Web Information Exchange Diagrams for UML

David Lowe and Rachatrin Tongrungrojana

University of Technology, Sydney
PO Box 123 Broadway NSW 2007, Australia
{david.lowe, rachatrin.tongrungrojana}@uts.edu.au

Abstract. A crucial aspect in the development of Web systems is the ability to ensure that the relationships between the system design and the business models, processes and workflows are understood. By representing these relationships and defining transformations between them we support the joint evolution business and web systems and ensure their compatibility and optimisation. In previous work we have developed and evaluated a model (called WIED) which creates this bridge. The existing model is generic, but the notations and transformations have been based on mappings between specific models - namely the e^3-value and WebML models. In this paper we illustrate how the WIED model can also be represented using a UML-compliant notation.

1 Introduction

In the span of a decade, the Web has transformed entire industries and entered the mass culture. It has created new expectations on ease of access, freshness, and relevance of information accessed via the Internet. Online Web-enabled systems have become increasingly crucial to both business success and to the provision of social and government services [1]. These systems are much more complex than simple web sites containing static pages. They typically employ Web technologies to provide a complex distributed front-end combined with high-performance back-end software systems that integrate new components with existing legacy applications to support critical business processes [1, 2].

One crucial aspect in the development of these complex systems is the ability to understand the relationships between system designs and business models, processes and workflows. (Good discussions in regard to this issue can be found in [3–5]). Further, by representing these relationships and defining transformations between them we potentially support the joint evolution of business and web systems and ensure their compatibility and optimization [6].

In response we have proposed an information model – the Web Information Exchange Diagram (WIED). This concept has been evaluated by a series of empirical studies and the results have provided evidence that WIED is a useful tool in supporting an understanding of ways in which business models affect the information design, but also ways in which the systems designs and changes to these designs affect the business model. This work has been published in [7–9].

The design of WIED is based around an abstract model which has been formalized as an XML DTD (Document Type Definition). In our previous work we showed how this model could be operationalized as a notation and associated diagram that are consistent with WebML [10]. We also developed a set of transformations between e^3-value (a business modelling notation [11]) and WIED, and then between WIED and WebML [12]. These related models were chosen as typical exemplars of business models and detailed design models rather than because of any particular strength these models may have. This has allowed us to evaluate the approach and ensure its effectiveness in linking typical modelling notations. We do however recognise that other modelling languages are also widely used - particularly UML.

In this paper we look at how the formal WIED model can be mapped into the UML notation, and show that the result is a new UML diagram that can capture the relationship between high-level business models and processes, and lower level information designs. We begin by providing, in the next section, a background of the development of WIED, followed by a brief overview of WIED. We then go on to look at the details of how WIED maps into UML. Following that we discuss how the WIED (UML-compliant) can be linked to other UML-based modelling approaches. We finish the paper with conclusions and present some ideas for further work.

2 Background

2.1 Web System Modeling

As discussed above, over the last decade we have seen the rapid emergence of systems that utilize web technologies to support the integration of complex functionality with rich information handling. This emergence has been accompanied by the development of modelling languages capable of capturing some – though not all – of the aspects of these systems. To model these systems there are a number of elements that we would like to represent. At the highest level of abstraction we have a business model showing the essential aspects (such as the strategic intent) of the way of doing business. Often this will be represented in terms of the value exchange between the organization and other entities that enable the organization to achieve its business goals. A typical example of a relevant modeling notation is the e^3-value notation [11, 13]. This model focuses on the core concept of value, and expresses how business value is created, interpreted and exchanged within a multi-party stakeholder network.

