IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB8083

Chapter XVII

Event Modeling

Lars Bækgaard
Independent Consultant, Denmark

INTRODUCTION

The purpose of this chapter is to discuss conceptual event modeling within a context of information modeling. Traditionally, information modeling has been concerned with the modeling of a universe of discourse in terms of information structures. However, most interesting universes of discourse are dynamic and we present a modeling approach that can be used to model such dynamics.

The Unified Modeling Language (UML) is based on the following notion of an event. "An event is a noteworthy occurrence that has location in time and space. It occurs at a point in time; it does not have duration. Model something as an event if its occurrence has consequences." (Rumbaugh, Jacobsen et al. 1999). This definition emphasizes that an event occurs at a point in time and that its occurrence has noteworthy consequences.

We characterize events as both information objects and change agents (Bækgaard 1997). When viewed as information objects events are phenomena that can be observed and described. For example, borrow events in a library can be characterized by their occurrence times and the participating books and borrowers. When we characterize events as information objects we focus on concepts like information structures. When viewed as change agents events are phenomena that trigger change. For example, when borrow event occurs books are moved temporarily from bookcases to borrowers. When we characterize events as change agents we focus on concepts like transactions, entity processes, and workflow processes.

Record-based modeling languages like the relational model (Codd 1970) support information modeling in terms of records on the form ($value_1$, …, $value_n$). In nested models each value may itself be a record (Thomas and Fischer 1986). Record-based models are not suited for conceptual information modeling (Kent 1979). Furthermore, these models offer a rather low-level support of the modeling

The research described was carried out while Lars Baekgaard was employed at the Aarhus School of Business, Denmark. This work was supported in part by the Danish Social Science Research Council, Project No. 9701783.

of events as change agents in terms of commands that insert, update, and delete records. For example, the relational query language SQL supports the specification of manipulation commands and transactions in a non-conceptual way. For these reasons record-based modeling languages are not suited for conceptual event modeling.

Entity-based modeling languages like the entity-relationship model (Chen 1976) and the functional model (Shipman 1981) were developed to overcome some of the modeling weaknesses of record-based modeling languages (Kent 1979). These modeling languages support the modeling of information structures in terms of concepts like entities, relationships, and attributes in ways that are very similar to our modeling approach. However, conventional entity-based languages do not support the modeling of dynamics.

Object-based modeling languages support information modeling in terms of relationship structures, generalization/specialization hierarchies, and aggregation structures (Booch, Rumbaugh et al. 1999), (Kim and Lochocsky 1989), (Kim 1995), (Rumbaugh, Jacobsen et al. 1999), (Zdonik and Maier 1990). An object is a totality of a unique object identifier, state, and behavior. The state is represented by a set of attributes. Individual object behavior is modeled in terms of object services. The object-based languages suffer from an asymmetric modeling of events as change agents due to their object-level coupling of state and behavior. Event classes that are shared by two or more object classes must be defined in a fragmented way as services attached to the participating objects. For example, if borrow events in a library are modeled as change agents they must be split into two object services—one service is attached to borrower objects and the other service is attached to book objects. We model events as change objects in a non-fragmented way.

Entity life cycles (Rosenquist 1982), (Jackson 1983), (Bækgaard 1993), (Bækgaard 1997) (Bækgaard and Godskesen 1998) and object life cycles (Kappel and Schrefl 1991) can be used to model global entity/object dynamics. Conventional entity/object life cycles are specified as regular expressions that are used to constrain the sequencing of events (Jackson 1983). UML supports graphical specification of regular expressions that are extended with concurrent sub cycles (Booch, Rumbaugh et al. 1999), (Rumbaugh, Jacobsen et al. 1999).

Our approach to event modeling supports the modeling of events as both information objects and change agents by combining entity-based information modeling (Hammer and McLeod 1981), (Shipman 1981) with an approach to dynamic modeling that avoids the disadvantages of object-based languages. Our modeling approach is based on a modeling language called EVER (the event-entity-relationship model). We model entity processes in a way that is more expressive and flexible than approaches based on regular expressions.

In the next section we discuss conceptual modeling of events. We then present an entity-based approach to the modeling of events as information objects. Then in the next section we show how to model events as change agents, i.e., as phenomena that have a changing effect when they occur. We thenshow, in the next section, how events can be aggregated into entity processes. Finally, we conclude the chapter.

Related Content

National Game Management Database of Hungary

Sándor Csányi, Róbert Lehoczkiand Krisztina Sonkoly (2010).

International Journal of Information Systems and Social Change (pp. 34-43).

www.irma-international.org/article/national-game-management-database-hungary/47180