

Integrating Value-based Requirement Engineering Models to WebML using VIP Business Modeling Framework ¹

Farooque Azam, Zhang Li, Rashid Ahmad

School of Computer Science and Engineering
Beijing University of Aeronautics & Astronautics (BUAA), China

{farooque, lily, r.ahmad}@buaa.edu.cn

ABSTRACT

Requirement engineering (RE) is emerging as an increasingly important discipline for supporting Web application development, as these are designed to satisfy diverse stakeholder needs, additional functional, information, multimedia and usability requirements as compared to traditional software applications. Moreover, when considering innovative e-commerce applications, value-based RE is an extremely relevant methodology which exploits the concept of economic value during the RE activity. In contrast, most of the methodologies proposed for the development of Web applications, primarily focus on the system design, and paying less attention to the RE, and specifically to value-based RE. Focusing this aspect, the paper presents integration of value-based RE models to WebML models using our recently proposed *VIP Business Modeling Framework* [1]. We also analyze the framework's potential in linking other modeling approaches, and argue about its significant integration potential with various E-R/OO-based, process aware Web modeling approaches.

Categories and Subject Descriptors

D.2.2 [Software Engineering]: Design tools and techniques – *object-oriented design methods*; H.1.0 [Information Systems]: models and principles – *general*.

General Terms

Design, Languages.

Keywords

Business value model, business information model, business process model, VIP, WebML

1. INTRODUCTION

Web systems are getting complex and typically employ distributed architecture consisting of complex middleware technologies with high-performance back-end software systems that integrate new components with existing legacy applications. The scope and complexity of current Web applications vary widely: from small scale, short-lived services to large-scale enterprise applications, distributed across the Internet, corporate intranets and extranets for achieving some desired business objectives [12]. Today, these applications represent the business itself, rather than only supporting the business. Hence, during RE process, the business and technology issues are tangled in such a way that these cannot be considered in isolation and an integrated

approach to Web system development is required, with which we can assess the impact of a business model on the technical architecture and vice versa.

Existing modeling approaches have been evaluated and discussed in numerous articles such as [16] and [10]. While the literature has considered the extent of support for modeling, the design of Web applications has not adequately considered the extent of modeling support at various levels of abstraction, particularly in terms of the various models that may exist between an original business need and the ultimate detailed design [26]. Figure 1 shows two fundamental models for Web application development: *business models* and *system design models*. The modeling pyramid highlights the fact that moving upward increases the abstraction and moving downward increases the detail in the models.

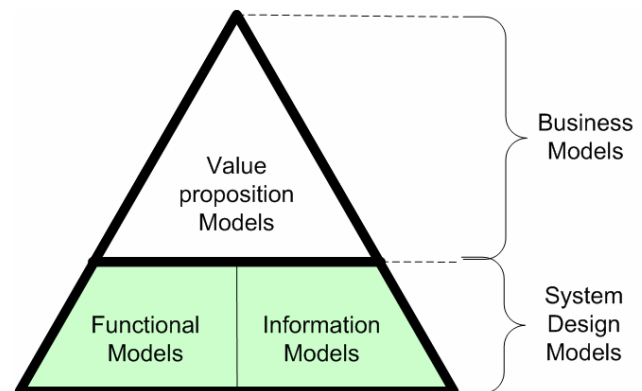


Figure 1. Modeling Pyramid

At the top level we can model the actual business utilizing these systems. As an example, a typical model at this level might represent the value propositions amongst the organizations and other entities that enable the organization to achieve its business goals. While modeling notations at this level are quite diverse, some more formal approaches do exist. A typical example is the *e³-VALUE* business modeling methodology [14], [13] providing *value-based RE through value proposition models*. Value-based requirements engineering exploits the concept of economic value during the requirements engineering activity and it's especially useful when doing requirements engineering for innovative e-commerce information systems. Such systems are generally based on a new, hardly understood, e-commerce idea with which actors potentially can make a profit, or when put into operation, produce something of economic value for the actors. This methodology provides modeling of value propositions, i.e., which parties

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

¹ Supported by Program for New Century Excellent Talents in University

exchange things of economic value with whom, and expect what in return. This methodology focuses on a description of three architectural viewpoints representing the interest of various stakeholders: the *business value* viewpoint, the *business process* viewpoint and the *software architecture* viewpoint. However, its architectural representation is not directed towards its integration with any of the current Web application modeling approaches, which limits its utilization as practical companion with any of the current approaches. Hence, we believe that development of a *generic integration framework* can serve as a bridge between *value proposition business models* and various *system design models* for existing Web application development approaches.

On the other hand, at the bottom of the diagram, we show *system design models*. For Web application design, these models are typically partitioned in two levels: high-level system design and lower-level detailed system design [26]. In MDA terminology we may consider these as PIM and PSM [24]. The lower-level models typically capture design elements that have a direct correspondence to specific implementation artifacts. For example, these models may describe concrete *functional elements*, such as code modules and communication interfaces; and *information elements*, such as page content and navigational links.

Furthermore, the high-level system design models typically capture the domains of functionality and the domains of information that are needed to support the business domains. *Functional aspects* at this level are well supported by UML as some UML models such as activity diagrams can be used to represent the business and operational processes of a system. *Information aspects* at this level could be such as; the relationships between underlying content and the user-perceived views of that content, the interactions with those views, and the ways in which information is represented and presented to the users.

Once we analyze the support of value-based RE in existing modeling approaches, we find that none has formally incorporated such business analysis approach. For example, the comparative study [10] analyzes ten different approaches and primarily discovers and focuses on functional and non-functional requirements. Functional requirements are further sub classified as; data, interface, navigation, personalization and transactional requirements. However, value-based requirement elicitation, specification and validation, which is extremely relevant for

innovative ecommerce Web application is not described as an integral part of any of the methodology. Hence the focus of our research is to develop a *generic framework*, that can integrate *value-based business models* to *system design models* of various Web modeling approaches, by providing appropriate business modeling notations, and further to demonstrate its linkages with some of the influential approaches, as proof of the concept. Specifically in this paper we present its integration with WebML.

The proposed *VIP Business Modeling Framework* [1] comprises three distinct models and in Section 2 we present the rationale for incorporating *VIP models*. In Section 3, using an ecommerce example, we provide example *VIP models* using this framework. Section 4 demonstrates its integration with WebML. Next, in Section 5 we provide comparative analysis of integrating this framework to various other approaches. In subsequent sections (6, 7, 8) we present related work, future work and conclusion respectively.