Conversely, at lower levels of abstraction we have models of the detailed system design. These models typically capture design elements that have a direct correspondence to specific implementation artifacts. Functional design models are relatively well-established, with the dominant model (arguably) now UML [14]. UML can be used to model both detailed design and higher-level designs through a complex suite of diagrams that are all treated as views onto a consistent underlying model. Whilst UML is effective in terms of modelling system functionality as well as data relationships, in terms of modelling the information design the situation is somewhat less mature. Typically we wish to model not only the data (i.e. content) itself, but also the relationship between the underlying content and the user-perceived views of that content, and the interactions with those views (such as navigational aspects). Given that UML has not been as successful at modelling these aspects, there has been an emergence of a number of informational modelling approaches specifically developed for Web (or hypermedia) applications. Example approaches include RMM (Relationship Management Methodology) [15] and OOHDM (Object Oriented Hypermedia Design Methodology) [16], and more recently WebML (Web Modeling Language) [12]. These models have typically focussed on modelling at a relatively low-level and have failed to address architectural and even business process modelling issues.

In between modelling the business operations and the detailed system designs we have models of the high level system architecture that capture the functionality and information that need to exist in order to support the business value exchanges. As discussed previously, functional aspects at this level are well supported by UML. A set of UML models can be used to represent the business and operational workflows (process) of a system. Examples of this include activity diagrams for presenting the sequential flow of activities and data flow diagrams for presenting the flow of data from external entities into and within the system.

However, modelling informational aspects at this intermediate level is more problematic – particularly in terms of explicitly showing the relationships between business models and processes as well as detailed information designs. For example, UML activity diagrams focus on the representation of activities and events (which can be seen as the functional aspects of business processes [17]), but are unable to accurately capture the rich domain context that is important in understanding information relationships and flows (as distinct from data flows) that support value exchanges in business. This includes 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. Interesting discussions on gaps in modelling can be found in [3].

2.2 WIED

In response to this problem, we have proposed a new modelling notation (which we call WIED - Web Information Exchange Diagram) aimed at representing information flows that occur between actors for supporting value exchanges in businesses [7–9]. The proposed model also represents the relationships between the underlying information and the actor interaction with the information. In effect, our model forms the bridge between business concerns and detailed information designs which we have argued is largely lacking in existing models [10].

The abstract WIED model has been formalized as an XML DTD. To illustrate the application of the model we mapped it into a custom diagram and notation that is consistent with WebML. In this context it can be considered as a companion model to the existing WebML diagrams (and hence, in our earlier work, we referred to it as WebML+).

Figure 1 shows an example of a WIED model (using the WebML-compliant notation) for a hypothetical example: TicketMaster is a ticketing partner for the sports and entertainment industries. The strategy adopted by TicketMaster is to expand their distribution channel by introducing real-time transactional ticketing through the Internet as well as enabling customers to enquire about the latest information on shows and events.

Let us consider what is represented in the model. The organization boundary (shown as a dashed geometrical polygon) encloses a set of information units. These units represent coherent and cohesive domains of information that are managed or utilized by the organization. All information within a single unit shares a common context and a common derivation (this is explained shortly). They do not map directly to pages or sets of pages – a single web page may contain partial information from multiple units. Similarly, an information unit may be distributed over multiple pages. Different types of information units are represented graphically using different types of icons. Some information units are provided directly by actors, whereas others are derived from other units. These derivations (shown as triangles with incoming and outgoing arrows) capture the inter-relationships between the information units. Furthermore, the system boundary (shown as a dotted geometrical polygon) encloses only the set of information units that are utilised and/or managed by the system under consideration. (i.e. The elements enclosed by the system boundary are a subset of the elements enclosed by the organisation boundary. The organisation boundary captures the interface between the organisation and external stakeholders (i.e. crucial in supporting the business modeling). Conversely, the system boundary captures those elements of the organisation that can be managed by the system (i.e. crucial in supporting the detailed system design). Further details and examples can be found in [7–9].

3 WIED for UML

Our previous research has shown the value of WIED in supporting the design process (particularly in terms of ensuring that the implications for business models of changes to systems designs are well understood). We do however recognise the wide adoption of UML as a modelling notation, and so wish to show how the abstract WIED model can be readily mapped into a UML-compliant notation, resulting in a new WIED-UML diagram that is consistent with existing UML diagrams. To represent the WIED model in UML-compliant notations we need to map the various WIED entities to equivalent UML entities (which have been stereotyped for clarity). This gives a notational consistency with UML - but a new UML diagram which is the aggregation of the WIED entities. The UML-compliant version of WIED is expressed in terms of stereotypes, tagged values and constraints. Next, we present the various WIED entities represented by UML-compliant notations. In discussing the WIED UML notations we will also show the original WIED WebML notations for the purpose of demonstrating the one-to-one mapping between these notations.

