Using Harel’s Statecharts to Model Business Workflows

Wai Yin Mok, University of Alabama in Huntsville, USA
David Paper, Utah State University, USA

ABSTRACT

In this paper, we model business workflows using Harel’s statecharts. Mapping to statecharts allows us to systematically identify potential workflow problems. It also allows us to investigate specific properties inherent in actual business workflows. Our research focuses on three desirable properties of active database systems — termination, confluence, and observable determinism. For termination and confluence, we develop algorithms to provide a theoretical lens linking desirable active database system properties to workflow management systems problems. We initially validate our algorithms by mapping business workflows from a case study. Our research thus builds preliminary theory by developing a systematic method for identifying workflow problems.

Keywords: Business workflows, workflow management systems

INTRODUCTION

Business workflows can be well defined, predictable, and frequently executed. We refer to these as structured business workflows. Workflows with these characteristics can be automated by machines to reduce clerical tasks and potential human intervention errors. We can use workflow management systems (WMS) to facilitate automation of structured business workflows. WMS, which are new generations of computerized systems, are designed to manage automated parts of business workflows (Brunwin, 1994). By separating workflow definitions from application software, WMS provide process and knowledge
independence, much like data independence provided by database management systems.

We propose using Harel’s statecharts to model structured business workflows (Harel, 1987) for three reasons. First, Harel’s statecharts are used in the Unified Modeling Language (UML) as a means for behavioral modeling (Object Management Group, 1999). Since the UML is the standard modeling language of the Object Management Group, Harel’s statecharts will soon become common. Second, statecharts are easy to understand and they do not have the problem of exponential growth of states that plague ordinary state transition diagrams (Harel, 1988). We shall elaborate this point in the Related Work section. Third, their semantics are rigorous enough for formal analysis on various aspects of structured business workflows (Harel and Naamad, 1996).

In the framework of statecharts, we will show how to model workflow concepts and present algorithms that determine whether a given business workflow has certain predefined properties. We will then use a case study with Moore BCS to explore the characteristics of a business workflow. The algorithms we develop in this study will become part of a software design tool that we will develop in the future.

RELATED WORK


Active database systems have been studied extensively (Paton and Diaz, 1999). Active database systems and workflow management systems are related since both types of systems employ triggers to respond to external and internal events and exceptions. We are interested in three important properties of active database systems in this paper, namely termination, confluence, and observable determinism, which are formally defined in Allen, Hellerstein, and Widom (1995). More discussion on active database systems, which include several research prototypes and commercial products, can be found in Zaniolo (1997).

The statemate approach, which uses statecharts in modeling reactive systems, is described in Harel and Politi (1998) and its semantics in Harel and Naamad (1996). By far, the statemate semantics of statecharts is the most rigorous and precise execution semantics defined for statecharts and it has been in use for more than ten years (Harel and Naamad, 1996). Here we point out the most significant aspects of the execution semantics. The reader may consult Harel and Politi (1998) and Harel and Naamad (1996) for details.

The behavior of a system described in statemate semantics is a set of possible runs, each representing the responses of the system to a sequence of external stimuli generated by its environment. A run consists of a series of detailed snapshots of the system’s situation; such a snapshot is called a status. The first in the sequence is the initial status, and each subsequent one is obtained from its predecessor by executing a step (see Figure 1).
Related Content

Scalable QSF-Trees: Retrieving Regional Objects in High-Dimensional Spaces
[www.igi-global.com/article/scalable-qsf-trees/3315?camid=4v1a](www.igi-global.com/article/scalable-qsf-trees/3315?camid=4v1a)

The Benefits of Data Warehousing at Whirlpool
[www.igi-global.com/chapter/benefits-data-warehousing-whirlpool/6209?camid=4v1a](www.igi-global.com/chapter/benefits-data-warehousing-whirlpool/6209?camid=4v1a)
Research on Improved Method of Storage and Query of Large-Scale Remote Sensing Images
www.igi-global.com/article/research-on-improved-method-of-storage-and-query-of-large-scale-remote-sensing-images/218924?camid=4v1a

Semantic Integrity Constraint Checking for Multiple XML Databases
www.igi-global.com/article/semantic-integrity-constraint-checking-multiple/3360?camid=4v1a