2. BACKGROUND

For integrating the value-based RE (value proposition) models, we do not directly link these models to *system design models*; rather we develop three distinct business models during business analysis, focusing specific concerns, as shown by the extended modeling pyramid in Figure 2. Moreover, in this figure, the generic framework called *VIP Business Modeling Framework* is also depicted in relation with the extended modeling pyramid, showing specific modeling levels being addressed by various business models in the framework. Nevertheless, it comprises three principal models: (1) BVM–Business Value Model (2) BIM–Business Information Model and (3) BPM–Business Process Model. First, we carry out value-based RE to find out value objects and value exchanges among various business entities and develop BVM, which can specify up to three modeling viewpoints (*value actor*, *detailed actor* and *value activity* viewpoints). Next, BIM is developed by mapping required concepts from the value model. It may specify two modeling viewpoints (*information subsystem* and *information system* viewpoints). Finally, BPM is developed, by mapping required concepts from BVM as well as BIM. It also specifies two modeling viewpoints (*business* and *sequence* viewpoints). These models provide three dimensional information in a ‘methodology neutral’ fashion that further can be used by adding the details

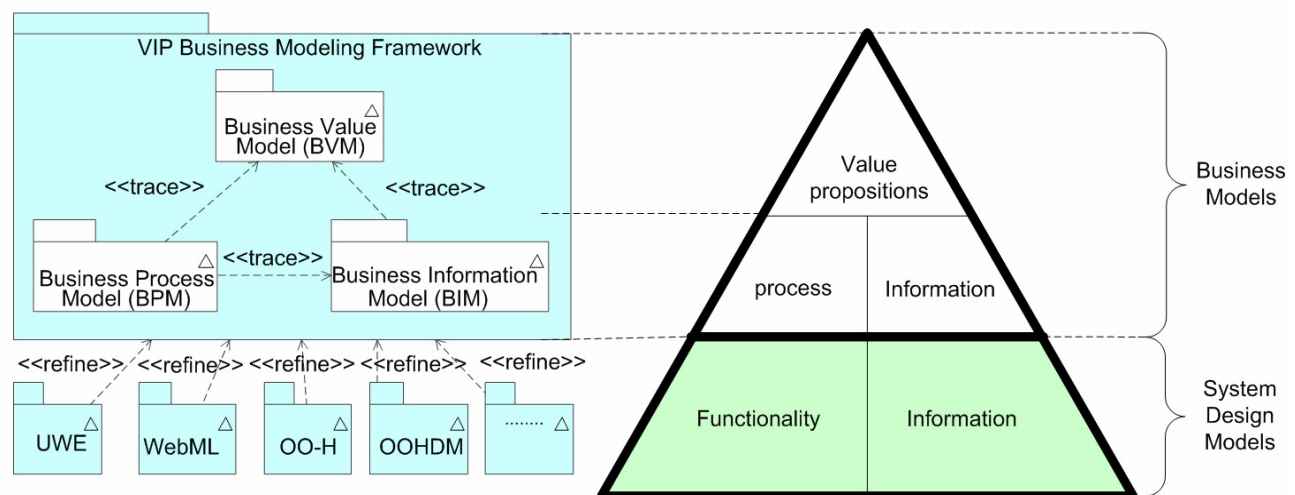


Figure 2. VIP business modeling framework and extended modeling pyramid

during high-level system design in any desired development methodology, such as, UWE [17], OOHDM [25], WebML [7] or OO-H [8] etc.

The rationale for incorporating the BIM is the need for higher level information modeling for current Web application modeling approaches as argued by [26], because these modeling approaches generally focus at a relatively lower-level and don't help the business analyst to communicate and capture the information requirements for analyzing and specifying business ideas. For example, these approaches typically incorporate; modeling of the underlying data (e.g., in WebML this is referred to as the *structural model*); the way in which this data is composed into pages (e.g., in WebML, the *compositional model*); and the topology of the page inter-relationships (e.g., in WebML, the *navigational model*). On top of these, *presentational* and *adaptation* models can be layered. The situation is almost alike in other approaches as well.

Similarly the rationale behind incorporating BPM is the need for merging the information and functional aspects as argued by [22], as well as the need to model *explicit* business process models in order to address the complexity of current Web applications [20]. Next we will present some of the relevant example VIP business models, which are developed using this framework. These models will further be utilized to demonstrate their integration with WebML system design models.

3. EXAMPLE VIP MODELS

The proposed VIP business modeling framework is based on the lightweight UML 2.0 profile (that is, by using stereotypes, tagged values and constraints only). Restricting notations and diagrams to those provided by the UML, offers significant advantage of making use of all benefits and tools that support UML. Metamodel specifications of the three models, UML profiles for supporting these metamodels, and idea of developing these models using UML profiles are presented in [1]. We take help of example models to demonstrate the integration process. For developing the example models, we use a well-known academic example of Ticket Master as elaborated in [1]. TicketMaster Web application supports a kind of e-business which can be categorized as functionally-complex and information-rich application. Here we will keep the example models simple, yet complex enough to demonstrate the concepts appropriately.

BVM: The first model to be developed through the framework is BVM. It stems from e³-VALUE [www.e3value.com] evaluation approach [14], which employs non UML compliant proprietary notations and doesn't support comprehensive integration strategy with existing Web modeling approaches. However, its UML adaptation represented through metamodel concepts and UML 2 profile are developed for BVM. A *global actor viewpoint* for the running example is presented in Figure 3 (*the diagram can easily be studied at 200% zoom*). This diagram depicts the major stakeholders involved in a business and answer the question: "who is offering what to whom and expects what in return" and it is normally used by the analysts during discussions with the business executives. The diagram shows one value actor (Ticket Master) and two market segments (Customers and Event organizers). Each of them owns multiple value interfaces, for example, *Customer Ads* is one of the value interfaces of Ticket Master. Each value interface exchanges value objects in a particular direction such as Event Performance and Participation

etc. Value offering may optionally be represented by a note element (such as bundle) and similarly a value transaction (such as transaction). Most of the UML elements and their stereotypes are evident from the diagram. However, the elided stereotypes are: <<value interface>> for UML port and <<value exchange>> for UML dependency. BVM also comprises two other viewpoints: *detailed actor viewpoint* which incorporates the concept of composite actor; and *value activity viewpoint* which incorporates the concept of value activity, comprising similar semantics, however, with different intent and are not described here considering redundant in presenting the key idea.

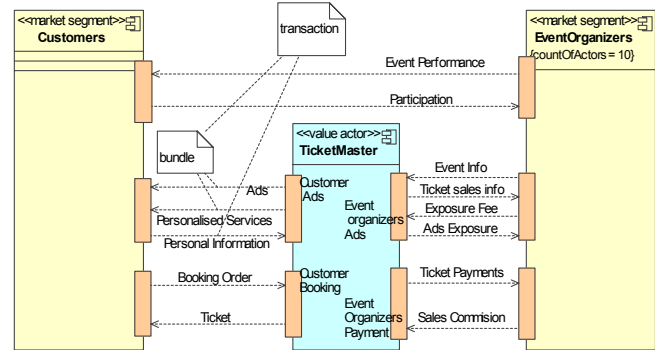


Figure 3. BVM-GAV (Global Actor Viewpoint)

