(5)

```

Figure 30.
OO-H: personalization
rules

```

#RULE: "acquireContext" priority="high"
When SessionStart do
  UM: User.DeviceContext=UM.DeviceContext.getDeviceContext()
  UM: User.CurrentDate=UM.TimeContext.getDate()
endWhen

```

Figure 31.
Listing 5 - OO-H: context
acquisition rule

Figure 32.
Listing 6 - OO-H:
customized activities
scenario

```
#RULE: "customizedActivities" priority="medium"
When Navigation.Activities (NM.Activity* acts)do
  Foreach a in (acts) do
    If (UM.User.Age >= a.RecAgeFrom and
        UM.User.Age <= a.RecAgeTo and
        UM.LocationContext.getLocationContext = a.region and
        UM.WeatherContext.getWeatherContext = a.reqWeather and
        Date < a.StartingDate and
        UM.TimeContext.getTime() < a.StartTime)
      then
        acts.SelectInstance(a)
      endif
    endForeach
  endWhen
```

Activities node. Depending on the user's age, location, the current date and time, as well as the current weather, the user will be presented a customized selection of activities. In this respect, the adaptation functionality is encapsulated in the action *SelectInstance* which selects the proper instances of the class activity to be shown.

Customization Scenario Special Offers. If the user views three hotels of the same region for at least 60 second (*LoadSet* event (Garrigos and Gomez, 2006)), the user will be presented the link *Special Offers* to the *Hotels* node. In this node, special offers for the specific region will be presented (Figure 33 Listing 7). For this purpose the *hideLink* action is used. In PRML all the actions have their opposite action defined and it is specified by negating them with a not operator. This is specified with the rule *specialOffers*:

Customization Scenario Administrator Links. The profile rule *defUser-Group* (Figure 34 Listing 8) identifies the user type of the current user, i.e. if the user has logged in/she will be associated to either the user or the admin profile group defined in the init section in Figure 30 with *AttachUserToPGroup* (Garrigos et al., 2005a).

Figure 33.
Listing 7 - OO-H: special
offers scenario

```
#RULE: "specialOffers" priority= medium"
When LoadSet [NM.Hotel (NM. Hotel h1, 1, 60) | NM.Hotel (NM. Hotel h2, 1,
60) | NM.Hotel (NM. Hotel h3, 1, 60) ] do
  If (h1.htoRegion.name=h2.htoRegion.name and
      h2.htoRegion.name=h3.htoRegion.name) then
    not(hideLink(NM.SpecialOffer))
  endif
endWhen
```

Figure 34.
Listing 8 - OO-H:
administrator links
scenario (1)

```
#RULE: "defUserGroup" priority="medium"
When SessionStart do
  If (UM.User.Usergroup="admin") then AttachUserToPGroup ("admin")
  endif
  If (UM.User.Usergroup="user") then AttachUserToPGroup ("user")
  endif
endWhen
```

The OO-H approach assumes that the current user group can be queried from the user model (Garrigos and Gomez, 2006).

The *editLink* personalization rule specified in Listing 9 (Figure 35) is associated to the admin profile group (init section Figure 30) and ensures that the service link *Edit Activity* will be available to administrators but not to registered users or guests of the TIWA.

Customization Scenario Multi-Delivery. The *defSmallScreen* profile rule in Listing 10 (Figure 36) associates the user with the *smallScreen* group defined in the init section in Figure 30, if the device used to access the TIWA is either a PDA or a mobile phone (*deviceContext* defined in the *contextAcquisition* rule of Listing 5 (Figure 31)).

Multi-delivery can be realized also at a fine-grained level. If a PDA or a mobile phone is used to access the TIWA, pictures shall be omitted. The rule *ignoreImages* (Figure 37 Listing 11) applies to the *Hotel* and *Activity* nodes. As is defined by the above profile rule, it is attached to *smallScreen* group.

Customization Scenario Season's Style. Finally, the *summerStyle* and *winterStyle* personalization rules change the CSS used to the ones specified. The rules are attached to the summer and winter profile groups, respectively, (*init section* Figure 30) and will cause the presentation to be adapted depending on the current season. According to the *defSeason* profile rule, the user will either be attached to the summer or the winter group depending on the current date (Figure 38).

Depending on the group the user is associated with, one of the *summerStyle* and *winterStyle* personalization rules will be triggered and define an appropriate CSS style sheet to be used (Figure 39).

```
#RULE: "editLink" priority="medium"
When LoadElement.Activity(NM.Activity a) do
  not(hideLink(NM.EditActivity))
endWhen
```

Figure 35.
Listing 9 - OO-H:
administrator links
scenario (2)

```
#RULE: "defSmallScreen" priority="medium"
When SessionStart do
  if (UM.User.DeviceContext="PDA" or UM.User.DeviceContext="Mobile")
  then
    AttachUserToPGroup("smallScreen")
  endif
endWhen
```

Figure 36.
Listing 10 - OO-H:
multi-delivery scenario (1)

```
#RULE: "ignoreImages" priority="medium"
When LoadSet [NM.Hotel(NM.Hotel h) | NM.Activity(NM.Activity a)]
do
  not(h.Attributes.SelectAttribute(picture));
  not(a.Attributes.SelectAttribute(picture))
endWhen
```

Figure 37.
Listing 11 - OO-H:
multi-delivery scenario (2)

3.7 The object-oriented web solution approach (OOWS), Pastor et al.

3.7.1 *Maturity*. Object Oriented web Solutions (OOWS) (Pastor et al., 2006) is an approach that was first proposed in 2000 (Pastor et al., 2000) by extending OO-Method (Pastor et al., 1997) to support web modeling. Unlike the OO-H approach, OOWS is still based on the OO-Method. Recently, first results of the implementation of tool support for the OOWS approach have been published (Valverde et al., 2007). (G.T). The approach is presented with modeling examples like a university department site, a ticket ordering system, a book store, a travel agency system, and a university research group management system (G.ME). Also, the university department site has been realized on basis of OOWS (www.dsic.upv.es) (G.A).

3.7.2 *web modeling*. The OOWS method supports all software development phases including a so-called solution development phase in which models are translated into code (W.Ph). Developers are provided with appropriate modeling means in each of OOWS's proprietary four-step development process (W.Pr). For the requirements engineering phase, use-case diagrams, scenarios, and task descriptions are used (Valderas et al., 2005), (Torres et al., 2005). Task descriptions comprise task taxonomies, whereby each elementary task is described with an UML activity diagram, as well as information templates which are based on Class-Responsibility-Collaboration (CRC) cards. Thereafter a conceptual schema is built in the analysis phase. Since the approach is based on the OO-Method, only the models of the hypertext and presentation level are special in the OOWS original proposal. Then, this proposal has been extended to build (1) open and (2) business process aware web applications. The first extension consists in the introduction of the Services Model (Quintero et al., 2006). This model allows developers handling external data and functionality seamlessly during the development process. The latter consists in the introduction of a Business Process Model (Torres and Petechano, 2006) (which invokes functionality

```
#RULE: "deSeason" priority="medium"
When SessionStart do
  IF (UM.User.Currentdate >= 21.03. and UM.User.Currentdate < 21.09) then
    AttachUserToPGroup("summer")
  endIf
  IF (UM.User.Currentdate >= 21.09. and UM.User.Currentdate < 21.03) then
    AttachUserToPGroup("winter")
  endIf
endWhen
```

Figure 38.
Listing 12 – OO-H:
season's style scenario (1)

```
#RULE: "summerStyle" priority="medium"
When SessionStart do
  SetCSSTemplate ("summer.css")
endWhen

#RULE: "winterStyle" priority="medium"
When SessionStart do
  SetCSSTemplate ("winter.css")
endWhen
```

Figure 39.
Listing 13 – OO-H:
season's style scenario (2)

defined in the internal system and in the services model). The OO-Method includes different kinds of models (Fons et al., 2003) (W.Ph): On the content level, the information is captured using the structural model, i.e. a UML-like class diagram, just like in the OO-Method. In addition, UML state and sequence diagrams can be used to describe behavioral aspects and represent the dynamic model in OOWS (W.L). A functional model describes service effects in a textual formal specification language which is based on the OASIS formal language (www.oasis-open.org/specs/index.php) (Pastor and Ramos, 1995) as is described in Pastor et al. (2006). A navigational model comprises navigational maps which are used to define the global and structural aspects of the navigation, i.e. how so-called navigation contexts are related via navigational links with each other. Apart from that, the navigation contexts can be modeled using the navigation context diagram which allows designing the white-box view of a navigation context (Figure 41(b)). The navigational contexts comprise navigational classes which are views on the content level. The interface between content and hypertext level is graphically specified. The hypertext level and the presentation level are intermingled though: A presentation model only enriches the "in-the-small" model of the navigational context with presentational patterns concerning aspects such as scrolling or ordering of information (W.I). As already mentioned, only the content and the hypertext level can be enriched with behavioral diagrams while on all three levels structural modeling is supported (W.F).

3.7.3 *Customization modeling*. The only context properties addressed in OOWS literature seems to be the user property. In Abraham et al. (2002), more possible context properties are mentioned but they would have to be introduced by means of the standard OOWS models (C.P), (C.CE). Apart from that, no mechanisms for chronology (C.C) or complex context (C.CC) are available and since the context properties have to be included in the content level, there is neither separation of context (C.SC). Nevertheless, OOWS allows the definition of adaptations very early in the design process, namely during requirements specification (Rojas et al., 2006), i.e. within activity diagrams, until the design phase, i.e. within navigational contexts (C.CP). In particular, the different user types can be incorporated into the content model by merging the base content model with a user stereotype model. Furthermore, when specifying tasks by means of activity diagrams, the modeler can define adaptation rules such as link or content hiding as well as sorting of information according to previous user behavior (C.O). These adaptation operations are based on OCL conditions and are limited to fine-grained adaptations such as accessibility, filter, or sorting conditions (C.AE), (C.G). Complex adaptations cannot be realized (C.CA). The adaptation operations can be used to influence the hypertext level, only (C.L), (C.I). Moreover, customization in the OOWS approach requires the adaptations to be integrated into the models, allowing no separation of adaptation (C.SA).

