MODELING WEB INFORMATION SYSTEMS FOR CO-EVOLUTION

Buddhima De Silva and Athula Ginige
School of Computing and Mathematics, University of Western Sydney, Sydney, Australia

Keywords: Co-evolution, web information system, end-user development.

Abstract: When an information system is introduced to an organisation it changes the original business environment thus changes the original requirements. This can lead to changes to processes that are supported by the information system. Also when users get familiar with the system they ask for more functionality. This gives rise to a cycle of changes known as co-evolution. One way to facilitate co-evolution is to empower end-users to make changes to the web application to accommodate the required changes while using that web application. This can be achieved through meta-design paradigm. We model web applications using high level abstract concepts such as user, hypertext, process, data and presentation. We use set of smart tools to generate the application based on this high-level specification. We developed a hierarchical meta-model where an instance represent a web application. High level aspects are used to populate the attribute values of a meta-model instance. End-user can create or change a web application by specifying or changing the high level concepts in the meta-model. This paper discusses these high level aspects of web information systems. We also conducted a study to find out how end-users conceptualise a web application using these aspects. We found that end-users think naturally in terms of some of the aspects but not all. Therefore, in meta-model approach we provide default values for the model attributes which users can overwrite. This approach based on meta-design paradigm will help to realise the end-user development to support co-evolution.

1 INTRODUCTION

In traditional approach to develop web information systems, when an organization finds the need for new information system, the requirements are analysed. Then if they go for a custom made information system the design specification is produced and the system gets developed tested and deployed. Otherwise organization may buy a product to match their requirements and adapt their processes accordingly. The system then needs to be maintained until decommissioned. However, we have found that many such systems that were deployed with the university and client organizations after a period of time no longer meet the user requirements. The failure is due to 3 reasons:
1) When an information system is introduced to an organisation it changes the original business environment thus changing the original requirements.
2) The processes that information system that supporting may change.
3) When users get familiar with the system they ask for more functionality.

This gives rise to a cycle of changes known as co-evolution (Costabile et al., 2006). Meta-design is proposed as a solution to the co-evolution (Costabile et al., 2006, Costabile et al., 2005). Meta-design paradigm characterises objectives, techniques, and processes for creating new media and environments allowing ‘owners of problems’ (that is, end-users) to act as designers (Fischer and Giaccardi, 2004, Fischer et al., 2004). A fundamental objective of meta-design paradigm is to create socio-technical environments that empower users to engage actively in the continuous development of systems rather than being restricted to the use of existing systems. If we are to empower end users to actively participate in development tasks then, they should be provided with a suitable web application development environment. Rode’s study on mental model of end-user developers reveals that end-users do have no or very little concerns about the critical issues in web applications such as authentication,

De Silva B. and Ginige A. (2007).
MODELING WEB INFORMATION SYSTEMS FOR CO-EVOLUTION.
In *Proceedings of the Second International Conference on Software and Data Technologies - Volume ISDM/WsEHST/DC*, pages 54-61
DOI: 10.5220/0001341600540061

session management, etc.(Rode and Rosson, 2003). Therefore we identify the need to bridge the gap between end-user mental model and developer's mental model for the success of meta-design paradigm. This gives rise to following 2 requirements:
1) End-users need to model a web application using high level aspects adequate to specify a web application.
2) Then we have to provide a set of tools and a framework that they can use to develop and change the application by specifying and changing these aspects.

Thus rather than developing "the application" for end-users to use we need to develop a meta-model to represent various aspects of applications and a set of tools which the end-users can use to create the applications that they want by populating the instance values in the meta-model. Then end-users can populate or change the values relating to various aspects according to their requirements to instantiate or change the web application. We have created a meta-model for web based information systems to support the meta-design paradigm (Ginige and De Silva, 2007).

We analysed a set of requirement specifications end-users written for different information systems within the University to identify how they specify web applications. In that study we found that end-users tend to specify aspects of information systems at a conceptual level. Therefore, we reviewed the literature on conceptual modelling of web applications to find different aspects required to model web applications at a conceptual model. We found that many conceptual models are proposed to model a specific type of web application such as data-intensive web applications, process intensive web applications. From these, we derived a set of aspects required to define web conceptual model for any type of web application. Then we refined the set of aspects by modelling different types of web applications. These systems varied from simple web sites to e-commerce web sites. We have excluded the special types of web applications such as e-mail or chat applications which are optimised for specific functionality. We found that there are 2 broader categories of web applications; information centric, and process centric.