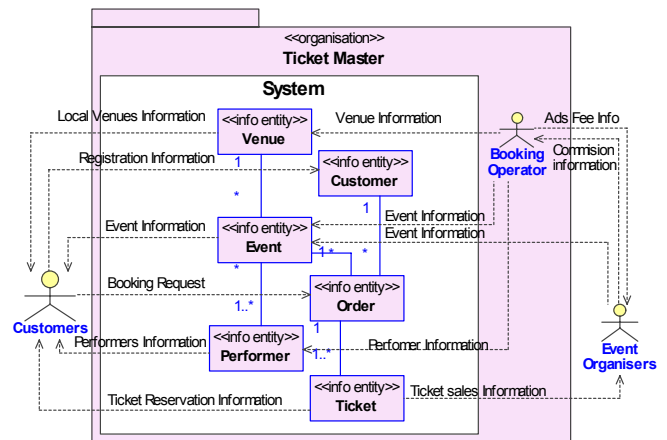


Figure 4. BIM-SV (System Viewpoint): Ticket Master [dependencies are stereotyped by <<flow>>, however, elided for the sake of clarity]

BIM: This model is based on the UML 2.0 semantics of InformationFlows package (Section 17.2 [28]). The development of this model is inspired from WIED [26]. The comparison between WIED and BIM is described in [1]. BIM maps its various concepts from the value model, such as, information flows, information items, external and internal actors etc. BIM comprises two viewpoints: *Subsystem Viewpoint* (SSV) and *System Viewpoint* (SV). Here we present BIM-SV in Figure 4 for the Ticket Master example (*the diagram can easily be studied at 200% zoom*). BIM-SV models the information items (such as booking request, event information etc) that are conveyed between the external actors (such as Customer and Event Organizers) and internal actors (such as Booking Operator) and the System. Information entity (such as Venue, Customer, Event, etc) is one of the key concepts in the BIM, which represents a supplier or consumer of information item being conveyed by the

information flow. An information entity stereotyped as <<info entity>> serves as a persistent information unit providing sink and source of information.

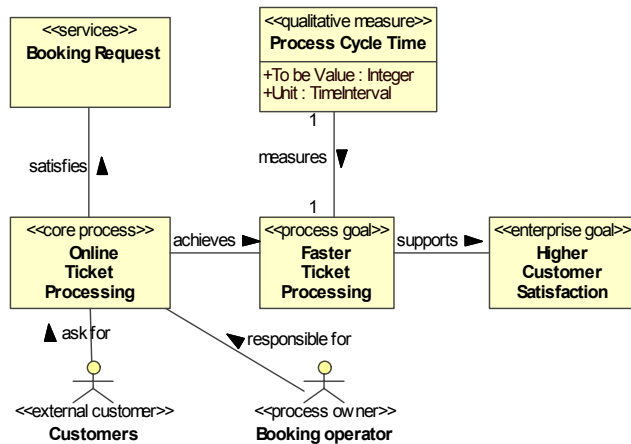


Figure 5. BPM-BV (Business viewpoint): business process—online ticket processing

BPM: For building BPM we make use of UML 2 profile for business process modeling [21]. However, for BPM we have defined addition metamodel extensions to this profile as described in [1]. The purpose of BPM is two fold. Firstly, it maps the important concepts of BVM and BIM to the Business Viewpoint (BV). Secondly, it provides detailed refinement of business processes through Sequence Viewpoint (SV) by tracing the concepts from BIM. An example *business process* (online ticket processing) from the Ticket Master example is presented in Figure 5, which is part of entire business viewpoint. The diagram specifically shows that it is a core business process providing service for (*Booking Request* information item – from Figure 4); hence achieves a measurable process goal (*Faster Ticket Processing*), which in turn supports an enterprise goal (*Higher Customer Satisfaction*), and while customers ask for the core process, the booking operator is responsible for it.

Next, sequence viewpoint is developed comprising multiple detailed process diagrams. In our proposed BPM, this diagram elaborates each business process by mapping the concepts from BIM. For example, Figure 6 further elaborates Online Ticket Processing – a core business process (Figure 5). This diagram presents a workflow by mapping the concepts from BIM-SV (Figure 4). Here note that relevant information items from BIM-SV are mapped to *ObjectNodes* stereotyped by <<information>>. Moreover, information entities in BIM-SV representing information sources and sinks are mapped to *DataStores* which semantically acts as data store node for non-transient information (Section 12.3.21 [28]). At this level, process modeling is quite abstract and should be further refined during system design phases, leading towards functional design of a Web application.

The primary focus of this paper is to show that once these *VIP models* are ready, next how we can integrate these models to various WebML models covering system design level of the modeling pyramid as depicted in Figure 2. Hence we have kept our discussion rather brief on the process of developing VIP models themselves. However, in the subsequent section we will discuss the integration of these models with WebML models in greater detail.

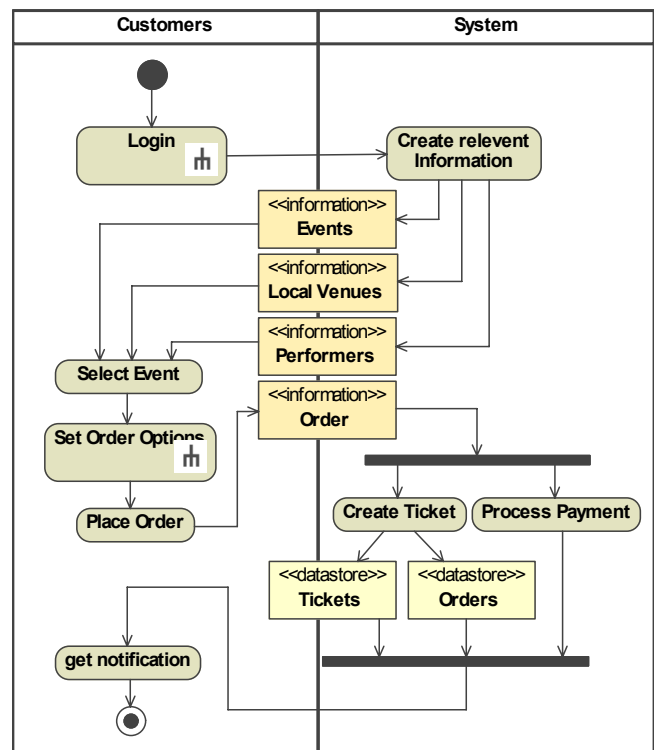


Figure 6. BPM-SV (Sequence viewpoint): detailed process diagram—online ticket processing

4. INTEGRATING VIP MODELS TO WEBML MODELS

WebML enables designers to express the core features for a Web application, without exposing detailed system descriptions. WebML concepts are associated with a graphic representation, which is supported by a CASE tool (Web Ratio). WebML supports XML syntax for storing model specifications. The earlier work on specification of a Website in WebML provides four fundamental perspectives: structural model, hypertext (composition, navigation) model, presentation model and personalization model [7]. However, acknowledging the need for explicit process modeling the later work [4] and [5] provides the extension for explicit process models in order to cater for the complexity of today's Web applications. However, integration for value-based RE models for supporting design of complex e-commerce Web applications is still a significant missing link that we have attempted to form.