3.7.4 *Model-driven engineering*. Recently, the OOWS language has been defined as a MOF-based metamodel and tool support on the basis of the Eclipse Graphical Modeling Framework (GMF) (www.eclipse.org/gmf) is under development (Valverde et al., 2007). Also a definition of the language in OWL has been proposed (Torres et al., 2004) (ML). OOWS supports an MDA-based approach including the transformation of platform-independent models. The possibility to generate code for the front-end of a web application is also currently under development within the above mentioned tool support (M.T). In order to generate the back-end of a web application, the commercial

tool OlivaNova (www.programmiermaschine.de) is used. In the future, the integration of the code generation for the back-end and the front-end is planned. A strategy of model (PIM) to code transformations (according to MDA) is applied, but no platform description models (PSM) are used (M.P).

3.7.5 *Modeling example*. In the following, three customization scenarios are presented, namely Customized Activities, Administrator Links, and Special Offers. Concerning the customization scenarios Multi-Delivery and Season's Style, it is not possible to cope with them in OOWS due to the fact that only the user context can be exploited for adaptation purposes.

Figure 40 shows the example's underlying class diagram, in which the different user stereotypes (i.e., Customer and Admin classes) are already integrated with the structural model of the TIWA. The User class and its sub-classes represent a hierarchy of possible user stereotypes of the TIWA. Following, for each scenario, the diagrams which are most relevant for customization in the OOWS approach are shown. These are, on the one hand, the activity diagrams from the requirements phase and, on the other hand, the resulting navigational models, in particular the navigational contexts, to present how the adaptation rules are incorporated into the hypertext.

Customization Scenario Customized Activities. In the OOWS approach, each type of user has a specific Navigational Map. As can be seen in the Navigational Map of Figure 41, an Anonymous User can navigate to the Activities Navigational Context through the Regions Navigational Context. This "Activities" Navigational Context provide the Anonymous user with all the existing activities.

A customer (registered user that needs to login) is a user specialization of Anonymous. This causes a customer to inherit the anonymous navigational map and its navigational contexts (see Figure 42). The navigational context "Activities" for a customer is a redefinition of the "Activities" navigational context for the anonymous user in order to satisfy the following adaptation expression: `self.RecAgeFrom > #Customer#.Age` and `self.RecAgeTo < #Customer#.Age`. This expression appears as the population condition of the Activity navigational class (see Figure 42(b)) to assure that only recommended activities will be provided to each customer. Using the keyword *self*, it is possible to access objects which should be displayed in this UML action (in this case, objects of the *Activity* class). With an OCL-like point-notation one

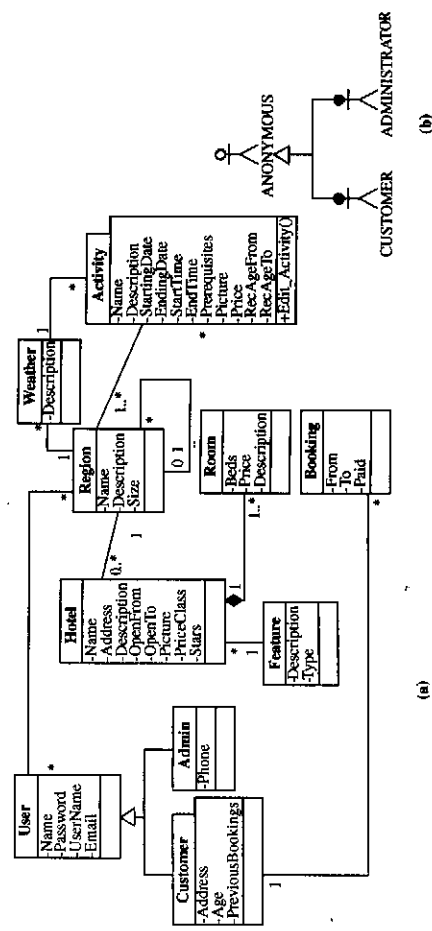


Figure 40. OOWS: (a) structural model; (b) user diagram model.

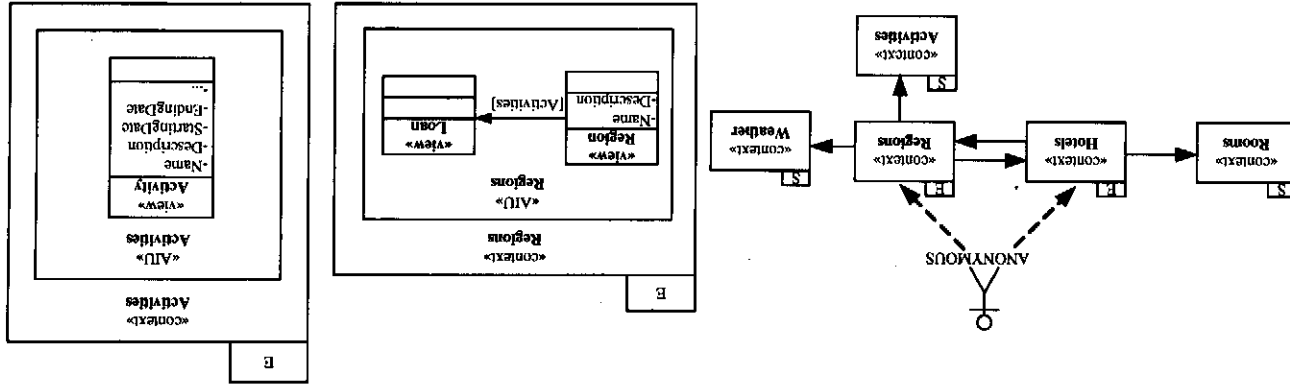


Figure 41. OOWS: anonymous (a) navigational map and (b) regions and activities navigational contexts

can navigate through the structural model and retrieve attribute values to be compared with values from the current context. One can access attribute values of the user with the #Customer# variable representing the current context.

Customization Scenario Administrator Links. For editing activities, administrators need to log into the web application and navigate to the list of activities for a certain region, as is specified in Figure 43(a). Having selected one activity, administrators are presented with a special link allowing them to edit the activity, as opposed to other normal users. This is ensured by defining a specialized navigational context from the anonymous Activities. This specialized context (Figure 43(b)) provides access to the *Edit_Activity()* operation for an Activity.

Customization Scenario Special Offers. This scenario is explained with special offers in the context of booking a hotel room. The activity diagram for this scenario is shown in Figure 44. It is depicted that, in the first step, a hotel has to be selected from an index of hotels (*action 1: Hotel*). The index is denoted with the "*" label associated to the action, while the "1" label indicates one instance to be displayed. Subsequently, the user can book a hotel, whereby the adaptation comes into play (*action 3: Book*). If the user has made more than three bookings in the past s/he gets a discount (3 per cent) provided by the *action 4: Get_Discount*. However, if the condition is not fulfilled, no discount is allowed.

Concerning the adaptation rule in the activity diagram, the resulting specification of the *GetDiscount()* operation is reflected in the functional description of the book operation in the Booking class (see Figure 30), following the OO-Method approach (Figure 45).

Customization Scenario Multi-Device. The OOWS method follows a fine-grained content adaptation approach to cope with multi-device customization (Torres *et al.*, 2007). For this purpose, it is necessary to (1) weigh all the data gathered in each Contextual Node and (2) specify the characteristics/limitations observed in the target output devices. On the one hand, weighting the data consists in organizing it in levels based on its relevance (Figure 46(a) where data has been group in two levels, where data from level 0 is more relevant than data from level 1).

On the other hand, the types of devices and their characteristics/limitations are kept in the *Device Model* (Figure 46(b)). This model defines categorized types of devices and their characteristics. No restrictions are applied when a device has no limitations defined (like the Desktop device). Taking into account all this new information, during

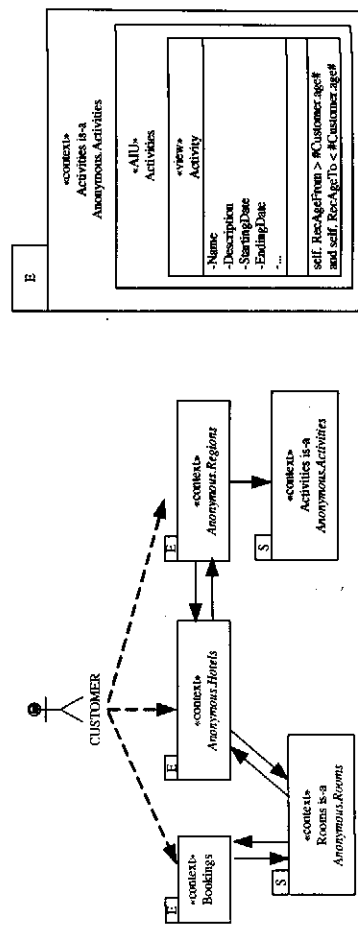


Figure 42. OOWS: customer (a) navigational map and (b) activities navigational context

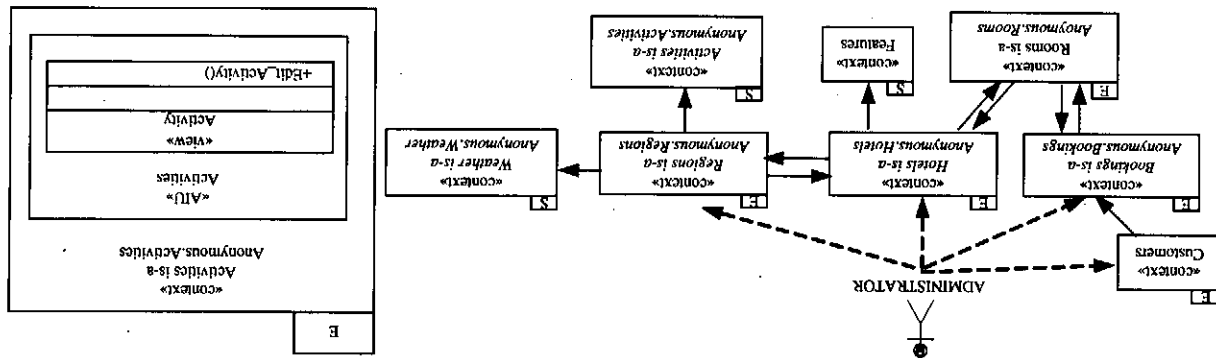


Figure 43. OOWS: administrator (a) navigational map and (b) activities navigational context

the code generation process we produce interfaces according to output device characteristics.