- Information centric: simple web sites with unstructured information. The focus is on effective presentation of information.
- Process Centric: web sites that support business processes by enabling users to perform actions such as filling a form, approving a form etc. These can further be divided into 2 types;
  - Data intensive: The focus is on efficient presentation on structured data such as product catalogue.
  - Workflow intensive: The focus is on efficient automation of business process consisting of sequence of steps such as order processing system.

In this paper, we discuss the 'holistic' web conceptual model to define different types of web information systems. This will answer the research question; how can we specify Web Information Systems? We use WebML, Web modelling language with extensions to model these aspects. Then for end-users we provide techniques such as visual tool, program by example to model these aspects. In section 2, we review the existing web conceptual models to identify the required aspects to model web applications. Section 3 presents the different aspects of web information systems. In section 4, we analyse end-user specifications to find out the concepts they use to define these aspects. Section 5 concludes the paper.

2 RELATED WORK

Most of the web modeling languages developed at early days such as HDM-Lite (Fratenali and Paolini, 1998), UWE (Koch and Kraus, 2002), WebRE (Escalona and Koch, 2006), W2000 (Baresi et al., 2001), OOHDM (Baresi et al., 2001) and initially WebML (Ceri et al., 2000) emphasized aspects required to model data intensive web applications such as data and hypertext. All these approaches provided data models to define the entities used and relationships between them, navigation & composition models to organize and to access pages and presentation model to define look and feel. WebML consists of models such as structural model (i.e. data model), hypertext model (composition navigation models), presentation models, personalised models and operational models. The operational model is providing a way to model functions in a data intensive web application. Modelling in WebML is based on visual notations which are simple and complete.

Later some of these languages such as WebML (Ceri et al., 2000), UWE (Koch and Kraus, 2002) and OOHDM (Baresi et al., 2001) were extended to model process intensive web applications. WebML was integrated with Business Process Modellign

Language (BPML) to model workflows in web applications. In OOHDM (Schwabe et al., 1996), the conceptual model is derived based on Object Oriented Modelling (OMT) principles. It is defined using classes and relations. The navigation model is also modelled as two classes, the navigation class schema and navigation context schema. State charts are used to define the browsing semantics. The interfaces are defined using configuration diagrams.

HDM-Lite (Fratenali and Paolini, 1998) uses structure schema, navigation schema and presentation schema to model web applications.

Two types of use cases exist in UWE (Koch and Kraus, 2002) approach called navigation and functional. Use cases are explained using behavioural diagrams or textual format.

Recently, Koch et.al have modelled the conceptual model of web application based on the web application behaviour and structure (Escalona and Koch, 2006, Koch et al., 2006). Their Requirement meta-model is focused on navigation and presentation of information. Therefore, business process is modelled as a kind of navigation. This type of generalization leads to complex models of web applications.

Jakob, et.al. (Jakob et al., 2006) and Schimid et.al (Schmid and Rossi, 2004) have identified the problem with most of these web application models as neglecting the business processes at the early stages of modeling. Jakob, et.al. (Jakob et al., 2006) use “Operational model” to model the business logic aspect of data intensive applications. However, their operational model is in the logical level, not in the conceptual level. They are more focused on application generation not on requirement elicitation. In Kobti et.al.’s Conceptual framework (Kobti and Sundaravadanam, 2006), they use WebML to model the hypertext aspects of web applications and Web Service Choreography Description Language (WS-CDL) to model the processes. In this approach they model the business process independent of hypertext modeling language. But the transaction nature of WS-CDL won’t be sufficient to define complex business processes. Also it doesn’t provide a mechanism to model the access control.

Olveira et.al (De Oliveira et al., 2001) have modeled the process-centric web applications using state models. They have modelled Access control as conditions associated with states and navigation as state transition. They have associated views with states. Even though this is an attempt to model the web application at logical level we see the state chart can be used to model the state dependant behaviour of use cases.

However, most of these modelling approaches have embedded the aspects such as process model and user model within other aspects rather than defining these separately during the analysis phase of web applications. Also we have identify the possibility of abstracting composition model at a higher level than the granular elements such as label, text, list box for process intensive web applications.

3 ASPECTS OF WEB CONCEPTUAL MODEL

We have analysed modelling of different web applications from information-centric and process-centric categories. Based on our finding we have revised WebML conceptual modelling aspects to enhance the naturalness and completeness. We believe these modifications will help end-user developers to efficiently develop or modify their web applications. We have identified 5 high level aspects required to completely define any type of web application called Hypertext model, User model, Process model, Data model and Presentation model.