4.1 Integration Strategy

The extension of WebML towards workflow driven hypertext [4], considers *process*, *data* and *hypertext* as key ingredients for model-driven development of process-aware Web applications, and integrates *process* model as suitable extensions of *structural* model and *hypertext* models. In WebML, integrating hypertexts with processes means; delivering Web interfaces that permit the execution of activities and embody constraints that drive the navigation of users. Our strategy for the integration of VIP business modeling framework focuses on integration of *VIP-BPM* and *VIP-BIM* to WebML *process* and *structural* models.

The recent work towards extending WebML [5], supports *process modeling* by adopting the Business Process Modeling Notations

(BPMN) [29]. Conversely, proposed VIP business modeling framework supports UML 2 activity diagrams in BPM-SV, for elaborating the business process models. Hence one of the integration steps involves mapping of BPM-SV to WebML process model represented by BPMN. Secondly BIM-SV is mapped to WebML structural model. Lastly, the hypertext model is developed which depends on both: WebML *process* models and WebML *structural* model. Hence in the subsequent sections, following the same sequence of steps, we present the integration process: first by deriving the *process* model, and then the *structural* model for WebML. The *hypertext* model is directly derived from the two earlier models and indirectly from the VIP models. Hence, in order to complete the discussion and for visualizing the overall impact of mapping, finally we do briefly present derivation of hypertext model.

4.2 Deriving WebML Process Model

For the running example of Ticket Master, BPM-SV elaborates an example business process—online ticket processing (Figure 6). For deriving the WebML process model we need a mapping between the elements of UML 2 activity diagram and BPMN. The resulting WebML process model is depicted by Figure 7. The rules for mapping are described below:

1. *Creating a Pool for each User Group:* WebML supports a separate siteview for each *user group* and communication between the various users can essentially be considered as asynchronous collaboration between the various *siteviews* through underlying data repository. From process perspective each *user group* can be considered as a separate *process actor*, and can be represented by a *pool* in the BPMN process model for WebML. BPMN semantics allows collaboration between various pools by passing messages through *message flow*, while after passing the message, the process within pool doesn't lose the control and continues to execute subsequent activities within the pool specified through the *sequence flows*. The alternative way to model various user groups as demonstrated by [5] is to specify one lane for each user group (process actor)

within a pool. However, using one pool for modeling various users groups doesn't properly convey an asynchronous and collaborative nature of the process, which is more prominent in case of WebML siteviews. Hence we have created pools from the BPM-SV (Figure 6) using following procedure:

- a. Identify various user groups (process actors) involved in the hypertext activities in order to achieve the business process, and associate a *pool* to each *process actor*. For example we identify two process actors (manager and customer) and we define a pool for each in Figure 7. Each pool is a candidate for a separate siteview in the hypertext model.
 - b. Identify the non-hypertext activities to be implemented as internal business logic and consider these to be performed by the system under consideration. If there is any such activity, then define a separate pool representing as System. Defining a pool for the system allows segregating hypertext and non-hypertext activities. Non-hypertext activities are needed to be implemented as internal business logic components for supporting hypertext activities.
2. *Specifying Activities for each pool:* Various actions from BPM-SV (Figure 6) are mapped to respective activities in WebML process model. Two types of actions: UML call behavior action and UML call operation actions are specified in BPM-SV. In a strict mapping scenario a *UML call behavior action* is logically mapped to *BPMN sub-process* which is semantically a compound activity that is defined as a flow of other activities. For example, Set Order Options call behavior action is mapped to the sub-process of the same name in Figure 7. Moreover, *UML call operation actions* shall be mapped to *BPMN Tasks*. For example in Figure 7; tasks like *select event*, *create relevant info*, *place order*, *process payment*, and *create ticket* etc., are mapped from the call operation actions of the same name. However, keeping in view the modeling primitives available in the target modeling language, such a strict mapping is not always required. For example, *Login* call behavior action (Figure 6) is not mapped to equivalent Login sub-process rather

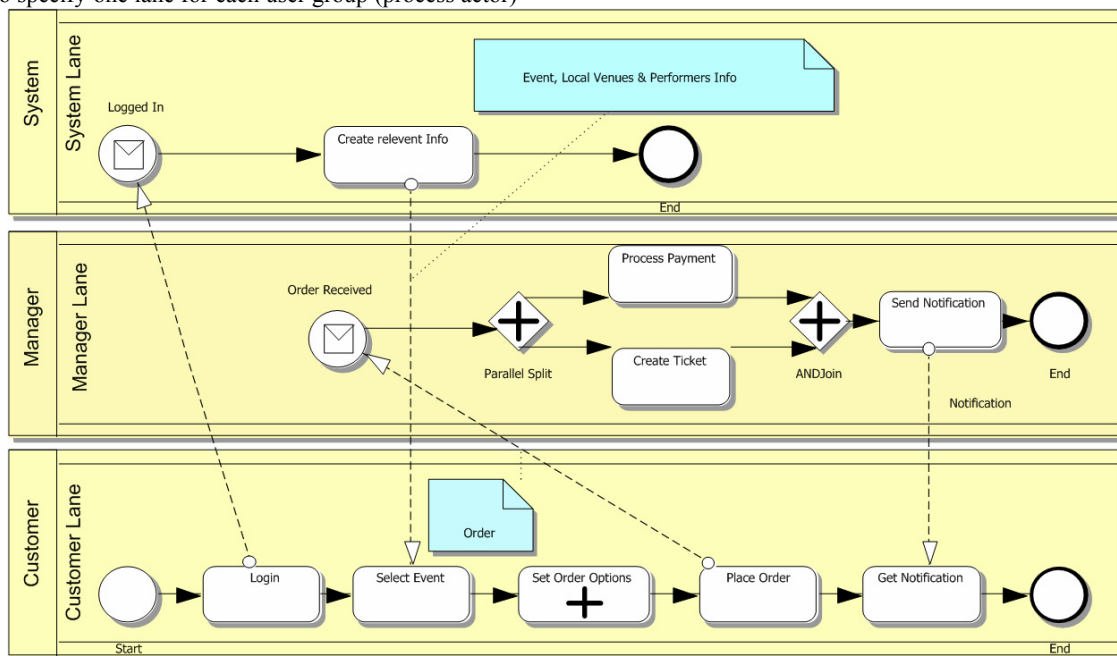


Figure 7. WebML Process Model

it is mapped to Login task, because WebML already contains a built-in hypertext primitive called Login operation unit, which already comprises the necessary behavioral semantics.

3. *Specifying Sequence flows and Message flows*: BPM-SV comprises two kinds of flows: *control flows* and *object flows*. These flows are mapped to BPMN flows as shown in Table 1. For example, in Figure 6, control flow is mapped from *login* action to *create relevant information* action. However, in Figure 7 *create relevant information* task is represented as internal business logic to be implemented by the System; and it is initiated through *message flow*, activating the *start message event* in System pool. Similarly data flows (events, local venues and performer) from *create relevant information* to *select event* in Figure 6, are mapped as message flow with attached data object in Figure 7. Here *select event* task is modeled as of *Receive Type*, which bears semantics that it waits for the incoming message before executing. Similarly *order* data flow in Figure 6, is mapped as message with attached *order* data object, from *place order* task to *order received* start message event in Figure 7.