4. Lessons learned

In this section, the experiences acquired during the evaluation of the selected approaches as well as during modeling of the running example shall be summarized. In particular, it illustrates the results at a glance with comparison tables according to the five categories of criteria. In addition, we point out future directions for web modeling approaches to comprehensively address the development of ubiquitous web applications, although the surveyed approaches may emphasize diverse issues in the course of their further development, thus having a more restricted focus.

4.1 Maturity

Small Set of Similar Modeling Examples. A first overview on the web modeling field showed that the individual approaches often made use of similar modeling examples. Since stemming from academia, it is not surprising that a conference management system as well as some kind of department web site have been used particularly often to demonstrate a web modeling approach. Beyond this, different kinds of e-stores, e.g. selling books and CD's, as well as art gallery web applications have been used several times. The current set of modeling examples used in the web modeling field, however, raises some important questions: Are those examples complex enough to show the approaches' applicability? Do the provided examples cover all different kinds of web applications, e.g. ubiquitous web applications, workflow-based web applications the approaches try to address? Consequently, what the web modeling field needs is a set of generally acknowledged modeling examples in a public catalogue. Moreover, reference implementations for those examples can serve as "testbeds", i.e. to show if it is possible to develop such web applications with existing web modeling approaches. Additionally, it might be worthwhile thinking whether all the different approaches actually shall intend to cover the same types of applications or rather diversity with respect to specialize on specific web application types.

Rare Application in Real-World Projects. Web modeling approaches have already a ten-year-old history. Nevertheless, their application in real-world projects in particular in the context of a commercial setting is still rare. According to the survey of Lang *et al.*

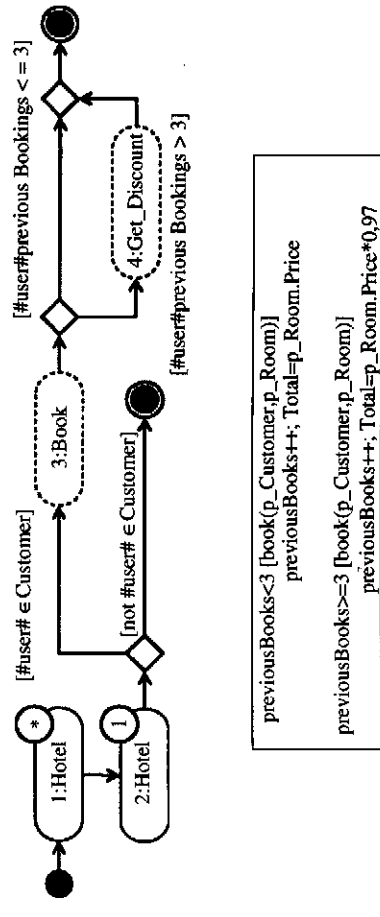


Figure 44.
OOWS: activity diagram
book hotel

Figure 45.
Book operation functional
description

Lang and Fitzgerald (2005) and our observations, in practice the awareness of academic methods is still rather low and consequently, they are rarely used by practitioners. To gain the desired impact on practitioners, there is an urgent need among all web application development approaches for more reference applications which are built with academic methods in order to demonstrate their maturity. Specifically, reaching beyond classical academic web application examples, towards more intensive cooperation with industry is necessary. Furthermore, besides presenting their approaches in terms of scientific publications, web modeling approaches will need to be presented in a way that is more suitable for practitioners in terms of detailed handbooks comprising the suggested models and concepts along with reference examples and guidelines how to apply the approach to reach them as target group.

Approaches are Continuously Evolving. All of the surveyed web modeling approaches have been presented in numerous publications including refereed papers, articles, books as well as manuals. Over the time, each of the approaches has been subject to extensions, e.g. for supporting business process modeling and customization modeling, as well as to major evolutions, e.g. the introduction of ontologies as a new formalism to specify models and the use of standards enabling model-driven Web engineering. From a practitioner's point of view, these developments might erratically communicate that current web modeling methods are not yet mature enough to be used in practice - maybe one reason why they are not yet used as extensively as one could desire. Additionally, some approaches provide alternative proposals for modeling customization for which it is not clear how they are related to each other and if they can be used in parallel. In this respect, a comprehensive description of the already mature parts of a web modeling approach in terms of a manual for designers might be beneficial (Figure 47).

4.2 web modeling

UML for Content Modeling - Proprietary Solutions for Hypertext Modeling. UML is quite popular for modeling the content level as well as the hypertext level of a web application. While UWE is based on UML 2.0 and OO-H is based on UML 1.x, OOHDH uses a UML-like notation. At hypertext level, however, UWE and OOHDH are the only approaches that continue using (stereotyped) UML class diagrams. Indeed, at hypertext level, one can find very different languages and notations for each individual approach. Although it is admitted that web specific concepts are important for the hypertext level and also the presentation level, not basing on a single (standard)

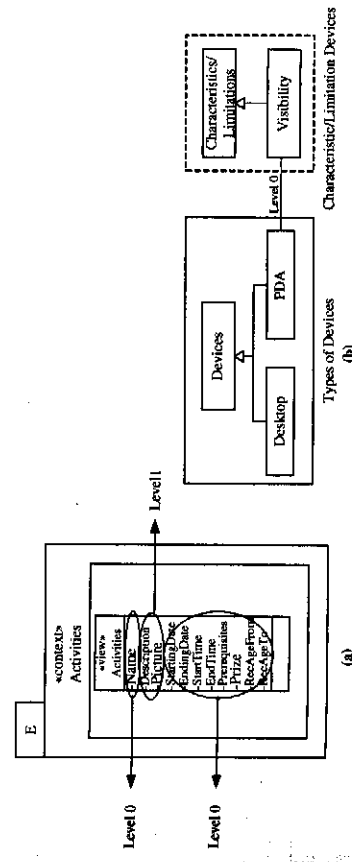


Figure 46.
OOWS: (a) grouping
context data by relevance,
(b) device model

formalism might hinder the application of solutions already provided in the model-engineering field like, e.g. model exchange, model transformation, or model checking. Additionally, probably a greater momentum could be gained when standards / standard formalisms are employed.

Presentation Level Seldomly Addressed. When it comes to the supported levels of web applications it becomes obvious that the presentation level is often only a marginal concern as already been diagnosed in Retschitzger and Schwinger (2000) some time ago. Although nearly every method provides presentation level support, it is, however, typically omitted in modeling examples. Interestingly, WebML is the only method not providing a presentation model at all, but leaves presentation concerns to the tool. With respect to the other approaches, although desirable when ubiquitous web applications need to be designed, nodes of the hypertext level are often mapped one-to-one to pages from the presentation level, thus not exploiting the chance to provide different presentations on bases of the same hypertext model. One exception is UWE which allows to design multiple presentation models for the same hypertext model.

Behavioral Modeling Not Comprehensively Considered. Behavioral modeling is typically not supported comprehensively for all levels. There is, however, a tendency to use behavioral diagrams, including use cases, activity diagrams, concurrent task trees in the requirements engineering and analysis phases, as well as for describing scenarios of user behavior with, e.g. sequence diagrams. Some form of behavioral modeling is also introduced by approaches providing support for business process modeling or workflow-based web applications. Ubiquitous web applications, however, encounter a greater need for addressing behavioral aspects than static web sites, thus a better support for behavioral modeling at all levels is desired.

Strong Processes. Almost all of the surveyed approaches support the developer with appropriate guidance to developing a web application from requirements engineering to implementation on the basis of their modeling techniques, i.e. the necessary steps, artifacts to be produced within each step, and actors are explained. While most of the approaches propose their own development process, WebML, OO-H and UWE do base their process on existing work, i.e. Boehm's Spiral model and RUP, respectively. Interestingly enough, all approaches start modeling the web application's design at the content level. Still, the applicability and usability of these processes need to be investigated in real-world projects raising in general demand to empirically evaluate the applicability of the approaches (Figure 48).

4.3 Customization modeling

Set of Context Properties Limited and Not-Extensible. All web modeling approaches do support customization with respect to the user context property, thus laying the path to

	WebML	Hera	WSDM	OOHDM	UWE	OO-H	OOWS
data-oriented	✓						
hypertext-oriented							
object-oriented							
Typicality (T)	1999-2007	2000-2007	1998-2006	1994-2006	1998-2007	2000-2007	2000-2007
Modeling Examples (ME)	7	5	3	5	6	6	5
Applications (A)	✓						

Legend:
 ✓ supported
 not supported
 partly supported
 n number of examples
 YYYY year of introduction/
 most recent publication

Figure 47.