- Hypertext model: Hypertext model consists of two sub models: composition and navigation. Navigation model defines how to get in to a specific page and how to go through a group of pages in the web application. Thus navigation defines the sequence of views presented to the user when user is interacting with the application. For example possible sequence of views present in the e-commerce application is customer details view, order detail view, payment detail view. Composition model defines the collection of elements in the view which help users to interact with web application. For example the composition model of a “log in” page consists of log in instruction provided at the top of the page, user name and password elements submit and cancel buttons.
- User Model: The underline structure in user model is that users belong to groups. User model consists of two sub models called access control and personalisation models. Access control model defines what functions user group can access in an information system. For example prospective buyer can access the shopping cart function on an e-commerce application. But only users who are waiting for delivery of an order can access their order data. Personalisation model defines the attributes for the user that can be used to personalise the web

application at group level and user level. When a user logged on in the same e-commerce application the list of products a user see on a personalized home page can be based on user profile.

- Data Model: Data model defines relevant objects and relationships among objects used in the web application. Data defines the domain objects stored and retrieved through the web application. For example in the above e-commerce application we have to store product information such as name, type, price, availability, available units, etc. We may also want to store order information such as customer name, product quantity, payment amount, etc. We may want to keep relationships between order and product to maintain stock management.
- Process Model: Process model defines high level abstraction of tasks and workflows to be performed in a web application. When modelling web applications sometimes it is necessary to model some routing tasks that should happen when a user interact with the application. Typically this is expressed in a use case such as use case description of "request for more information" in an e-commerce application. These functionalities can be executed independent of the status of any other processes. Details of these functionalities are captured in a Task Model. On the other hand there may be workflows in web applications to sequence series of tasks to carry out a business process. Some examples are order processing and stock management. In a workflow, when a task needs to be performed may depend on the status of some other tasks. The order in which tasks need to be performed is captured in the Workflow model.
- Presentation: Presentation model captures the layout and graphic appearance of all generic elements appearing on the views of the web application. This is modelled at two levels in a web information system: site level, and page level. At site level we define the look and feel of the web application. For example in the e-commerce site we may define the size and style of the fonts used in labels and style, size and color for the instructions etc. At page level we define the same attributes only applicable to a specific page.

We find that some of these aspects can be specified independently. For example data aspect can be specified using objects and relationships among the objects. However, other aspects such as composition model and process model depend on data model to find out the objects or object attributes which associate with user interfaces or tasks. The process model also depends on composition model for actions which trigger states and data model for data updates. Presentation aspect depends on personalisation sub model of the user model. Navigation model depends on access control sub model of the user model to find out access rights of the user when authorisation is required. These dependencies are shown in figure 1.

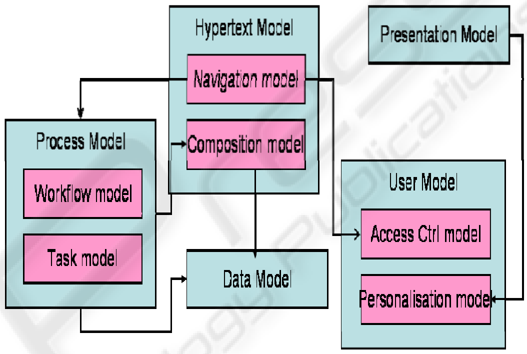

Figure 1: Dependencies between aspects .

3.1 Hypertext Model

Hypertext model consist of composition and navigation models. Navigation can be defined explicitly using menus or implicitly with in the composition model in a web information system. We have to define navigation for two different types of tasks: state dependant, and state independent. In an e-commerce application the tasks such as "view product catalogue", "order" are available for all users. Also views such as "home page" or "contact us" pages are available for all users. Then there can be tasks such as "view order summary" which is available for authorised users. These tasks which are always available for users are categorised as state independent tasks. These tasks can be executed independent of the state of any processes. Once a user log in, user will be provided with a menu to access state independent tasks. On the other hand, the "view order status is an example for state dependant functions. These tasks are available to users only if an order waiting for processing for that particular user at that time. Navigation model can identify views to sequence from the access control model.
Composition model defines the user interfaces of a web application. There are three different types of Composition models called unstructured, form, and