Table 1. Mapping between UML 2 and BPMN flows

	UML 2 activity diagram flows	WebML process flows
1.	Control flow within a swimlane	Sequence flow within a pool
2.	Control flow across the swimlanes	Message flow across pools
3.	Data flow within a swimlane	Sequence flow with attached data object
4.	Data flow across the swimlanes	Message flow with attached data objects

Table 2. Mappings between UML nodes and BPMN gateways

	UML 2 activity diagram nodes	BPMN gateways
1.	Decision node	Exclusive OR split gateway
2.	Merge node	Exclusive OR join gateway
3.	Fork node	Parallel Split gateway (AND split)
4.	Join node	Parallel Join gateway (AND Join) gateway

4. *Specifying Gateways*: In BPMN, gateways are modeling elements that are used to control how sequence flows interact as they converge and diverge within a Process. Conversely, UML 2 Activity diagram comprises decision/merge and fork/join nodes for this purpose. BPMN gateways comprise much bigger set of semantics; hence UML nodes do not have straightforward mapping to all BPMN gateways. For example, *inclusive* gateways and *complex* gateways in BPMN have complex semantics as compared to UML decision or fork nodes, and require specific patterns of UML elements to implement these gateways. However, vice versa can be achieved directly. A good comparison between the two type of notations can be seen in [30], where the comparison is based on 21 well-know workflow patterns, and its guideline can be used for various mappings based on specific patterns. Table 2

shows the mapping from UML activity nodes to BPMN gateways for simple cases. The implementation of these mappings is evident in Figure 7, where *Parallel split node* (AND split) are mapped from *Fork node* and *AND join* is mapped from *Join node* specified in BPM-SV.

Discussion on Derived WebML Process Model (Figure 7): The figure shows three pools and hence process actors: Customer, Manager and System. The Customer and Manager pools depict two specific groups of users those take part in navigational activities and would represent respective siteviews in hypertext space. The System pool represents non-navigational activities providing underlying business logic to support user actions performed in hypertext space. The process starts once customer logs in for initiating ticket processing activity. The successful user login sends *logged in* message to the *system pool* and this message triggers start message event, which in turn generates a token to start *create relevant info* task for the system. After this the system's part of the process is finished as depicted by an end event. The *select event* task in Customer pool waits for the message from *create relevant info* task. This message also has attached data object (Event, Local venues and Performer). As the message is received the Customer can now perform Select Event task and then moves to *set order options* sub-process, and then to *place order* task. As the customer places the order, a message is send to the *manager* to start two tasks which can be completed in any order: *process payment* and *create ticket*. As the two tasks are successfully completed, the *manager sends the notification* to the customer regarding details of the order, and finishes her part of the process. The customer *gets the notification*, saves or prints its details and also finishes his part of the process.

4.3 Deriving WebML Structural Model

Recent work on WebML extensions [5] for supporting process-aware Web applications proposes a *process reference model* and various *process management units* for specifying and deploying process-driven hypertexts as sets of interconnected Web pages and operations. Consequently, the basic structural model is extended as a composition of three sub-models: (1) process reference model (PRM), (2) user reference model (URM), and (3) application data model (ADM). The extended derived *structural model* for the running example of Ticket Master is as depicted in Figure 8. Amongst these three models, ADM is the most interesting for our integration purpose, because from BIM-SV elements we map ADM elements.

In WebML, ADM is developed using conceptual database design guidelines. However, to cope with process representation, the designer specifies an arbitrary number of relationships amongst various ADM entities and the *ActivityInstance* entity in PRM. The relationships are specified for those ADM entities which are required to be connected to the process activities for manipulating the associated data. For deriving the *structural model*, from of the *VIP business models*, we specifically target ADM, and use following rules:

1. *Deriving Entities*: For each information entity in BIM-SV (System Viewpoint) we create an entity of ADM. For example, in the running example of Ticket Master, entities like *venue*, *performer*, *event*, *ticket* and *order* in BIM-SV (Figure 4) are mapped to equivalent entities of the same name in ADM (Figure 8). However, some leverage in the choice of entities may be made by the designer, for example, *Customer* entity in BIM-SV, is a user specific entity, and WebML already

provides a *user reference model* (URM) comprising two entities (User and Group). Hence while deriving the ADM, the designer may either choose to define a *Customer* entity in ADM and provide 1:1 relationship with *User* entity in URM, or she may directly use the *User* entity and suitably enhance its features to suite the application data requirements. Here we have adopted the later approach and therefore mapped the Customer entity in BIM-SV to the User entity in URM.

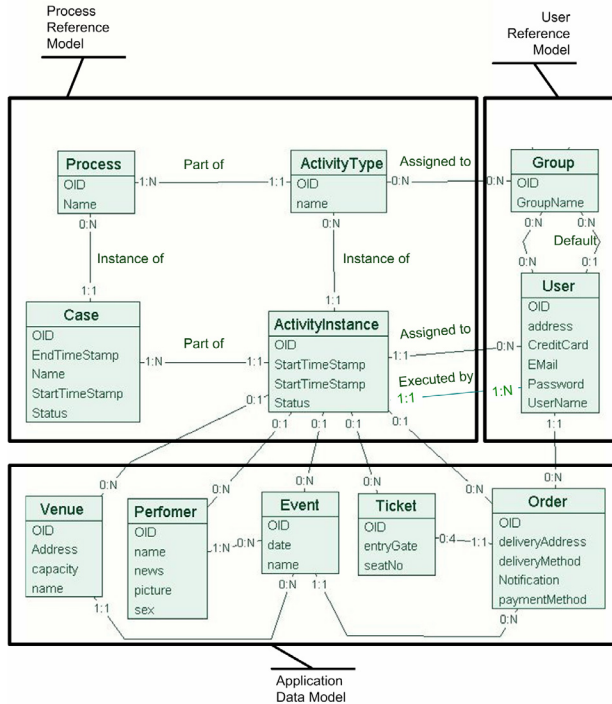


Figure 8. Extended WebML Structural Model –Ticket Master Application

2. *Deriving relationships*: All the associations in BIM-SV are mapped to relationships among the equivalent entities in ADM. However, additional relationships between entities and *ActivityInstance* entity in process reference model (PRF) are also defined. These relationships are defined for those *entities* which are required to fulfill the data requirements on instantiation of various *process activities*.

4.4 Developing Other WebML Models

The *hypertext model* is the most affected amongst the other indirectly affected models, as it has been significantly extended to support process flows and named as *process-aware hypertext model*. Hence, it is pertinent to present the *process-aware hypertext model* for completing the discussion, in order to show the impact of the proposed integration with VIP business models. Conversely, the *presentation models* and *personalization models* are not discussed here and these may be developed as described in WebML literature, such as [7].