Maturity

	WebML	Hera	WSDM	OOHDM	UWE	OO-H	OOWS
Levels							
Interfaces (I)							
Features (F)							
Phases (Ph)							
Development Process (P)							
Actors							
Support							
Legend:							
supported	✓	✓	✓	✓	✓	✓	✓
not supported							
n/a not applicable							
s/b structure / behavior							
n number of phases							
RUP Rational Unified Process							
ER Entry Relationship Diagram							
UML Unified Modeling Language							
CD Class Diagram							
SD Sequence Diagram							
SM State Machine Diagram							
OCL Object Constraint Language							
ORM Object Role Modeling							
RDFS Resource Description Framework Schema							
data-oriented	WebML	ER	RDFS	RDFS	ERM / OWL	UML-own	CD, SD, SM
hypertext-oriented							
object-oriented							
Levels							
Interfaces (I)							
Features (F)							
Phases (Ph)							
Development Process (P)							
Actors							
Support							
Legend:							
supported	✓	✓	✓	✓	✓	✓	✓
not supported							
n/a not applicable							
s/b structure / behavior							
n number of phases							
RUP Rational Unified Process							
ER Entry Relationship Diagram							
UML Unified Modeling Language							
CD Class Diagram							
SD Sequence Diagram							
SM State Machine Diagram							
OCL Object Constraint Language							
ORM Object Role Modeling							
RDFS Resource Description Framework Schema							

Figure 48.
Web modeling

personalization, while other context properties are often not taken into account. The investigation revealed that context properties typically are considered in isolation and that complex adaptations regarding several context properties are rare. Furthermore, the extension of the supported set of context properties is typically not discussed in current web modeling approaches. In this respect, the context model of OO-H represents the only exception considering user, location, device, time, and network context as well as allowing for their extension. Furthermore, except for the WebML and OO-H approaches, there is no support for explicit modeling concepts capturing context information. In OOWS, context is modeled as part of the content model (Rojas and Pelechano, 2005) allowing the consideration of user, location, device, time and network and may be extended by applying context description enhancements to the Class Diagram. Typically, it is assumed that context information is updated by some external service. Context captured comprehensively and in combination, however, is the important prerequisite to enable ubiquitous web applications and consequently should not be limited to some selected context properties, only.

Context Chronology and Complex Context not Addressed. Currently, none of the investigated web modeling approaches considers complex contexts as well as basing adaptations on historical context information. Consequently, adaptations have to be specified as a reaction to the sum of simple contexts. Concerning context chronology, some approaches allow for adaptations according to the user's navigation behavior. This historical information about the user is often implicitly available in terms of predicates of a rule language such as in WebML and OO-H. Still, other historical context information, such as the user's location over time, cannot be stored. Unless approaches allow for a proper representation of complex context, the full potential of ubiquitous web applications can only hardly be exploited properly.

Set of Adaptation Operations Limited and Not-Extensible. The set of adaptation operations that some web modeling approaches provide is limited to operations that can be performed upon typical concepts of a web application, including add/remove a link, change the style, add/remove a node, and sort some information. Currently, there seems to be no web modeling approach that supports operations on media types such as "resize image" or "shorten text". Of course, it has to be admitted that adaptation operations can be modeled in terms of normal behavior / actions. However, providing a set of generic adaptation operations to influencing the semantics of web applications that are applicable in a large range of application scenarios would facilitate reuse and also reduce the burden of the developer needing to model such adaptation operations.

Complex Adaptation Operations not Considered. Complex adaptations currently have been realized in the WSDM approach only. The Adaptation Specification Language provides an abstraction mechanism (scripts) to combine one or more primitive operations to form more complex operations. For example, the *promoteNode* and *demoteNode* operations can be used as such, and have been built upon primitive operations such as *add/remove link*. None of the other approaches provide the necessary modeling means that allow specifying complex adaptation operations on the basis of primitive ones. Likewise as before, complex adaptations allow to better support the modeler in what s/he is enabled to model in terms of concepts better suitable to the problem at hand.

Limited Support for Content, Presentation, and Interface Adaptations. Adaptations at the hypertext level are considered within all of the investigated approaches. With

respect to content and presentation level only half of the approaches provide necessary adaptation operations, while interface adaptation is supported by WSDM and OOHDM, only. More interestingly, approaches supporting presentation adaptations rather operate at a coarse-grained level, e.g. by providing adaptation operations that change the complete style of the web applications presentation. In contrast, the Hera approach allows defining alternative layout managers for parts of the presentation model, thus realizing adaptation at a more fine-grained level (Figure 49).

Customization not Comprehensively Considered in all Development Phases. Customization modeling is predominantly considered during design, the exceptions being the OOWS and the WSDM approaches. It seems that customization is treated as a separate step in the design phase in which an existing web application is extended with customization issues. Instead, customization needs to be considered during all phases in the software development lifecycle. Consequently, web modeling approaches might want to adapt their current development processes in order to appropriately include customization concerns in all phases.

Disregarded Crosscutting Nature of Customization. Current web modeling languages do not allow for modeling customization with the desired separation from the rest of a web application model. More specifically, this is due to the missing separation of context and adaptation. Considering separation of context, in general, all web modeling approaches acknowledge the need for defining context information in a separated model. Still, guidelines supporting a developer in constructing a context model are often not available. Some guidance can be found in the WebML approach as well as in the OO-H framework for modeling context information. Nevertheless, none of the approaches currently is able to achieve full separation of context from the rest of the web application models, i.e. typically, concepts from the context model are connected to concepts from the content model in some way or the other, e.g. via associations. As another example, it is often not decidable if a certain concept shall be modeled within the content or the context model, such as the user concept of which some attributes might contribute to the core functionality of the web application and some others contribute to customization. With respect to separation of adaptation, developers typically are required to define adaptations as annotations to existing models. In the UWE, OO-H, and Hera approaches, adaptations at the hypertext level can be captured separately from the rest of the web application models, however. ASL specifications in WSDM are also completely separate from the (regular) design. Thus, customization functionality is not intermingled with the rest of the web application.

	Context					Adaptation				
	Properties (P)	Network	Identity (ID)	Chronology (C)	Complex Context (CC)	Levels (L)	Interactivity (I)	Adaptation	Phases (CP)	
WebML	x	x	x	x	x	x	x	mi/ma	x	x
Hera	x	x	x	x	x	x	x	mi/ma	x	x
WSDM	x	x	x	x	x	x	x	mi/ma	x	x
OO-H	x	x	x	x	x	x	x	mi/ma	x	x
UWE	x	x	x	x	x	x	x	mi/ma	x	x
OO-H	x	x	x	x	x	x	x	mi/ma	x	x
OOWS	x	x	x	x	x	x	x	mi	x	x

Legend:
 x supported
 o not supported
 mi/ma micro/macro

Figure 49.
Customization modeling

Nevertheless, there is a need for comprehensively capturing customization from all levels of a web application. Furthermore, it is important to know where adaptations do take effect in models, meaning a modeling language needs to specify these subjects of adaptations. This is particularly interesting to know since customization support adds an additional level of complexity to the models. While this is supported by UWE's and Hera's aspect-oriented approaches to modeling customization for the hypertext level, in OO-H the same information is captured within the *Event* and *Condition* parts of its rule language. Concluding, aspect-orientation seems to represent a suitable mechanism for separately capturing customization. But up to now, aspect-orientation has not been used comprehensively for all levels in a web application model. For example, at the content level aspect-orientation can be used to separate those parts (e.g. attributes) of the concept "user" within a separate aspect that represent context information, while the application-specific parts of the concept "user" remain in the content model.

4.4 Model-driven engineering criteria

From Notations to Languages. Most web modeling approaches have emerged, rather focusing on notations than on using standards for specifying their language. Today, with the rise of model-driven engineering, the semantic web and the general need to produce a running system from the web application models, more and more web modeling approaches provide formal specifications of their languages in terms of either metamodels, UML-profiles, or ontologies (Figure 50).

Model Transformations not Based on MDE Standards. Model transformations are supported by almost all approaches in one way or another. Still, only the UWE Web modeling language has recently been extended to better support model transformations in the sense of MDE, i.e. through providing ATL and QVT transformations. The employment of standards for model transformations would allow benefiting from existing transformation engines of the MDE field.

Lack of Platform Description Models. None of the approaches provides platform description models and consequently no model transformations from platform-independent to platform-specific models are supported. The employment of platform description models together with transformation techniques could broaden the base of application platforms to be employed and if realized within the accompanying tool, could give the approaches a broader application base.

4.5 Tool support

Lack of (Extensible) Tool Support. One of the most problematic issues in web modeling is the lack of tool support for the individual approaches. Without proper tool support,

Tool	Language Definition (ML)		Model Transformation Type (MT)				GETC/POSTC supported not supported
	XML	DTD	UML	UML2	UML2	UML2	
WANG							
FACE							
WETM							
OCBDM							
UWE							
OC-H							
COUWS							

Figure 50. Model-driven engineering

however, web modeling methods will not gain acceptance in practice. From the set of surveyed approaches, only four provide tool support that has been made available to the public community. Out of the four tools, WebRatio and VisualWade represent the only commercial tools having left the status of a prototype implementation. None of the available tools is offered under an open source license, which would attract developers of open source web frameworks and technologies. Although ArgoUWE, the tool accompanying the UWE method, has been built upon the open source tool ArgoUML, the extensions made are not open source themselves. WebRatio is the only tool offering some built-in extension mechanism for the WebML language, i.e. a plug-in mechanism for so-called custom units.

Customization Modeling not Supported by Tools. Currently, all tools provide support for basic web modeling with some deviations from the original notation, however. Still, modeling support for dealing with customization is only provided by Hera's tool which allows modeling context information within user profiles used to statically generate adapted hypermedia presentations. For the customization modeling extensions of WebML and OO-H, prototype implementations have been reported on in literature but are not made available yet.