table. Unstructured page can contain content such as images, links and, text. Form or table model consists of UI guide, UIElementGroup, and or UIElements, and UI Actions. UI elements can be WebML data units, which store or present data. UI elements can be in input mode or output mode. In a form we have UI elements in Input mode, i.e. called entry units in WebML. In a table we can have UI elements in output mode. It can be WebML index, single data, multi data, scroll data or search data unit. UIGuide provide the guidelines to use the particular interface. For example, in a form interface, the guidelines can help the users to understand the purpose of the form. In a table interface, the guidelines can help the users to interpret the report properly. UI is associated with a primary business object. For a UI element in input mode, we can also have associated help tip. Help tips help users to enter the values of the form UI element correctly. We extend WebML entry unit by adding help information for each field of entry unit. Sometimes, it is required to logically group UIElements. For example a product order may include more than one product. The data of the each product order can consist of quantity of product, price of product. Thus, product details can be in a UIElement group of the order UI. UI Actions such as add product, amend product actions can be associated with that UIElement group. Order UI Model can have UI Action to process the order.

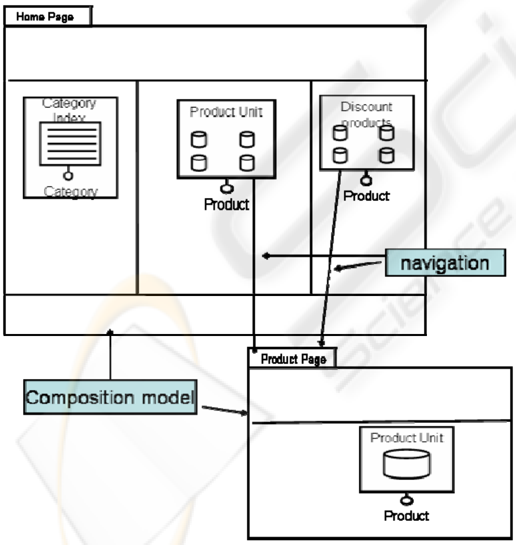

Figure 2: Example Hypertext model from Amazon.

Figure 2 shows the partial hypertext model for amazon site. The composition model consists of table of products. We model table with WebML multi data unit as shown in figure 2.

3.2 Presentation Model

Presentation model specifies the look and feel of a web application. It dictates the layout and graphic appearance of all generic elements appeared on a page. We specify the presentation model at two levels: site level and page level. The template and style settings of the presentation model at site level apply to all the web pages of the application. If we want to model a custom page we have to define a template and style settings at page level.

If we consider personalisation then presentation model has to get the values of the presentation model for the user from the user model. Figure 3 shows the site level template for amazon. Then for some pages it may have page level templates.

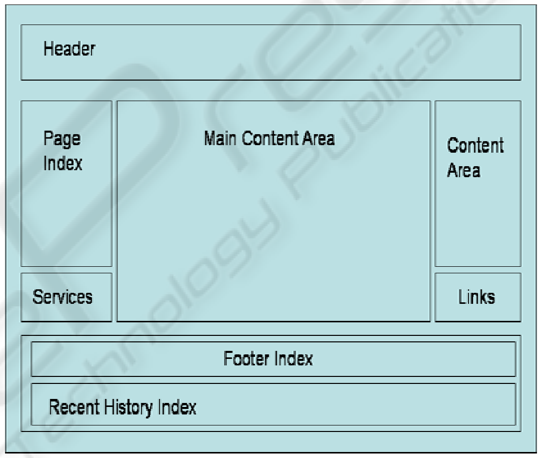

Figure 3: Site level Template for Amazon.

3.3 User Model

User model is based on user, group concept. User can belong to many groups. Group can have one or more users. Group can access one or more tasks or pages using access control model. “Public” is a special group which, has access to all tasks and pages which doesn’t require authentication. “Admin” is another special group which has access to all the tasks and pages.

We can store values for presentation model at user level or group level. These personalised values are used by presentation model to generate views for given user. We have extended WebML conceptual model by adding a user model. We identify the need to define access control explicitly at conceptual level by end-user developers since this is required to generate the applications. We also identify possibility for expressing personalisation model with in the user model. Since user has attributes and relations similar to object we use class diagram to model users. User model is a special data model because it is common in any web information system. Figure 4 shows the class diagram for user model to define the access control and personalisation sub model.