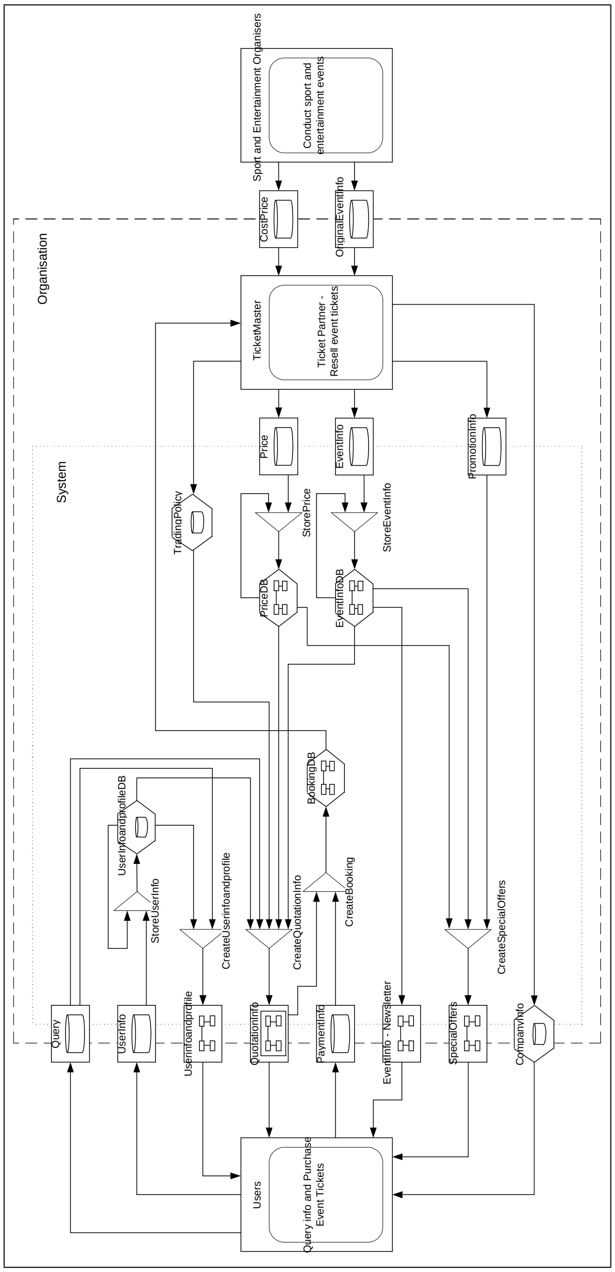

Fig. 1. Typical Web System represented using WIED

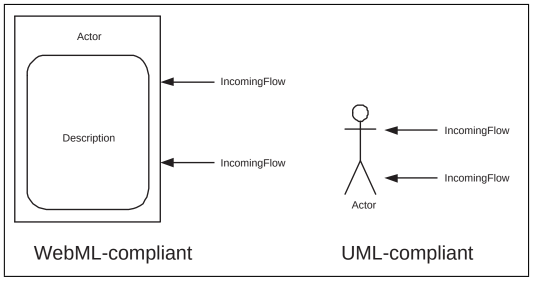

Fig. 2. Actor Units

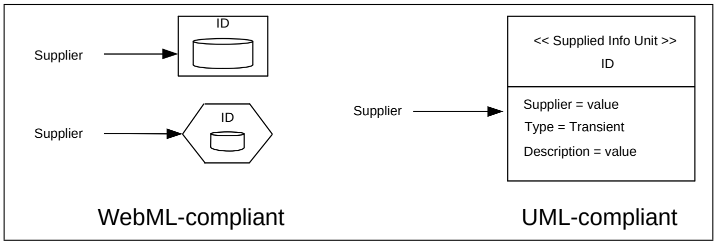

Fig. 3. Supplied Information Units

Actor Units : Actor units are defined to show the roles that the users play with respect to the system. In WIED, actor units show how users participate with information flows. Figure 2 shows actor units represented in WebML-compliant notation and UML-compliant notation.