Support for Model Transformations Provided. Although not based on MDE standards, all tools offer some form of model transformations, e.g. for generating a hypertext model from the content model, or a presentation model from a hypertext model (ArgoUWE, WebRatio, and VisualWade), or for integrating different models as a prerequisite for code generation (HPG).

Code Generation for a Single Platform Only. Except for the ArgoUWE tool, all approaches provide code generation support. Still, code generation is limited to specific platforms, only. While VisualWade generates PHP code, WebRatio targets J2EE platforms by producing code for the Struts framework. WebRatio is shipped with a Tomcat Servlet Container and provides for simple deployment of the web application. The HPG tool of Hera, however, does support generating static hypermedia presentations in several formats, including HTML as well as WML. As a consequence, the employment of a web modeling tool also determines the runtime platform which probably is not desired by the developers (Figure 51).

Limited Process Support. Remarkably, only the HPG tool of Hera in combination with its Model Builders implements the process as described by the supported method. All other approaches have only partial tool support for their defined processes. Typically, the tools do not offer means for supporting developers in producing artifacts (e.g. use cases) from earlier development phases but start with designing the content model. WebRatio allows going back and forth between different models as well as making several changes to either of them without loss of previously modeled artifacts. Instead, ArgoUWE as well as OO-H require the user to follow the web applications' "levels dimension" in first designing the content, then the hypertext, (whereby OO-H's tool does permit the designer to go back and forth between these two models) and finally the presentation level, in a linear fashion.

Lack of Collaboration Facilities in Tools. Collaborative work on web application development is currently insufficiently supported. WebRatio, however, as a first step in that direction, foresees the employment of a CVS.

5. Related surveys

In an effort to shed light on the different approaches to web application development, some surveys have already been presented. In the following, these surveys are distinguished according to their specific goals and foci into closely related work representing customization modeling surveys and more widely related work representing web modeling surveys focusing on general web modeling criteria, development processes, requirements engineering, and support for modeling rich internet applications.

5.1 Customization modeling surveys

Barna *et al.* provide a comparison of four approaches, amongst them Hera, OOHDM, and UWE, also investigated in the present survey (Barna *et al.*, 2003). The approaches are investigated according to their specific design models for content, hypertext, and presentation levels as well as their support for customization design, though the focus is rather on personalization. The discussion is supported using a simple running example of a virtual art gallery.

In a further evaluation but already some time ago, Kappel *et al.* (2001) compare the customization modeling capabilities of OOHDM and WebML with respect to supported context, granularity of adaptations, and the degree of customizability.

Similar to the above mentioned evaluations, this survey's focus is on investigating the support of modeling customization in current web modeling languages but in contrast, specifically considers also the model-driven development of UWAs including tool support. Besides this difference in goals, this survey is also different in terms of comprehensiveness by surveying seven recent web modeling approaches which provide means for customization modeling and if available their tool support. Furthermore, the evaluation is supported with a running example consisting of five different customization scenarios which is used to better explain the general modeling concepts of each approach and in particular the provided means for customization modeling. The evaluation is based on a well-defined as well as fine-grained catalogue of more than 30 criteria, which allows a detailed investigation of each approach with respect to general web modeling characteristics, customization modeling characteristics, as well as model-driven engineering and tool support. In contrast, the above mentioned related work provide less than five criteria or no explicit description in terms of a catalogue at all.

5.2 web modeling surveys

General web Modeling. In Schwinger and Koch, an introduction into modeling web applications is given, including a brief overview of eleven web modeling approaches based on a set of twelve criteria, amongst them one criterion evaluating support for customization modeling as well as code generation (Schwinger and Koch, 2006). In contrast, this survey focuses on web modeling approaches providing support for modeling customization. It also differs in that this survey applies a more detailed criteria catalogue as well as a running example while it includes the relevant approaches also surveyed in Schwinger and Koch (2006).

Development Processes. Nora Koch has evaluated eleven approaches in Koch (1999), amongst those only OOHDM and WSDM that are still evolving. That survey specifically focuses on the approaches' development process, supported development

Tool	Version (T.V)	Modeling Support (T.M)	Customization Modeling (T.CM)	Web Modeling Support (T.WMS)	Code Generation (T.CG)	Consistency Check (T.CC)	Model Pre-Generation (T.MG)	Process of Method (T.PM)	Process Type (T.PT)	Collaboration (T.CO)	Phases (T.PH)	Wizard (T.WZ)	Commercial (T.COM)	Freeware (T.FRE)	Supported (T.SUP)	Not supported (T.NOSUP)	Partly supported (T.PARTSUP)	n/a (T.NA)	
WebML: Web Rule	4.3 (acad.)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Hera: Hera Presentation Generator	1.3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
UWE: ArgUWE	0.16	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OO-H: Virtual Wale	1.2 rev 163	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Legend:

✓	supported
✗	not supported
◐	partly supported
◑	not applicable
com	commercial
fre	freeware
sup	supported
od	on demand
wm	while modeling

Figure 51.
Tool support

phases, modeling techniques and notations used, as well as tool support. In the end, the characteristics of the Rational Unified Process (RUP) (Kruchten, 2000) are presented as well as a discussion on how some approaches support parts of RUP. Customization, however, is not a focus as in the survey of that work.

In Woukeu *et al.* eight approaches are investigated, having in common with this survey the WebML, OOHDM, and WSDM web modeling approaches (Woukeu *et al.*, 2003). The evaluation is focused on the supported development process, i.e. their phases, as well as the modeling techniques used. Furthermore, each approach's concepts from the hypertext level are listed. Finally, each approach is evaluated if it allows to model read-only or read-write web applications so that the survey differs considerably in terms of focus.

Our survey, aiming at a fine grained set of criteria, has adopted some of the criteria in the works of Nora Koch and Woukeu *et al.* Furthermore, where appropriate, the criteria have been endowed with a clear definition including a measurement scale or they have been refined. Such a refinement generally means the decomposition of a criterion into several criteria.

Requirements Engineering. In Escalona *et al.*, the scope is requirements engineering for web applications (Escalona and Koch, 2004). A comparison of ten web modeling approaches is provided including WSDM, OOHDM, UWE, and WebML, which are also investigated within this survey. In particular, the types of requirements, the activities and techniques employed during requirements elicitation, specification and validation, and the methodologies' focus on the requirements process, techniques, or artifacts have been evaluated.

In contrast to the work of Escalona *et al.*, this survey is rather concerned with the design level means of today's web modeling languages. Nevertheless, the surveyed approaches are investigated with respect to their support of a requirements engineering phase and more particularly, if customization modeling is already considered during requirements engineering. Consequently, this evaluation is complementary to the one of Escalona *et al.*

Support for Rich Internet Applications. Preciado *et al.* compare fifteen representatives from web modeling, multimedia and hypermedia methodologies according to their applicability to model rich internet applications (Preciado *et al.*, 2005). Again, five of the approaches are also evaluated within this survey, namely, UWE, OO-H, WebML, WSDM, and OOHDM. The set of ten evaluation criteria include multimedia modeling, personalization modeling, and tool support are rated according to a weighted measurement scale consisting of four degrees of coverage. Again this survey's focus is different to the one of Preciado *et al.* which investigates their selection of approaches concerning their applicability to model rich internet applications. Customization modeling, however, is only a marginal concern supported in the work of Preciado *et al.* with one criterion, only, which in this survey is evaluated in much more detail.

Model-driven web Engineering Methods in the Context of MDA. Moreno *et al.* have recently presented an overview of the current state of Model-Driven Software Development, and of Model-Driven web Engineering (MDWE) in particular, especially in the context of MDA (Moreno *et al.*, 2007b). They analyzed the main concerns (or *viewpoints*) that need to be addressed by any MDWE approach, and evaluated how they are tackled by seven of the existing proposals (namely OOHDM, W2000, UWE, WebML, WSDSM, OOWS and OO-H). The objective of that comparison analysis was

to examine the possible definition of interoperability bridges among these proposals, so that they can seamlessly exchange models and tools. Apart from introducing the advantages and opportunities that MDA can bring to MDWE, the authors also discussed the current problems and threats that MDWE and MDA face for its successful adoption in industrial settings. Again, customization was just one of the concerns discussed in Moreno *et al.* (2007b), which has been analyzed here in this paper in much more detail.

6. Summary

This paper has presented how existing web application methods support a model-driven development of ubiquitous web applications. More specifically, an in-depth comparison of seven web modeling approaches currently supporting the development of ubiquitous web applications has been provided. An evaluation framework has been designed on the basis of a detailed and well-defined catalogue of evaluation criteria. The actual evaluation by means of this criteria catalogue is supported by a modeling example, i.e. a tourism information web application, used to provide an initial insight into each approaches' concepts for modeling customization as well as to facilitate their comparability. More specifically, a set of five customization scenarios has been defined, to be modeled with each approach. The per-approach evaluation is complemented with a report on lessons learned, summarizing the approaches' strengths and shortcomings, thus pointing towards possible future developments. This survey points out limitations of current web modeling languages with respect to the model-driven development of ubiquitous web applications lacking of a proper MDE foundation in terms of metamodels as well as missing tools allowing to model customization. Furthermore, the proposed customization mechanisms are often limited, since they neither cover all relevant context factors in an explicit, self-contained, and extensible way, e.g. within a dedicated context model, nor allow for a wide spectrum of extensible adaptation operations. Furthermore, the provided customization mechanisms frequently do not allow dealing with all different parts of a web application in terms of its content, hypertext, and presentation levels as well as their structural and behavioral features. Finally, current web modeling approaches may be extended to better consider the crosscutting nature of customization by not providing the necessary means to comprehensively capture customization separately from all levels of a web application model.