The *process-aware hypertext design method* comprises two main phases: *high-level hypertext design*, where the overall structure of the hypertext is sketched and *detailed hypertext design*, where the operational details of the hypertext are fully specified. Four groups of *process management units* are available in hypertext model, namely (1) assign units (2) activity units (3) process aware content units, and (4) decision units. Their details can be seen in [4] and [5].

4.4.1 High-level Hypertext Design

High-level hypertext design identifies the main siteviews and pages of the application front end. In this phase, the contents of pages in terms of units and links can be sketched at a variable degree of precision, to highlight only the most important aspect of the interface and delimiting the areas of the hypertext that support the execution of activities. Typically, high-level hypertext design produces a hypertext "skeleton" for each siteview, highlighting the entry and exit points of each activity, and omitting the details of the pages and units necessary to build the interface for activity execution. For the running example of Ticket Master, we have already identified two process actors which are involved in hypertext activities: *Customer* and *Manager*. Hence we provide a distinct siteview for modeling hypertext activities of each process actor. Figure 9 depicts the high-level hypertext design of *Customer Siteview*. It describes that a customer reaches to the *Home* page of this siteview as soon as she successfully logs in to the system. The customer starts the ticket processing activity, through *Place Ticket Order* navigation link. From this link customer reaches *Event Information Page*.

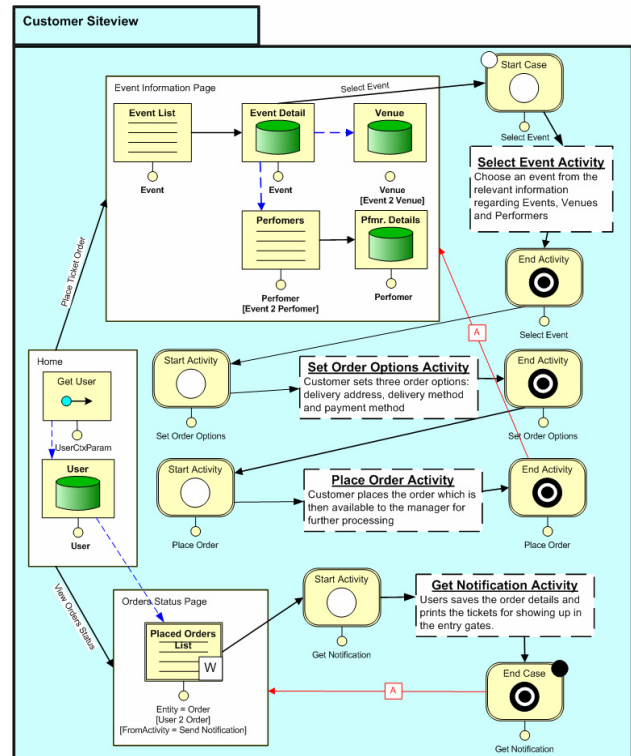


Figure 9. Customer Siteview

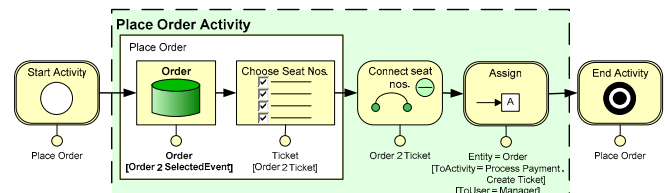


Figure 10. White-box view of 'place order' activity

Then from here through *Select Event* navigation link '*Select Event Activity*' starts, which is delimited by *Start Case* unit and *End Activity* unit. The *Select Event Activity* has been depicted by a dashed rectangle and its details are specified informally at this level, and will be formally modeled in detailed hypertext design.

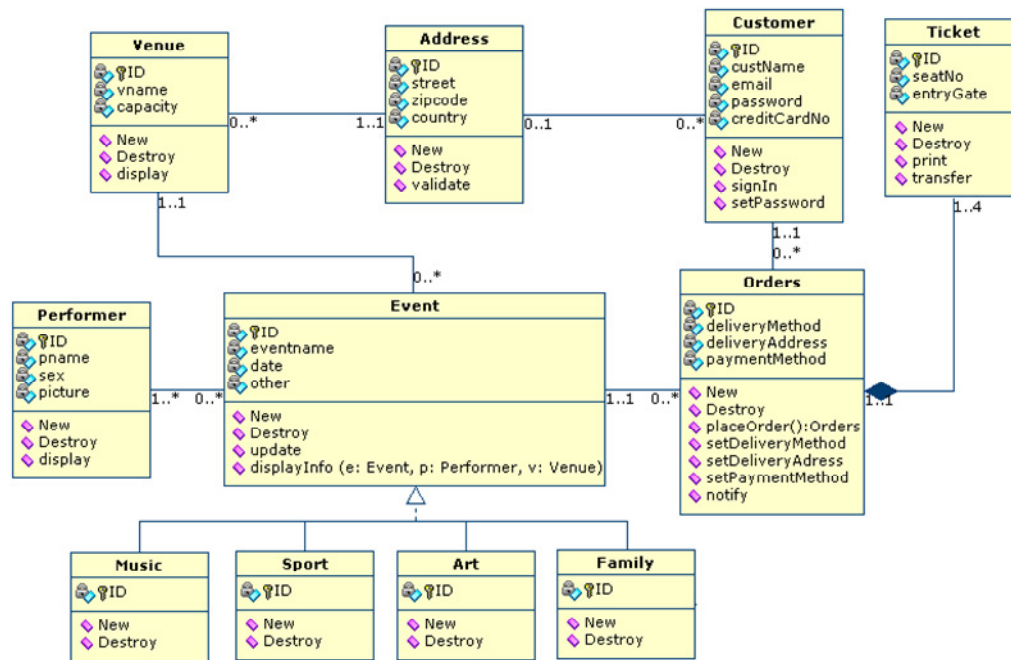


Figure 11. Conceptual Model for OO-H

Next, the customer performs various subsequent activities in the order specified in the *Customer* pool of the process model (Figure 7). The subsequent activities are: *Set Order Options* and *Place Order* activities. Finally the customer reaches back to *Event Information Page* through an automatic link labeled as 'A'.

Once the customer desires to know the status of her already placed orders, then she navigates through the *View Orders Status* link from the *Home* page. Through this link, the customer reaches to *Orders Status Page* which contains a process-aware index unit. This index unit receives the *Order* data from *Send Notification* activity which is part of *Manager Siteview* (not shown) and lists the status of his already placed orders. From this index unit the Customer can start *Get Notification* activity for any of the selected order for saving its details and printing the respective tickets for showing up in the event entry gates.

4.4.2 Detailed Hypertext Design

In the detailed hypertext design, white-box view of the informally specified activities is provided. The white-box view contains various hypertext and process-aware primitives for achieving the desired functionality. For example Figure 10 shows white-box view of the *place order* activity. Similarly, the white-box views of various activities are also developed separately.