Supplied Information Units and Derived Information Units : In general, an information unit represents a single information object in an organization business process. To represent information units using UML, classes are used since they describe the types of objects in the system and the relationships that exist between them. Two types of information units are defined in WIED: supplied information units and derived information units. A supplied information unit presents information about a single information object that is provided by a supplier actor. A stereotype «SuppliedInfoUnit» (with attributes such as supplier, type and description) is used for representing a supplied information unit (as illustrated in Figure 3). A derived information unit presents information that is derived from other information units. This derivation may ultimately be implemented as a process that generates the underlying content. To specify a derived information unit we provide both the unit itself (represented again using a stereotype: «DerivedInfoUnit») and an indication of the source of the information (represented using a transition, since a transition describes a change of an object from one state to another). This is illustrated in Figure 4

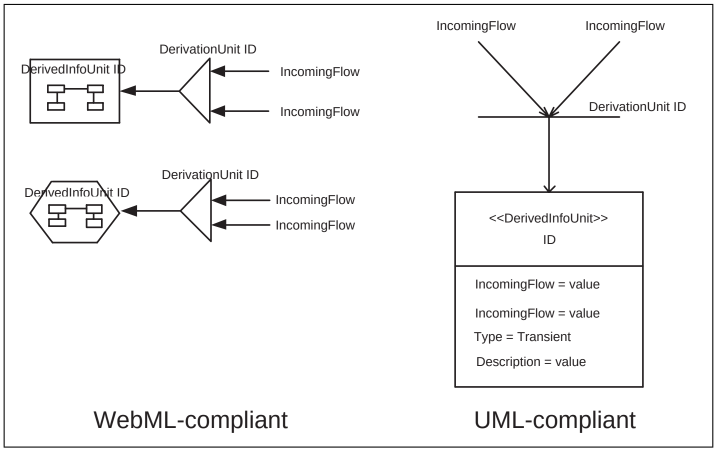

Fig. 4. Derived Information Units

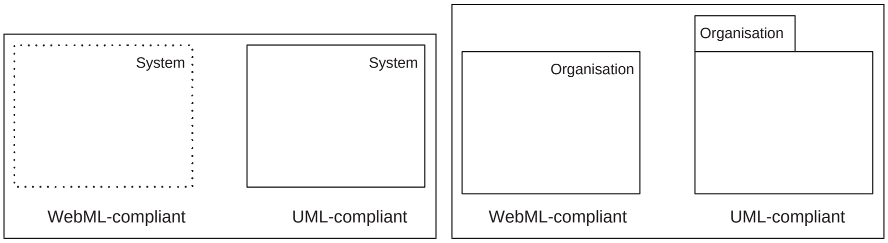

Fig. 5. Systems and Organisations

Information flows : An arrow represents flow of information in both WIED custom notations (WebML compliant) and UML compliant version of WIED.

System and Organisation : The system boundary encloses the set of information units that are utilized and/or managed by the system under consideration, and the organisation encloses the set of information units that are utilised and/or managed by the organisation. The system can be considered as a super class of a set of information units, and so can also be represented as a class. The organization on the other hand can be considered as a superset of the system where some information units are exchanged within the organization (and with external actor) but not in the system. A UML package is preferred for representing the organization since in some cases more than one system are used in the organization and a system can be distributed across multiple organisations. Figure 5 shows system and organisation units represented in WebML-compliant notation and UML-compliant notation.

Figure 6 shows an example of a WIED model for the same hypothetical example discussed earlier: TicketMaster.