Notes

1. Candido Portinari (December 29, 1903 - February 6, 1962) was amongst one of the most important painters of Brazil and also a prominent and influential practitioner of the neo-realism style in painting.
2. Please note that OOHDM's notion of context is different to the notion of context in the realm of customization used in this thesis.

References

- Abraham, S.M., Fons, J., Gonzalez, M. and Pastor, O. (2002), "Conceptual modeling of personalized web applications", *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)*, Malaga, Spain, LNCS 2347, pp. 358-62.

- Barna, P., Frasinca, F. and Houben, G.-J. (2002), "Specification framework for engineering adaptive web applications", *Proceedings of the 11th International World Wide web Conference, web Engineering Track (WWW 2002)*, Honolulu, Hawaii, USA.
- Barna, P., Frasinca, F. and Houben, G.-J. (2006), "A workflow-driven design of web information systems", *Proceedings of the 6th International Conference on web Engineering (ICWE 2006)*, Palo Alto, CA, USA, ACM, pp. 321-8.
- Barna, P., Frasinca, F., Houben, G.-J. and Vdovjak, R. (2003), "Methodologies for web information system design", *Proceedings of the International Conference on Information Technology: Computers and Communications (ITCC 2003)*, Las Vegas, NV, USA, IEEE Computer Society, pp. 420-4.
- Baumeister, H., Koch, N. and Mandel, L. (1999), "Towards a UML extension for hypermedia design", *Proceedings of the 2nd International Conference on the Unified Modeling Language (UML 1999)*, Fort Collins, CO, USA, LNCS 1723, pp. 614-29.
- Baumeister, H., Knapp, A., Koch, N. and Zhang, G. (2005), "Modelling adaptivity with aspects", *Proceedings of the 5th International Conference on web Engineering (ICWE 2005)*, Sydney, Australia, LNCS 3579, pp. 406-16.
- Bozzon, A., Comai, S., Fraternali, P. and Carughi, G.T. (2006), "Conceptual modeling and code generation for rich internet applications", *Proceedings of the 6th International Conference on web Engineering (ICWE 2006)*, Palo Alto, CA, USA, ACM, pp. 353-60.
- Brambilla, M., Cabot, J. and Moreno, N. (2007), "Tool support for model checking of web application designs", book chapter in *web Engineering, Lecture Notes in Computer Science*, Vol. 4607/2007, pp. 533-8.
- Brambilla, M., Ceri, S., Fraternali, P. and Manolescu, I. (2006b), "Process modeling in web applications", *ACM Trans. Softw. Eng. Methodol.*, Vol. 15 No. 4, pp. 360-409.
- Brambilla, M., Celino, I., Ceri, S., Cerizza, D., Valle, E.D. and Facca, F.M. (2006a), "A software engineering approach to design and development of semantic web service applications", *Proceedings of the 5th International Semantic web Conference (ISWC 2006)*, Athens, GA, USA, LNCS 4273, pp. 172-86.
- Cachero, C., Gomez, J. and Pastor, O. (2000), "Object-oriented conceptual modeling of web application interfaces: the OO-H method abstract presentation model", *Proceedings of the 1st International Conference on Electronic Commerce and web Technologies (EC-web 2000)*, London, UK, LNCS 1875, pp. 206-15.
- Cachero, C., Gomez, J. and Parraga, A. (2001), "Migration of legacy systems to the web", paper presented at VI Jornadas de Ingenieria del Software y Bases de Datos (ISBD 2001), Almagro (Ciudad Real), pp. 601-14.
- Cachero, C., Gomez, J., Parraga, A. and Pastor, O. (2001), "Conference review system: a case of study", paper presented at 1st International Workshop on Web-Oriented Software Technology (IWWOST 2001), Valencia.
- Casteleyn, S. (2005), "Designer specified self re-organizing websites", PhD dissertation, Vrije Universiteit, Brussel.
- Casteleyn, S., De Troyer, O. and Brockmans, S. (2003), "Design time support for adaptive behavior in web sites", *Proceedings of the 18th Symposium on Applied Computing (SAC 2003)*, Melbourne, FL, USA, pp. 1222-8.
- Casteleyn, S., Garrigos, I. and De Troyer, O. (2005), "Automatic runtime validation and correction of the navigational design of web sites", paper presented at 7th Asia-Pacific web Conference on web Technologies, Research and Development (APweb 2005), Shanghai, pp. 453-63, LNCS 3399.

- Casteleyn, S., Plessers, P. and De Troyer, O. (2006), "On generating content and structural annotated websites using conceptual modeling", *Proceedings of the 25th International Conference on Conceptual Modeling (ER 2006)*, Tucson, AZ, USA, LNCS 4215, pp. 267-80.
- Casteleyn, S., Woensel, W.V. and Houben, G.-J. (2007), "A semantics-based aspect-oriented approach to adaptation in web engineering", *Proceedings of the 18th Conference on Hypertext and Hypermedia (HT 2007)*, Manchester, UK, pp. 189-98.
- Ceri, S., Daniel, F. and Matera, M. (2003a), "Extending WebML for modeling multi-channel context-aware web applications", *Proceedings of the Workshop on Mobile Multi-channel Information Systems, in Conjunction with WISE 2003*, Rome, Italy, pp. 615-26.
- Ceri, S., Daniel, F. and Facca, F.M. (2006), "Modeling web applications reacting to user behaviors", *Computer Networks*, Vol. 50 No. 10, pp. 1533-46.
- Ceri, S., Daniel, F., Facca, F.M. and Matera, M. (2007a), "Model-driven engineering of active context-awareness", *World Wide Web*, Vol. 10 No. 4.
- Ceri, S., Daniel, F., Matera, M. and Facca, F.M. (2007b), "Model-driven development of context-aware web applications", *ACM Transactions on Internet Technology*, Vol. 7 No. 1.
- Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S. and Matera, M. (2003b), *Designing Data-intensive web Applications*, 1st ed., Morgan Kaufmann, San Francisco, CA.
- Chen, P.P. (1976), "The entity-relationship model - toward a unified view of data", *ACM Transactions on Database Systems*, Vol. 1 No. 1, pp. 9-36.
- Cowan, D.D. and de Lucena, C.J.P. (1995), "Abstract data views: an interface specification concept to enhance design for reuse", *IEEE Trans. Software Eng.*, Vol. 21 No. 3, pp. 229-43.
- De Troyer, O. and Casteleyn, S. (2004), "Designing localized web sites", *Proceedings of the 5th International Conference on web Information Systems Engineering (WISE 2004)*, Brisbane, Australia, LNCS 3306, pp. 547-58.
- De Troyer, O. and Leune, C.J. (1998), "WSDM: a user centered design method for web sites", *Computer Networks*, Vol. 30 Nos 1/7, pp. 85-94.
- De Troyer, O., Casteleyn, S. and Plessers, P. (2005), "Using ORM to model web systems", paper presented On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops, International Workshop on Object-Role Modeling (ORM'05), LNCS 3762.
- De Troyer, O., Casteleyn, S. and Plessers, P. (2007), "WSDM: web semantics design method", in Rossi, Gustavo, Pastor, Oscar, Schwabe, Daniel and Olsina, Louis (Eds), *web Engineering: Modeling and Implementing web Applications, Human-Computer Interaction Series*, Springer, Berlin.
- Escalona, M.J. and Koch, N. (2004), "Requirements engineering for web applications - a comparative study", *Journal of web Engineering*, Vol. 2 No. 3, pp. 193-212.
- Fiala, Z., Hinz, M., Houben, G.-J. and Frasinca, F. (2004a), "Design and implementation of component-based adaptive web presentations", *Proceedings of the Symposium on Applied Computing (SAC 2004)*, Nicosia, Cyprus, pp. 1698-704.
- Fiala, Z., Frasinca, F., Hinz, M., Houben, G.-J., Barna, P. and Meißner, K. (2004b), "Engineering the presentation layer of adaptable web information systems", *Proceedings of the 4th International Conference on web Engineering (ICWE 2004)*, Munich, Germany, LNCS 3140, Vol. 3140, pp. 459-72.
- Fons, J., Pelechano, V., Albert, M. and Pastor, O. (2003), "Development of web applications from web enhanced conceptual schemas", *Proceedings of the 22nd International Conference on Conceptual Modeling (ER 2003)*, Chicago, IL, USA, LNCS 2813, pp. 232-45.

- Frasincar, F. and Houben, G.-J. (2002), "Hypermedia presentation adaptation on the semantic web", *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)*, Malaga, Spain, LNCS 2347, pp. 133-142.
- Frasincar, F., Houben, G.-J. and Barna, P. (2006), "HPG: the Hera presentation generator", *Journal of web Engineering*, Vol. 5 No. 2, pp. 175-200.
- Frasincar, F., Houben, G.-J. and Vdovjak, R. (2001), "An RMM-based methodology for hypermedia presentation design", *Proceedings of the 5th East European Conference on Advances in Databases and Information Systems (ADBIS 2001)*, Vilnius, Lithuania, LNCS 2151, pp. 323-37.
- Frasincar, F., Barna, P., Houben, G.-J. and Fiala, Z. (2004), "Adaptation and reuse in designing web information systems", *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2004)*, Las Vegas, Nevada, USA, pp. 287-91.
- Garrigos, I. and Gomez, J. (2006), "Modeling user behaviour aware websites with PRML", paper presented at 3rd Workshop on web Information Systems Modelling (WISM 2006), in conjunction with ICWE 2006, Palo Alto, CA.
- Garrigos, I., Castelyn, S. and Gomez, J. (2005a), "A structured approach to personalize websites using the OO-H personalization framework", *Proceedings of the 7th Asia-Pacific web Conference on web Technologies Research and Development (APweb 2005)*, Shanghai, China, LNCS 3399, pp. 695-706.