5. INTEGRATING OTHER APPROACHES

As already highlighted, the purpose of *VIP Business Modeling Framework* is to integrate value proposition models to various influential Web modeling approaches and for this purpose we have developed two high-level business models (BIM and BPM) providing smooth translation from value-based primitives to system design primitives of such approaches. In this regard we have also attempted the *framework* integration with UWE, OO-H and OOHDH. In this section we discuss the outcome of our experience of integrating VIP modeling Framework to such influential approaches.

Essentially, various Web modeling approaches have been proposed over a period of time with diverse methodologies.

However, within this diversity there are commonalities as well, especially amongst data-driven approaches. For example we see various E-R based approaches like WebML which base its underlying information on E-R models. Similarly there are various OO approaches which primarily base their methodology on static class structure models, such as OO-H, UWE and OOHDH, just to name few. Although these approaches follow different methodologies, their common style of information structuring and process modeling provides us an elegant linkage proposition. For example, in case of BIM-SV its further refinement allows us to translate it to the conceptual models in OO-H and UWE; and these conceptual models provide the baseline for developing all other models in both of these approaches.

Figure 11 depicts an OO-H conceptual model for the running example of Ticket Master. Deriving conceptual model from BIM-SV (Figure 4) requires a simple approach as *information entities* may directly be mapped to *conceptual classes*. Then the designer works out necessary details for each '*information object*' being conveyed to/by each '*information entity*'; and then decides how to enrich the class features (attributes and operations, at time associations as well). Some additional classes may be defined in course of refinement. For example, specializations of Event class (Music, Sport, Art, and Family). Moreover, some optimizations can also be made like an Address class is added which is commonly required by Customer and Venue classes.

On the other hand, the growing complexity of modern Web applications calls for exclusive process modeling, which allows separating the process concerns from the information. Due to emergence of UML as an industry standard, various UML diagrams are commonly used for this purpose. For example, OOHDH, UWE, and OO-H are amongst those which use activity diagrams. However, WebML incorporates BPMN for this purpose. Essentially, in such approaches, the different techniques of using process models do not affect the integration of BPM-SV with these approaches. For example, there is difference in using

process models with the navigational models in UWE and OO-H [20]; as UWE enriches the navigation model by defining *process integration points* through which user may leave and return to navigation mode. Whereas, in OO-H (also in OOHD and WebML), the navigation models are driven by the process models. However, this difference in approach doesn't restrict us to adopt a common straightforward translation from BPM-SV to their process models. A detailed account of integration with UWE is presented in [2].

Besides these, there are various other approaches which incorporate a different technique for capturing business. For example, few approaches make use of *functional refinement trees* [19] or *task taxonomies* [27] during requirements analysis, for the hierarchical decomposition of business functions. Amongst these, the extended work on OOWS [27], proposes two steps in task modeling: (1) *task identification* in which identified tasks are organized in a task taxonomy, and (2) *task description* in which leaf tasks are further elaborated using UML activity diagrams. In such a case, BPM-SV may support either of the two steps. For example, Figure 12 depicts task taxonomy of the core business process 'online line ticket processing' (Figure 6). Here *ticket processing* business process is depicted as top-level business function and all enclosed business activities are depicted as child tasks. Any further refinements may be made during task decomposition, for example, *login* task (call behavior action in BPM-SV) is further decomposed to *register* and *sign in* tasks, and similarly *set order options* is also decomposed. Conversely, *task description* step may also be supported by BPM-SV, as *call operation actions* in BPM-SV may directly be elaborated through *task descriptions* using activity diagrams.

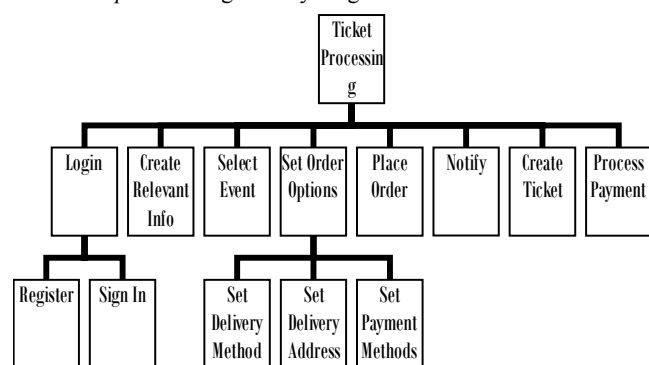


Figure 12. Task Taxonomy for business process 'online ticket processing'

During conceptual modeling, the significance of utilizing BIM-SV amongst OOWS [11] based approaches is also undeniable, as it relies on UML class diagram for defining system structure and relationships between classes.

Moreover, the approach specified in [27], provides *User Diagram* during navigational modeling. This diagram provides mechanisms to cope with additional user management capabilities, such as, *user specialization* that allows defining user taxonomies to improve navigational specification reuse. Here BIM-SV again does provide the input as it already identifies the users.

Hence, from the experience gained so far, we may argue that *VIP Business Modeling Framework* may integrate *value proposition models* to *E-R/OO-based, process-aware* Web modeling approaches. As such approaches are suitable for developing information-rich and functionally-complex *ecommerce*

applications, the modeling capability of such approaches can be enhanced through the proposed framework, because value proposition models can now be used with such approaches to explore, specify, and validate innovative *ecommerce* ideas.

6. RELATED WORK

In this work we only focused on functional requirements. However, the need to capture non functional requirements (NFRs) cannot be overlooked. For handling NFRs, various goal-based approaches are well-known, which do capture high-level strategical goals during early-phase RE. Such approaches generally capture NFRs (quality goals) under the notion of *soft goals*. One of such approach is known as *i** (distributed intentionality) methodology [9]. Incidentally, we may use the recent work [15] which provides the guidelines to integrate various primitives of *i** models to *e³-VALUE* models. This guideline can be used to handle NFRs in the context of the proposed framework.

7. FUTURE WORK

Another influential work is Hera [18], which applies Web engineering principles in the area of semantic Web technologies. Hera's model-driven approach most notably uses RDF(S) [6] to specify the design of Web information systems. We intend to work in future to explore how it can be supported by the proposed framework; specifically we intent to explore its existing support for process modeling as well as compatibility of conceptual model primitives with BIM-SV.

Besides *i** models, AWARE [3] is also an interesting approach. It extends traditional goal and scenario-based approaches for refining high-level goals into website requirements. Since it provides hypermedia requirement taxonomy to facilitate Web conceptual design, the linkages of the framework to AWARE are interesting as a future work; with a view to analyze it's the utility in extending the taxonomy to capture value-based requirements.

8. CONCLUSION

The contribution made by this paper is to show how we can integrate *VIP Business Modeling Framework* to WebML in Section 4. Moreover, based on the experience of this work, we provide comparative analysis of integrating this framework to various other approaches in Section 5. Furthermore, we draw attention in Section 6 to show how the proposed framework may handle NFRs. At present we are working on a large case study for improving the framework definitions and for carrying out empirical validations of the approach, and results will be published elsewhere.