4 Discussion

Representing the WIED model using UML compliant notations enables us to create relevant linkages (in some aspects) between the WIED model and other aspects of the overall UML-modelled systemm. For example, we can map between the information flows (as represented by our UML-compliant WIED model) and the business process and workflows (as represented using a UML activity diagram). Similarly we can then map from the information flows to the detailed information designs (often represented using class diagram, state diagrams, etc.) This can be seen as a forming of the bridge between informational aspect and functional aspect of high level system architecture which leads to a solution for resolving a problem of disconnection between these two aspects (one of crucial problems in Web system development [3] since the problem is caused by an absence of informational aspect modelling at this level of abstraction). The following key subjects are considered for mapping. First, business actors, who play roles in workflows (activity diagram), can be mapped directly to be actors in information flows (WIED). Second, the scenarios are combined to produce comprehensive information flows. Then, where a significant information unit is triggered by an information flow, adding derivations. Figure 7 shows an example of an activity diagram that is mapped from the same hypothetical example discussed earlier: TicketMaster.

Another interesting aspect is how an information unit (represented as a class) might be deconstructed into sub-classes which represent the actual information units in the conceptual schema and then subsequently into the navigational structure [18]. Previously, we have proposed similar guidelines for this aspect, mapping the WIED (WebML-compliant) to structural schema and system siteview represented using WebML notations [10]. So, when the WIED is alternatively represented using UML, guidelines for mapping the WIED (UML compliant) to UML-based approach such as Conallen's UML extensions for Web applications [19] and other similar approaches (e.g. Baumeister's approach [18]) will be also considered. Figure 8 and Figure 9 show examples of a navigational structure and sub-classes representing the actual information units respectively,these are mapped from the same hypothetical example discussed earlier: TicketMaster. Moreover, linkages between the WIED model and business models is also relevant. These two latter aspects will lead to the important issue of forming the bridge between UML-based system design and business model (another crucial problem in Web system development [3]). This concept has been addressed and demonstrated previously using WIED (WebML-compliant) [10]. So we believe that the UML compliant WIED will replicate what have been done by the WebML compliant version of WIED with more compatibility with the standard modelling, UML (i.e. be operationalized as a notation and associated diagram that are improved consistent with UML).

Space limitations preclude us from including the detailed procedures how to map the WIED (UML compliant) to other UML-based modelling approaches. These will be published elsewhere.