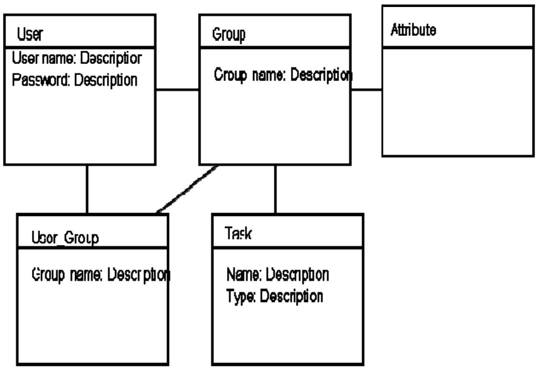

Figure 4: user model.

3.4 Process Model

Process model consist two different types of processes: task related and workflow related. We define business logic included in processes and this determines the behaviour of web application. In other words in a web information system, the business rules govern what happens next when a task is performed. Presence of process aspect in an Information-centric web application is minimum compared to a process-centric web application. Jacob et. Al. (Jakob et al., 2006) defines the minimum operations a data intensive web application should support as to Add new content objects or new relations between content objects, alter existing content objects or existing relations between content objects, delete existing content objects or existing relations between content objects, filter content objects according to conditions and sort content objects by specified criteria. There can also be tasks such as send notification associate with data-intensive web applications.

In the e-commerce example when a customer submits an order, order processing officer should be able to see the order information. Conditional rules can also be associated to the flow. We define the flow of information in the workflow model. Workflow model defines the state, entry condition, exit condition, transition, transition activity, etc. Transition also associated with a task and UI action. A simplified process model for order processing system is shown in figure 5. User can place an order. Then order processing officer process order.

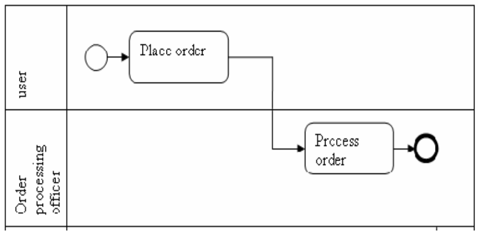

Figure 5: Simplified workflow model for order processing.

3.5 Data Model

Data Model defines the relevant objects and relationships among objects used in the web information system. Data model is the key concept in data-intensive web applications. Objects have attributes. We define high level attribute types such as photo, e-mail ,etc as defined in Smart Business Object Modelling Language (SBOML) (Liang and Ginige, 2006). This high level abstract attribute types will help to improve the naturalness in modelling data aspect. The partial data model for amazon is shown in figure 6. There are 3 different kinds of products such as book, software, electronic. Product can has discount during a given time frame. Product can have 0 or more reviews.

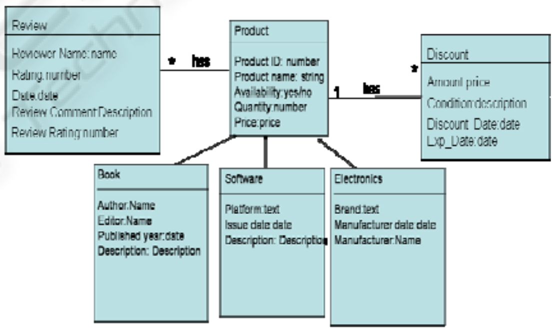

Figure 6: Partial Data model.

4 STUDY ON END-USER SPECIFICATIONS

We analysed 32 proposals submitted by users in administration positions of an academic organisation to develop or modify existing information systems. From these 32 proposals 17 were for new Information Systems or enhancements to existing Information Systems. Rest were for new hardware or hardware upgrades. We further analysed the software proposals to identify the concepts they have used. 12 out of 17 users have specified objects and some attributes for the objects. In most cases users had a manual system. Therefore they have an understanding of data that needs to be managed. No one has specified navigation personalisation or presentation models. Only one has specified the composition model. 50% have specified processes. 80% has specified user model with access control and some attributes for user groups. Usage of concepts as a percentage of 17 is shown in table 1.

Table 1: Covered Aspects in end-user specifications.

Aspect	Usage
Data	70%
Hypertext	4%
Presentation	0%
Process	50%
User	80%

This is only a sample of end-users. However, this sample of end-users also confirms Rode's conclusion on end-user mental model. They want systems to store data and then manipulate data. They are not concern about other aspects such as presentation and hypertext. All of that will come as usability issues when they start to use the system. However, when they become the owners of the system they need to create or modify the system. Then they also need to consider the other aspects such as hypertext model. When designing the tools for end-users to develop web information systems we have to incorporate default values for these aspects in presentation model and navigation model. This will help to match the tools to end-user mental model.

