Form-and-Fact Based Modeling

Terry Halpin[1] and Gerald Weber[2]

[1] LogicBlox, Australia and INTI International University, Malaysia
[2] Department of Computer Science, The University of Auckland, New Zealand
terry.halpin@logicblox.com, gerald@cs.auckland.ac.nz

Abstract. A conceptual data model for an information system specifies the fact structures of interest as well as the constraints and derivation rules that apply to the business domain being modeled. Fact-based modeling approaches provide rich graphical and textual languages for specifying conceptual data models, using attribute-free fact structures that enable models to be verbalized and populated in natural sentences that are easily understood by the domain experts best qualified to validate the models. Form-based modeling approaches offer a natural way for domain users to agree upon suitable user interfaces for interacting with the information system. This paper proposes a synthesis of the two approaches, in which prototype forms are used to seed the conceptual data model, which is then used to generate the final user interface. Semantic and practical aspects of form design are discussed, and screen transition diagrams are employed to help visualize and validate the underlying dynamic processes.

1 Introduction

Computer-based information systems model information about the relevant business domain at various levels. For persistent storage, data is often maintained in structures such as relational database tables, XML schema documents, RDF triple stores, or deductive database clause sets. Although such data structures could be directly updated by the technically savvy, practical information systems typically provide higher level user interfaces, often based on screen forms, to enable users to interact with the system without needing to master the technical aspects of the internal storage structures. Many different user interfaces could be designed for the same information. Moreover, information might need to be displayed in different ways for different user groups. Partly for such reasons, data requirements for an information system are often captured in a *conceptual data model* that provides a simpler, more fundamental way of specifying the fact structures and instances of interest as well as the business rules (constraints or derivation rules) that apply to the relevant business domain.

A conceptual model should be cast in terms of concepts that are intelligible to the business users, so that it can be used to validate the model with them. Once validated, the conceptual model may be implemented using a variety of user interfaces and internal structures, including those used for transient storage of the data (e.g. object oriented code structures, or deductive clause sets) and those used for persistent storage of the data.

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 285–299, 2012.
© Springer-Verlag Berlin Heidelberg 2012