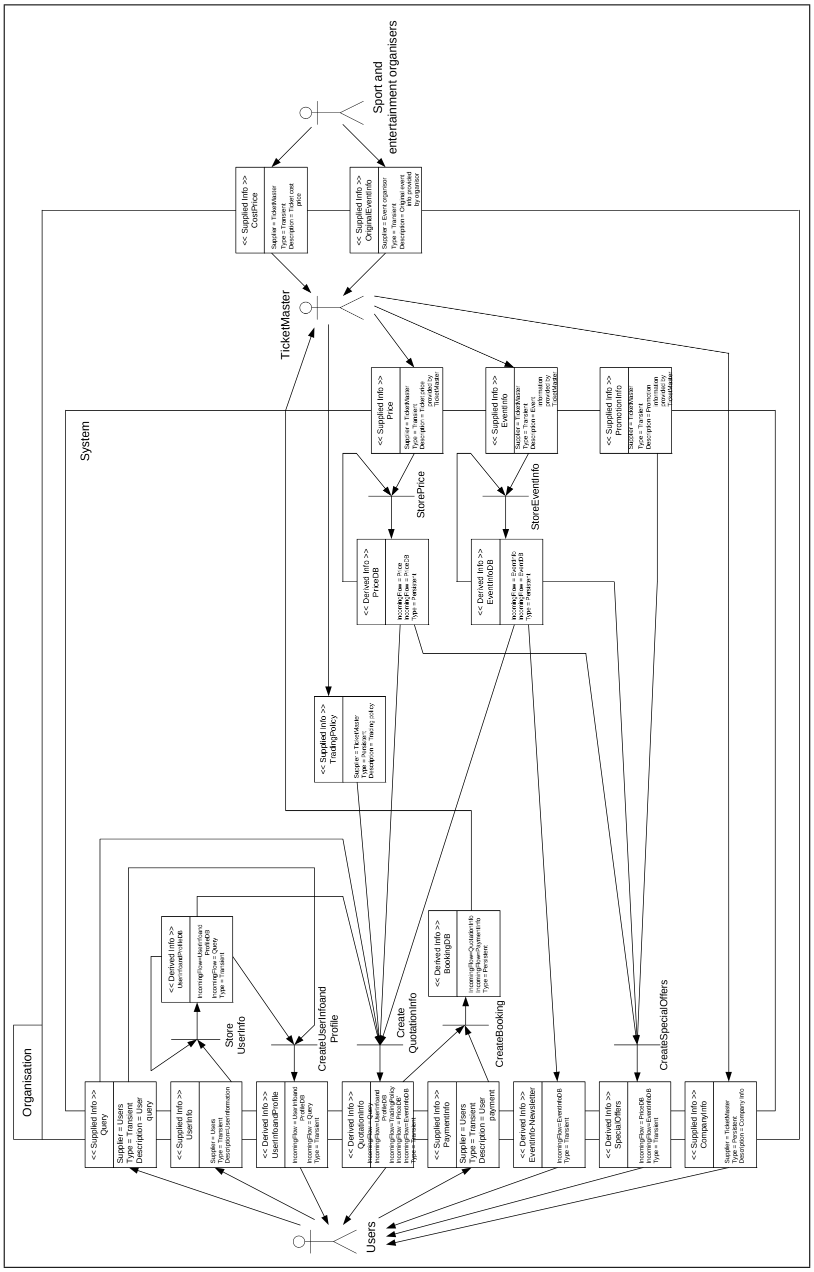
Sport and entertainment organisers
<< Supplied Info >> CostPrice
Supplier = TicketMaster
Type = Transient
Description = Ticket cost price
<< Supplied Info >> OriginalEventInfo
Supplier = Event organisor
Type = Transient
Description = Original event info provided by organisor
TicketMaster
<< Supplied Info >> Price
<< Supplied Info >> EventInfo
<< Supplied Info >> PromotionInfo
System
StorePrice
StoreEventInfo
<< Derived Info >> PriceDB
<< Derived Info >> EventInfoDB
<< Supplied Info >> TradingPolicy
Supplier = TicketMaster
Type = Persistent
Description = Trading policy
<< Derived Info >> UserInfoandProfileDB
<< Derived Info >> BookingDB
Store UserInfo
CreateUserInfoand Profile
Create QuotationInfo
CreateBooking
CreateSpecialOffers
Organisation
<< Supplied Info >> Query
Supplier = Users
Type = Transient
Description = User query
<< Supplied Info >> UserInfo
<< Derived Info >> UserInfoandProfile
<< Derived Info >> QuotationInfo
<< Supplied Info >> PaymentInfo
Supplier = Users
Type = Transient
Description = User payment
<< Derived Info >> EventInfo-Newsletter
<< Derived Info >> SpecialOffers
<< Supplied Info >> CompanyInfo
Users

Fig. 6. Typical Web System represented using WIED-UML

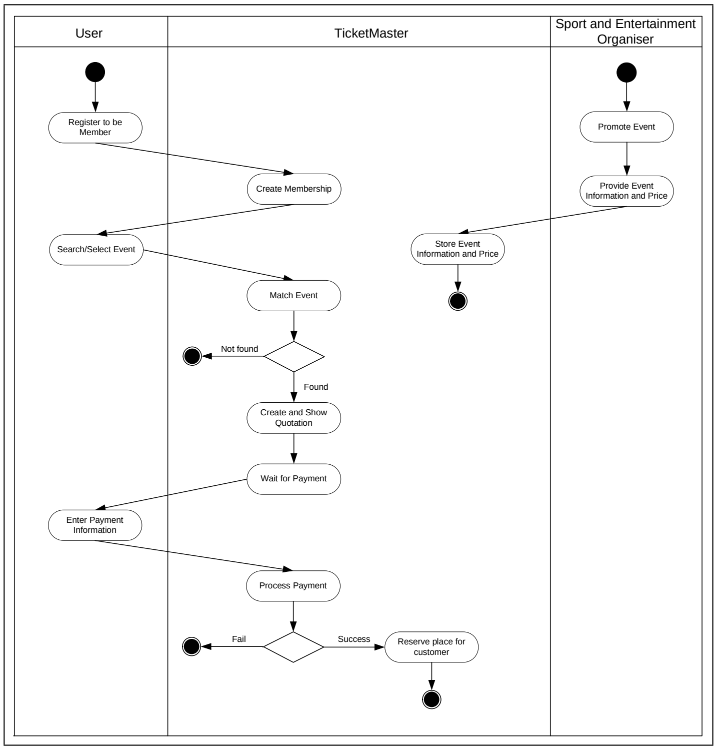
User
TicketMaster
Sport and Entertainment Organiser
Register to be Member
Create Membership
Search/Select Event
Match Event
Not found
Found
Create and Show Quotation
Wait for Payment
Enter Payment Information
Process Payment
Fail
Success
Reserve place for customer
Promote Event
Provide Event Information and Price
Store Event Information and Price