5 CONCLUSION

This paper presents high-level aspects of web information systems that can be used by end-users to specify the web applications that they want. These high level aspects can be used in end-user development of web information systems to support co-evolution. In our study of end-user specifications we found that some of the aspects such as data, and processes are naturally specified by end-users while some of the aspects such as navigation and composition models are rarely specified. Even for the aspects such as data they hardly define the relationships between objects. These missing aspect specifications are necessary to define the web applications completely. Therefore in the process of matching end-user mental model to developers we have to use some strategies and techniques to capture minimum aspects necessary to define a web application.

REFERENCES

Baresi, L., Garzotto, F. & Paolini, P. (2001) Extending UML for Modelling Web Applications. *Annual Hawaii Int.Conf. on System Sciences.* Miami , USA.

Ceri, S., Fratenali, P. & Bongio, A. (2000) Web Modelling Language (WebML): a modelling language for designing Web sites. *WWW Conference.* Amsterdam.

Costabile, M., F., Fogli, D. & Marcante, A. (2006) Supporting Interaction and Co-evolution of Users and Systems. *Advanced Visual Interfaces -AVI.*

Costabile, M., F., Fogli, D., Mussio, P. & Piccinno, A. (2005) A meta-design approach to End-User Development. *VL/HCC05.*

De Oliveira, M. C. F., Turine, M. A. S. & Masiero, P. C. (2001) A Statechart-Based Model for Hypermedia Applications. *ACM Transactions on Information Systems (TOIS),* 19**,** 28-52.

Escalona, M. J. & Koch, N. (2006) Metamodeling Requirements of Web Systems. *International Conference on Web Information System and Technologies (WEBIST 2006).* Setúbal, Portugal.

Fischer, G. & Giaccardi, E. (2004) A framework for the future of end user development. IN Lieberman, H., Paterno, F. & Wulf, V. (Eds.) *End User Development: Empowering People to flexibly Employ Advanced Information and Communication Technology.* Kluwer Academic Publishers.

Fischer, G., Ye, E. G. Y., Sutcliffe, A. G. & Mehandjiev, N. (2004) Meta Design: A Manifesto for End -User Development. *Communications of the ACM,* 47**,** 33-37.

Fratenali, P. & Paolini, P. (1998) A conceptual model and a tool environment for developing more scalable and dynamic Web applications. *EDBT 98.* Valencia, Spain.

Ginige, A. & DE Silva, B. (2007) CBEADS©: A framework to support Meta-Design Paradigm. *12th International Conference on Human-Computer Interaction (HCII 2007).* Bejing, P.R. China, Springer.-accepted.

Jakob, M., Schwarz, H., Kaiser, F. & MITSCHANG, B. (2006) Modeling and Generating Application Logic for Data-Intensive Web Applications. *Internationa Conference on Web Engineering (ICWE06).* ACM Press.

Kobti, Z. & Sundaravadanam, M. (2006) An enhanced conceptual framework to better handle business rules in process oriented applications. *International Conference on Web Engineering (ICWE06).* Palo Alto, California, USA, ACM Press.

Koch, N. & Kraus, A. (2002) Power of UML-based Web Engineering. *Second International Workshop on Web-oriented Software Technology (IWWOST02).* Málaga, Spain.

Koch, N., Zhang, G. & Escalona, M. J. (2006) Model Transformation from Requirements to Web System Design. *International Conference on Web Engineering (ICWE06).* Palo Alto, California, USA, ACM Press.

Liang, X. & Ginige, A. (2006) Smart Business Objects: A new Approach to Model Business Objects for Web Applications. *1st International Conference on Software and Data Technologies.* Setubal, Portugal.

Rode, J. & Rosson, M. B. (2003) Programming at Runtime: Requirements and Paradigms for Nonprogrammer Web Application Development. *Human Centric Computing Languages and Environments-2003.* Auckland, New Zealand, IEEE Symposium.

Schmid, H. A. & Rossi, G. (2004) Modeling and designing processes in e-commenrce applications. *Internet Computing IEEE,* 8**,** 19-27.

Schwabe, D., Rossi, G. & Barbosa, S. D. J. (1996) Systematic hypermedia application design with OOHDM. *seventh ACM conference on Hypertext.* Bethesda, Maryland, United States, ACM Press.