9. REFERENCES

- [1] Azam, F., Li, Z., and Ahmad, R. Introducing VIP Business Modeling Framework for Innovative, Information-rich and Process-aware E-commerce Web Applications, in IEEE International Conference on Emerging Technologies (ICET'2006), Nov, 2006, Peshawar, Pakistan.
- [2] Azam, F., Li, Z., and Ahmad, R. Integrating VIP Business Modeling Framework to UML-based Web Engineering Approach. In IEEE International Conference on Emerging Technologies (ICET'2006), Nov, 2006, Peshawar, Pakistan.
- [3] Bolchini, D. Mastering the Requirements Analysis for Communication-intensive Websites, PhD Thesis, Faculty of Communication Sciences, University of Lugano, Nov 2003

- [4] Brambilla, M., Ceri, S. Comai, S., Fraternali, P. Specification and Design of Workflow-Driven Hypertext. *Journal of Web Engineering*, Vol. 1, No. 2, 163-182 (2002):
- [5] Brambilla, M., Ceri, S. Comai, S., Fraternali, P., Manolescu, I. *Process Modeling in Web Applications*. ACM Transactions on Software Engineering and Methodology (TOSEM), 2006
- [6] Brickley, D., Guha, R.V. *RDF Vocabulary Description Language 1.0: (2002) RDF Schema*. W3C Working Draft
- [7] Ceri, S. Fraternali, P., Bongio, A., Brambilla M., Comai S., Matera M.: *Designing Data-Intensive Web Applications*. Morgan Kaufman (2003).
- [8] Cristina, C., Jaime, G.. *Advanced Conceptual Modeling of Web Applications: Embedding Operation Interfaces in Navigation Design*. 21st International Conference on Conceptual Modeling. El Escorial, Madrid (2002).
- [9] Yu, E. *Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering*. in Proc. 3rd IEEE Int'l Symp. Requirements Eng. (RE 97), IEEE CS Press, 1997, pp. 226-235.
- [10] Escalona, M. J., & Koch, N. *Requirements Engineering for Web Applications: A Comparative Study*. *Journal of Web Engineering*, Rinton Press, Vol. 2, No. 3, February 2004, 192
- [11] Fons, J., Pelechano, V., Albert M., and Pastor, O. *Development of Web Applications from Web Enhanced Conceptual Schemas*, pp. 232 - 245, Conference on Conceptual Modeling (ER), Is International, 22nd, 2003 - October, Chicago, Illinois (EE.UU.), Il-Yeol Song, Stephen W. Liddle, Tok Wan Ling, Peter Scheuermann, Springer-Verlag, Lecture Notes in Computer Science, 3-540-20299-4, 2813, 2003
- [12] Ginige, A., and Murugesan, S. *Web Engineering: An Introduction*, IEEE Multimedia, Special issues on Web Engineering, vol 8, no 1, pp 14-18 (2001).
- [13] Gordijn, J. *E3value in a Nutshell*. Technical report, HEC University Lausanne, 2002. URL: <http://e3value.few.vu.nl/docs/bibtex/pdf/e3valueNutshell.pdf>
- [14] Gordijn, J. *Value-based Requirements Engineering: Exploring Innovative e-Commerce Ideas*, PhD Thesis, Vrije University Amsterdam, (2002).
- [15] Gordijn, J., Yu, E., and Raadt, B., and der. V. *E-Service Design Using i* and e3value Modeling*. In *IEEE Software*, Vol. 23(3):26-33, May 2006
- [16] Gu, A., Henderson-Sellers, B., and Lowe, D. *Web Modeling Languages: The Gap between Requirements and Current Exemplars*. In *Proceedings of the 8th Australian World Wide Web Conference*, (2002), Gold Coast, pp. 362-375. <http://ausweb.scu.edu.au/aw02/papers/refereed/lowe/index.html>
- [17] Hennicker, R., Koch, N. *A UML-based Methodology for Hypermedia Design*. *Lecture Notes in Computer Science*. Proc. UML'2000. York, England.
- [18] Houben, G. J., Barna, P., Frasinca, F., and Vdovjak, R. *HERA: Development of semantic Web information systems*. In *Third International Conference on Web Engineering ICWE 2003*, vol. 2722 of LNCS, Springer-Verlag, pp. 529-538
- [19] Insfr'an, E., Pastor, O., Wieringa, R. *Requirements Engineering-Based Conceptual Modelling*. *Requirements Engineering* 7 (2002) 61-72
- [20] Koch, N., Kraus, A., Cachero, C., and Meliá, S. *Integration of Business Processes in Web Applications Models*. *Journal of Web Engineering*, Rinton Press, Vol. 3, No. 1, May 2004, 22-49
- [21] List, B., & Korherr, B. *A UML 2 Profile for Business Process Modeling*. In 24th International Conference on Conceptual Modeling, Klagenfurt, Austria, October 24-28, 2005, Proceedings. *Lecture Notes in Computer Science* 3716 Springer 2005, ISBN 3-540-29389-2 (pp 85-96)
- [22] Lowe, D., and Henderson-Sellers, B. *Characterising Web Systems: Merging Information and Functional Architectures*. In V. K. Murthy & N. Shi (Eds.). *Architectural Issues of Web-Enabled Electronic Business*. Hershey, PA, USA: Idea Group Publishing (2003).
- [23] MOF QVT Final Adopted Specification ptc/05-11-01.
- [24] OMG. *Model Driven Architecture*, A Technical Perspective, Document ab/21001-02-05, February 2001
- [25] Schwabe, D., and Rossi, G. *Developing Hypermedia Applications using OOHDM*. In *Proceedings of Workshop on Hypermedia Development Processes, Methods and Models (Hypertext'98)*, Pittsburgh
- [26] Tongrunglejana, R. and Lowe, D. (2004). *WIED: A Web Modeling Language for Modeling Architectural-Level Information Flows*. *Journal of Digital Information*, Volume 5 Issue 2, Article No. 283, 2004-08-10. <http://jodi.ecs.soton.ac.uk/Articles/v05/i02/Tongrunglejana/>
- [27] Torres V., Pelechano, V., Ruiz, M., and Valderas P. *A Model Driven Approach for the Integration of External Functionality in Web Applications*. The Travel Agency System .pp. 1 - 11, Workshop on Model-driven Web Engineering (MDWE), Is International, I, 2005 - July, MDWE, Sydney (Australia), 2005
- [28] UML 2.0 Superstructure Final Revised Adopted specification <http://www.omg.org/cgi-bin/doc?ptc/2004-10-02>
- [29] WHITE, S. *Business processing modeling notation (BPMN)*, version 1.0 (2004). <http://www.bpmn.org>.
- [30] WHITE, S. *Process Modeling Notations and Workflow Patterns* (2004b). URL: http://www.omg.org/bp-corner/bp-files/Process_Modeling_Notations.pdf