- Garrigos, I., Cruz, C. and Gomez, J. (2007), "A prototype tool for the automatic generation of adaptive websites", *Proceedings of the 2nd International Workshop on Adaptation and Evolution in web Systems Engineering (AEWSE 2007)*, in conjunction with ICWE 2007, Como, Italy. CEUR Workshop Proceedings.
- Garrigos, I., Gomez, J. and Cachero, C. (2003a), "Modelling adaptive web applications", *Proceedings of the IADIS International Conference WWW/Internet, Algarve, Portugal*, pp. 813-6.
- Garrigos, I., Gomez, J. and Cachero, C. (2003b), "Modelling dynamic personalization in web applications", *Proceedings of the 3rd International Conference on web Engineering (ICWE 2003)*, Oviedo, Spain, LNCS 2722, pp. 472-5.
- Garrigos, I., Gomez, J., Barna, P. and Houben, G.-J. (2005b), "A reusable personalization model in web application design", paper presented at 2nd Workshop on web Information Systems Modelling (WISM 2005), in conjunction with ICWE 2005, Sydney.
- Gomez, J., Bia, A. and Parraga, A. (2005), "Tool support for model-driven development of web applications", paper presented at 6th International Conference on web Information Systems Engineering (WISE 2005), New York, NY, USA, pp. 721-30, LNCS 3806.
- Gomez, J., Cachero, C. and Pastor, O. (2000), "Extending a conceptual modelling approach to web application design", *Proceedings of the 12th International Conference on Advanced Information Systems Engineering (CAISE 2000)*, Stockholm, Sweden, LNCS 1789, pp. 79-93.
- Gomez, J., Cachero, C. and Pastor, O. (2001), "Conceptual modeling of device-independent web applications", *IEEE MultiMedia*, Vol. 8 No. 2, pp. 26-39.
- Grün, Ch., Schwinger, W., Pröll, B., Retschitzegger, W. and Werthner, H. (2006), "Pinpointing tourism information onto mobile maps - a light-weight approach", *Proceedings of ENTER 2006*, 18-20 January 2006, Lausanne, Switzerland.
- Halpin, T. (2001), *Information Modeling and Relational Databases*, Morgan Kaufmann, San Francisco, CA.

- Hester, A., Borges, R. and Ierusalimsky, R. (1998), "Building flexible and extensible web applications with Lua", *Journal of Universal Computer Science*, Vol. 4 No. 9, pp. 748-62.
- Houben, G.-J., Frasincar, F., Barna, P. and Vdovjak, R. (2004), "Modeling user input and hypermedia dynamics in Hera", *Proceedings of the 4th International Conference on web Engineering, Munich, Germany*, LNCS 3140, Springer, pp. 60-73.
- Isakowitz, T., Stohr, E.A. and Balasubramanian, P. (1995), "RMM: a methodology for structured hypermedia design", *Communications of the ACM*, Vol. 38 No. 8, pp. 34-44.
- Jacyntho, M.D., Schwabe, D. and Rossi, G. (2002), "A software architecture for structuring complex web applications", *J. web Eng.*, Vol. 1 No. 1, pp. 37-60.
- Kappel, G., Pröll, B., Retschitzegger, W., Schwinger, W. and Hofer, T. (2001), "Modeling ubiquitous web applications - a comparison of approaches", *Proceedings of the 3rd International Conference on Information Integration and Web-based Applications & Services (IIWAS 2001)*, Linz, Austria.
- Kappel, G., Pröll, B., Reich, S. and Retschitzegger, W. (2006), "An introduction to web engineering", in Kappel, G., Pröll, B., Reich, S. and Retschitzegger, W. (Eds), *web Engineering - Systematic Development of web Applications*, Wiley, New York, NY.
- Kappel, G., Pröll, B., Retschitzegger, W. and Schwinger, W. (2003), "Customisation for ubiquitous web applications a comparison of approaches", *International Journal of web Engineering and Technology*, Vol. 1 No. 1, pp. 79-111.
- Koch, N. (1999), "A comparative study of methods for hypermedia development", Tech. Rep. 9905, Ludwig-Maximilians-University, Munich.
- Koch, N. (2001), "Software engineering for adaptive hypermedia systems: reference model, modeling techniques and development process", PhD thesis, Ludwig-Maximilians-University, Munich.
- Koch, N. (2007), "Classification of model transformation techniques used in uml-based web engineering", *IET Software*, Vol. 1 No. 3, pp. 98-111.
- Koch, N. and Kraus, A. (2002), "The expressive power of uml-based web engineering", *Proceedings of the 2nd International Workshop on Web-oriented Software Technology (IWOST 2002)*, in conjunction with ECOOP 2002, Malaga, Spain, pp. 21-32.
- Koch, N. and Kraus, A. (2003), "Towards a common metamodel for the development of web applications", *Proceedings of the 3rd International Conference on web Engineering (ICWE 2003)*, Oviedo, Spain, LNCS 2722, pp. 497-506.
- Koch, N. and Wirsing, M. (2002), "The munich reference model for adaptive hypermedia applications", *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)*, Malaga, LNCS 2347, pp. 213-22.
- Koch, N., Kraus, A., Cachero, C. and Melia, S. (2004), "Integration of business processes in web application models", *Journal of web Engineering*, Vol. 3 No. 1, pp. 22-49.
- Koch, N., Zhang, G. and Escalona, M.J.E. (2006), "Model transformations from requirements to web system design", *Proceedings of the 6th International Conference on web Engineering (ICWE 2006)*, Palo Alto, CA, USA, ACM, pp. 281-8.
- Kraus, A., Knapp, A. and Koch, N. (2007), "Model-driven generation of web applications in UWE", *Proceedings of the 3rd International Workshop on Model-Driven web Engineering (MDWE 2007)*, CEUR-WS, Vol. 261.
- Kruchten, P. (2000), *The Rational Unified Process: An Introduction*, Addison-Wesley, Reading, MA.
- Lang, M. and Fitzgerald, B. (2005), "Hypermedia systems development practices: a survey", *IEEE Software*, Vol. 22 No. 2, pp. 68-75.

- Manolescu, I., Brambilla, M., Ceri, S., Comai, S. and Fraternali, P. (2005), "Model-driven design and deployment of service-enabled web applications", *ACM Transactions on Internet Technology*, Vol. 5 No. 3, pp. 439-79.
- Moreno, N., Fraternali, P. and Vallecillo, A. (2007a), "WebML modelling in UML", *IET Software*, Vol. 1 No. 3, pp. 67-80.
- Moreno, N., Romero, J.R. and Vallecillo, A. (2007b), "An overview of model-driven development and the MDA", in Rossi, G., Pastor, O., Schwabe, D. and Olsina, L. (Eds), *web Engineering: Modelling and Implementing web Applications*, Chapter 12, Springer, Berlin, pp. 353-82.
- Nunes, D.A. and Schwabe, D. (2006), "Rapid prototyping of web applications combining domain specific languages and model driven design", *Proceedings of the International Conference on web Engineering (ICWE 2006)*, Trento, Italy, pp. 153-60.
- OMG, O. (2002), *Meta Object Facility (MOF) Specification 1.4*, available at: www.omg.org/docs/formal/02-04-03.pdf
- OMG, O. (2005), *Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification*. Final Adopted Specification. ptc/05-11-01.
- OMG, O.M.G. (2003), *MDA Guide Version 1.0.1*, available at: www.omg.org/docs/omg/03-06-01.pdf
- Pastor, O. and Ramos, I. (1995), *OASIS version 2 (2.2): A Class-Definition Language to Model Information Systems*, 3rd ed., Publications Services of the Valencia University of Technology, Valencia.
- Pastor, O., Abrahao, S. and Fons, J. (2000), "OOWS: an object-oriented approach for web solutions modeling", *Proceedings of the International Conference on Information Society (ICIS 2000)*, Ljubljana, Slovenia.
- Pastor, O., Fons, J., Pelechano, V. and Abrahao, S. (2006), "Conceptual modelling of web applications: the OOWS approach", in Mendes, E. and Mosley, N. (Eds), *web Engineering: Theory and Practice of Metrics and Measurement for web Development*, Springer, Berlin, pp. 277-302.
- Pastor, O., Insfran, E., Pelechano, V., Romero, J. and Merseguer, J. (1997), "OO-method: an OO software production environment combining conventional and formal methods", *Proceedings of the 9th International Conference on Advanced Information Systems Engineering (CAISE 1997)*, Barcelona, Catalonia, Spain, LNCS 1250, pp. 145-58.
- Paterno, F. (2000), "Model-Based Design of Interactive Applications", *ACM Intelligence*, Vol. 11 No. 4, pp. 26-38.
- Plessers, P., Casteleyn, S. and De Troyer, O. (2005a), "Semantic web development with WSDM", *Proceedings of the 5th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot2005)*, in conjunction with ISWC 2005, November 6-10, Galway, Ireland. CEUR Workshop Proceedings.
- Plessers, P., Casteleyn, S., Yesilada, Y., De Troyer, O., Stevens, R., Harper, S. and Goble, C. (2005b), "Accessibility: a web engineering approach", *Proceedings of the 14th International World Wide web Conference (WWW2005)*, Eds. Allan Ellis, Tatsuya Hagiwara, Publ. ACM.
- Preciado, J.C., Trigueros, M.L., Sanchez, F. and Comai, S. (2005), "Necessity of methodologies to model rich internet applications", *Proceedings of the 7th IEEE International Workshop on web Site Evolution*, Budapest, Hungary, pp. 7-13.
- Quintero, R., Torres, V. and Pelechano, V. (2006), "Model centric approach of web services composition", *Proceedings of the Workshop on Emerging web Service Technology (WEWST 2006) at the 4th IEEE European Conference on web Services (ECOWS 2006)*, Zurich, Switzerland, pp. 65-82.