Fig. 7. Typical Activity Diagram representing Business Processes

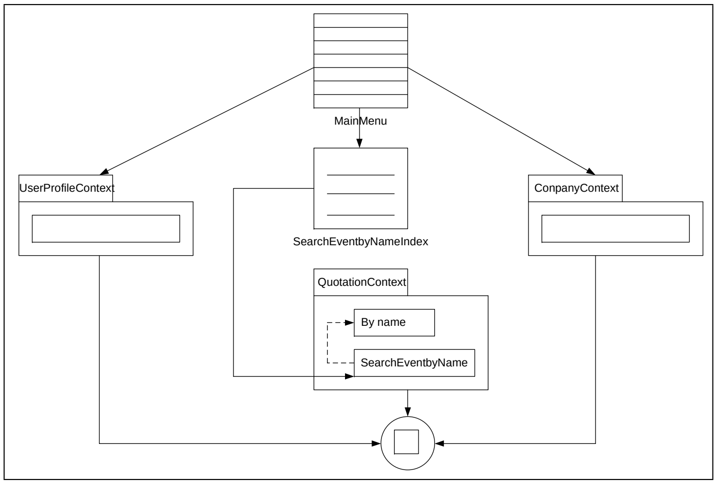

Fig. 8. Partial Navigational Structure represented using Baumeister's approach

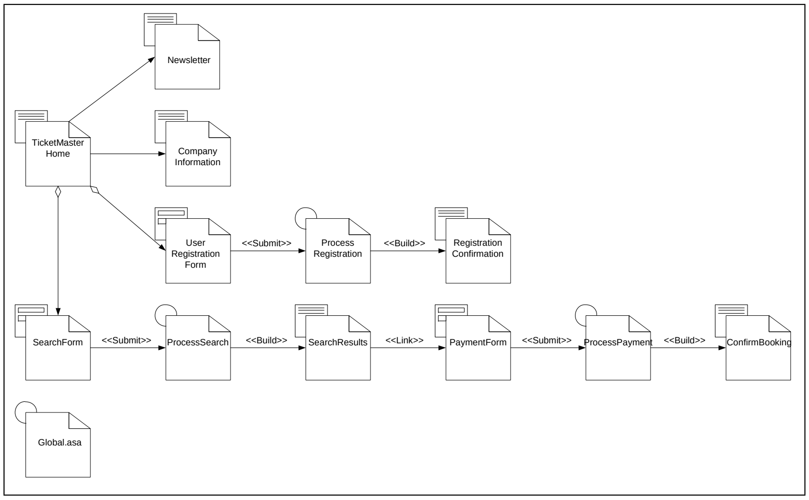

Fig. 9. Partial Browsing Overview Class Diagram using Conallen's approach

5 Conclusion and future work

In this paper, we have demonstrated how the WIED model can be mapped into UML compliant notations and a new UML diagram. We argue that this enables WIED to be compatible with UML and provide improved connections between WIED and other models that are typically constructed. Therefore, it will lead to a more standardized WIED and subsequently enhance the design process of Web-enabled systems. We believe that this will also assist developers and clients, who use UML-based notations in their system developments, in understanding the impact on business process and models which arise from changes that are made to the underlying information designs (as has been illustrated in our previous work with WebML-compliant WIED.

