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 e3-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 [211].

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 [315]). 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 [7189].

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 e³-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 e³-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