- Retschitzegger, W. and Schwinger, W. (2000), "Towards modeling of data web applications a requirements' perspective", *Proceedings of Americas Conference on Information Systems (AMCIS 2000)*, Long Beach, USA.
- Rojas, G. and Pelechano, V. (2005), "A methodological approach for incorporating adaptive navigation techniques into web applications", paper presented at VI International Conference on web Information Systems Engineering - WISE 2005, New York, NY, USA, November 20-22.
- Rojas, G., Valderas, P. and Pelechano, V. (2006), "Describing adaptive navigation requirements of web applications", *Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2006)*, Dublin, Ireland, LNCS 4018, pp. 318-22.
- Rossi, G. and Schwabe, D. (2006), "Model-based web application development", in Mendes, E. and Mosley, N. (Eds), *web Engineering: Theory and Practice of Metrics and Measurement for web Development*, Springer, Berlin, pp. 203-333.
- Rossi, G., Schwabe, D. and Guimaraes, R. (2001), "Designing personalized web applications", *Proceedings of the 10th International World Wide web Conference (WWW 2001)*, Hong Kong, China, pp. 275-84.
- Rossi, G., Pastor, O., Schwabe, D. and Olsina, L. (Eds) (2007), *web Engineering: Modelling and Implementing web Applications*, Human-Computer Interaction Series, Springer, Berlin.
- Rossi, G., Schwabe, D., de Lucena, C.J.P. and Cowan, D.D. (1995), "An object-oriented model for designing the human-computer interface of hypermedia applications", *Proceedings of the International Workshop on Hypermedia Design (IWHHD 1995)*, Montpelier, France, pp. 123-43.
- Schauerhuber, A., Schwinger, W., Wimmer, M., Retschitzegger, W. and Kappel, G. (2007a), "A survey on web modeling approaches for ubiquitous web applications", Tech. rep., Vienna University of Technology, October.
- Schwabe, D. and Rossi, G. (1994), "From domain models to hypermedia applications: an object-oriented approach", paper presented at International Workshop on Methodologies for Designing and Developing Hypermedia Applications, Edinburgh.
- Schwabe, D. and Rossi, G. (1998), "An object oriented approach to web-based applications design", *Theory and Practice of Object Systems*, Vol. 4 No. 4, pp. 207-25.
- Schwabe, D. and Salim, C.S. (2002), "Integrating knowledge management applications in the enterprise the xerox knowledge portal project", *Knowledge and Process Management*, Vol. 9 No. 3, pp. 190-201.
- Schauerhuber, A., Wimmer, M., Kapsammer, E., Schwinger, W. and Retschitzegger, W. (2007b), "Bridging WebML to model-driven engineering: from document type definitions to meta object facility", *IET Software*, Vol. 1 No. 3, pp. 81-97.
- Schwabe, D., de Almeida Pontes, R. and Moura, I. (1999), "OOHDM-web: an environment for implementation of hypermedia applications in the WWW", *ACM SIGWEB Newsletter*, Vol. 8 No. 2, pp. 18-34.
- Schwabe, D., Guimaraes, R. and Rossi, G. (2002), "Cohesive design of personalized web applications", *IEEE Internet Computing*, Vol. 6 No. 2, pp. 34-43.
- Schwabe, D., Rossi, G. and Barbosa, S.D.J. (1996), "Systematic hypermedia application design with OOHDM", *Proceedings of the 7th Conference on Hypertext (HT 1996)*, Washington DC, pp. 116-28.

Further reading

- Barry, C. and Lang, M. (2003), "A comparison of 'traditional' and multimedia information systems development practices", *Information & Software Technology*, Vol. 45 No. 4, pp. 217-27.
- Schwinger, W. (2001), "Modelling ubiquitous web applications - requirements and concepts", PhD thesis, Department of Information Systems, JKU, November 2001.
- Authors and affiliations**
- Wieland Schwinger, Department of Telecooperation, Johannes Kepler University Linz, Austria.
 Werner Retschitzegger, Johannes Kepler University Linz, Austria.
 Andrea Schauerhuber, Gerti Kappel and Manuel Wimmer, Business Informatics Group, Vienna University of Technology, Austria.
 Birgit Pröll, Institute for Application Oriented Knowledge Processing, Johannes Kepler University Linz, Austria.
 Cristina Cachero Castro, University of Alicante, Spain.
 Sven Casteleyn and Olga De Troyer, Department of Computer Science - WISE, Vrije Universiteit Brussel, Belgium.
 Piero Fraternali, Department of Electronics and Information, Polytechnic University of Milan, Italy.
 Irene Garrigos, Department of Languages and Information Systems, University of Alicante, Spain.
 Franca Garzotto, Department of Electronics and Information, Polytechnic University of Milan, Italy.
 Athula Ginige, School of Computing and IT, University of Western Sydney, Penrith South, DC, Australia.
 Geert-Jan Houben, Department of Software Technology, Delft University of Technology, The Netherlands.
 Nora Koch, Institute of Computer Science, Ludwig-Maximilians-University Munich, Germany.
 Nathalie Moreno, University of Malaga, Spain.
 Oscar Pastor, OO-Method Group, Polytechnic University of Valencia, Spain.
 Paolo Paolini, Department of Electronics and Information, Polytechnic University of Milan, Italy.
 Vicente Pelechano Ferragud, OO-Method Group, Polytechnic University of Valencia, Spain.
 Gustavo Rossi, Computer Science College, La Plata National University, Buenos Aires, Argentina.
 Daniel Schwabe, Department of Information Technology, Pontificia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brazil.
 Massimo Tisi, Faculty of Design, Polytechnic University of Milan, Italy.
 Antonio Vallecillo, Department of Languages and Sciences of Computing, University of Malaga, Spain.
 Kees van der Sluijs, Department of Mathematics and Computer Science, Technical University of Eindhoven, The Netherlands.
 Gefei Zhang, Institute of Computer Science, Ludwig-Maximilians-University Munich, Germany.

To purchase reprints of this article please e-mail: reprints@emeraldinsight.com
 Or visit our web site for further details: www.emeraldinsight.com/reprints

- Schwabe, D., Szundy, G., Silva de Moura, S. and Lima, F. (2004), "Design and implementation of semantic web applications", *Proceedings of the WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web*.
- Schwinger, W. and Koch, N. (2006), "Modeling web applications", in Kappel, G., Pröll, B., Reich, S. and Retschitzegger, W. (Eds), *web Engineering - Systematic Development of web Applications*, Wiley, New York, NY, pp. 39-64.
- Torres, V., Giner, P. and Pelechano, V. (2007), "Modeling ubiquitous business process driven applications", in Eder, J., Tomassen, S.L., Opdahl, A.L. and Sindre, G. (Eds), *Forum Proceedings*.
- Torres, V. and Pelechano, V. (2006), "Building business process driven web applications", in Dust-dar, S., Fiadairo, J.L. and Sheth, A. (Eds), *Business Process Management*, 4th International Conference, BPM 2006. Lecture Notes in Computer Science, Vol. 4102, Springer, Berlin/Heidelberg, pp. 322-37.
- Torres, V., Fons, J., Pelechano, V. and Pastor, O. (2004), "Navigational modeling and the semantic Web: an ontology based approach", *Proceedings of the Joint Conference 10th Brazilian Symposium on Multimedia and the web & 2nd Latin American web Congress, (WebMedia & LA-web 2004), Ribeirao Preto-SP, Brazil*, IEEE Computer Society, pp. 94-6.
- Torres, V., Pelechano, V., Ruiz, M. and Valderas, P. (2005), "A model driven approach for the integration of external functionality in web applications. The travel agency system", *Proceedings of 1st Workshop on Model-Driven web Engineering, in conjunction with ICWE 2005*, Sydney, Australia, pp. 1-11.
- van der Sluijs, K., Houben, G.-J., Broekstra, J. and Casteleyn, S. (2006), "Hera-S - web design using sesame", *Proceedings of the 6th International Conference on web Engineering (ICWE 2006)*, Palo Alto, CA, USA.
- Valderas, P., Fons, J. and Pelechano, V. (2005), "Using task descriptions for the specification of web application requirements", *paper presented at Anais do WER05 - Workshop em Engenharia de Requisitos, Porto, Portugal*, pp. 257-68.
- Valverde, F., Valderas, P., Fons, J. and Pastor, O. (2007), "A MDA-based environment for web applications development: from conceptual models to code", paper presented at International Workshop on Web-oriented Software Technology (IWWOST 2007), in conjunction with ICWE 2007, Como.
- Vdovjak, R. and Houben, G.-J. (2002), "Providing the semantic layer for WIS design", *Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAISE 2002)*, Toronto, Canada, LNCS 2348, pp. 584-99.
- Vilain, P., Schwabe, D. and de Souza, C.S. (2000), "A diagrammatic tool for representing user interaction in UML", *Proceedings of the 3rd International Conference on the Unified Modeling Language (UML 2000)*, York, UK, LNCS 1939, pp. 133-47.
- W3C (2004a), *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0*, available at: www.w3.org/TR/CCPP-struct-vocab/
- W3C (2004b), *RDF Primer (XML) 1.1*, 2nd ed., available at: www.w3.org/TR/rdfprimer/
- W3C (2004c), *OWL web Ontology Language Overview*, available at: www.w3.org/TR/owl-features/
- Woukeu, A., Carr, L., Wills, G. and Hall, W. (2003), "Rethinking web design models: requirements for addressing the content", Tech. Rep. ECSTR-IAM03-002, University of Southampton, Southampton.
- Zhang, G. (2005), "Towards aspect-oriented class diagrams", *Proceedings 12th Asia-Pacific Software Engineering Conf. (APSEC'05)*, IEEE Computer Society, pp. 763-8.