In this work, we have also briefly proposed guidelines to support the mapping process linking the WIED (UML-compliant) to other modelling approaches based on UML. This should be a good start for making the WIED to be practically companion to existing widely-used UML models and potentially integrate with those models to create a better UML-based modelling suit for Web system developments.

One key aspect of supporting the effective use of the models described in this paper would be modelling tool support which allows the capture of (and ultimately reasoning about) the relationships between the WIED models and the associated business models and detailed design models. This would support developers in more effectively managing these relationships.

Ongoing work is focusing on clearly defining and improving linkages between the WIED (UML-compliant) and UML-based modelling approaches. We will also possibly conduct an empirical study for evaluating the WIED (UML-compliant) model. The outcomes of improved mapping between modelling approaches and the evaluation will be reported elsewhere.

References

1. Shelford, T.J., Remillard, G.A.: Real Web Project Management: Case Studies and Best Practices from the Trenches. 1st edn. Addison Wesley Professional (2002)
2. Powell, T.A.: Web Site Engineering. Prentice-Hall (1988)
3. Gu, A., Henderson-Sellers, B., Lowe, D.: Web modelling languages: The gap between requirements and current exemplars. In: AusWeb02, Gold Coast, Australia (2002)
4. Kappel, G., Proll, B., Retschitzegger, W., Hofer, T.: Modeling ubiquitous web applications - a comparison of approaches. In: The International Conference on Information Integration and Web-based Applications and Services, Austria (2001)
5. Christodoulou, S., Styliaras, G., Papatheodourou, T.: Evaluation of hypermedia application development and management systems. In: ACM Hypertext'98 Conference, Pittsburgh (1998)
6. Lowe, D.: Web requirements: An overview. Requirements Engineering Journal (2003)
7. Lowe, D., Tongrungrojana, R.: WebML+ for communication of information flows : An empirical study. In: the Third International Conference on Web Engineering, Oviedo, Asturias, Spain (2003)
8. Tongrungrojana, R., Lowe, D.: Modelling forms of information derivation in modified WebML+. In: the Ninth Australian Web Conference, Gold Coast, Australia (2003)
9. Tongrungrojana, R., Lowe, D.: WIED : a web modelling language for modelling architectural-level information flows. Journal of Digital Information (2004)
10. Tongrungrojana, R., Lowe, D.: WebML+: a web modeling language for forming a bridge between business modeling and information modeling. In: the Fifteenth International Conference on Software Engineering Knowledge Engineering, San Francisco (2003)
11. Gordijn, J., Akkermans, J.: e^3-value: Design and evaluation of e-business models. IEEE Intelligent Systems **16** (2001) 11–17
12. Ceri, S., Fraternali, P., Bongio, A.: Web modeling language (webml): a modeling langauge for designing websites. In: the Proceeding of WWW9 Conference, Amsterdam (2000)
13. Gordijn, J.: e^3-value in a nutshell. In: International workshop on e-business modeling, Lausanne (2002)
14. OMG: Introduction to omg's unified modeling language (uml) (2003)
15. Isakowitz, T., Stohr, E., Balasubramanian, P.: RMM: A methodology for structured hypermedia design. Communications of the ACM **38** (1995) 34–44
16. Schwabe, D., Rossi, G.: Developing hypermedia applications using oohdm. In: Workshop on Hypermedia Development Processes, Methods and Models (Hypertext'98), Pittsburgh, USA (1998)
17. Lieberman, B.: Uml activity diagrams: Detailing user interface navigation (2001)
18. Baumeister, H., Koch, N., Mandel, L.: Towards a UML extension for hypermedia design. In: «UML»1999: The Second International Conference on The Unified Modeling Language, Fort Collins, Colorado, USA, IEEE (1999) 614–629
19. Conallen, J.: Modeling Web application architectures with UML. Communications of the ACM **42** (1999) 63–70