Handling Change Management using Temporal Active Repositories

Avigdor Gal and Opher Etzion

Information Systems Engineering Department
Faculty of Industrial Engineering and Management
Technion - Israel Institute of Technology
Haifa, Israel, 32000
Email: [avigal, ieretzn]@ie.technion.ac.il

Abstract. Business Re-engineering requires frequent changes in the enterprises' information systems, however the current technology of data dictionaries is not effective for the tracing of required changes and their management. In this paper we introduce an architecture of change management using active temporal repositories. Flexible change management allows the support of information about past or future versions of information systems, as well as the capability to retrieve and update temporal information. The implementation of change management in a temporal environment is carried out by the partition of the temporal universe among temporal agents, each of them handles a single version of an application with a required collaboration among them. The change management process, and the inter and intra agent processing are described in this paper.

keywords: change management, information agents, cooperative databases, temporal databases, active databases

1 Introduction and Motivation

1.1 Background

In today's business environment, re-engineering has become a vital process in many enterprises [11]. The process of re-engineering involves changes in the enterprise's structure and processes to meet its evolving goals in a constantly changing environment. Alas, current information technologies, using abstractions such as data dictionaries, form an obstacle to the implementation of re-engineering due to difficulties in tracing the consequences of such changes on the information systems and on the application programs.

Data dictionaries [1] have been proposed as abstractions of concepts that are vital to model information systems. However, the contribution of data dictionaries to change management has been marginal due to the following three major problems:
• Data dictionaries are loosely coupled with the actual applications. Consequently, there is no automatic way to infer the implication of a change in an application. Modelling such an implication requires manual intervention; thus, the reliability of such information is left to the user discretion, resulting in frequent inconsistencies.

• Data dictionaries are passive in the sense that they only document information about the application but do not include automatic tools to initiate actions in the wake of a change in any of the application’s components.

• Many application systems maintain concurrently active versions. Changes in the applications may refer to various versions or create a new one. Data dictionaries are unable to assist in change management over versioned systems, since they do not have temporal capabilities.

In this paper we propose an architecture for a change management based on a combination of several technologies:

• The information repository technology which controls the application behavior in a tightly-coupled manner [12].

• The object oriented technology that enables flexible structuring.

• The active technology that supports automatic activation of derived changes [7].

• The temporal technology which supports several versions of the information repository entities and sustain the modelling of time characteristics for any information about the application [15].

1.2 A Motivating Example

Throughout this paper we use the following example as both a motivating example and a demonstration of the change management architecture capabilities.

A distributed information system consists of a global information repository and local information repositories with local agents. The global repository includes information about globally compatible data-items. Globally compatible data-items are data-items whose type is enforced to be the type defined in the global repository. We assume a shared naming conventions to avoid semantic conflicts such as synonyms and homonyms.

Each data-item has an accessibility indication that states whether it is shared with other repositories or it is a private data-item. For simplicity, we assume that if a data-item is shared, it is accessible to all repositories. Unlike shared data-items, private data-items do not have to be globally compatible, even if they have the same name. Both the type and the accessibility indication of each data-item may change over time, e.g., a data-item may be accessible during certain temporal element and inaccessible during other temporal element.

The rest of the paper is organized as follows. Section 2 describes the data model of a temporal active database, based on [4]. Section 3 discusses the change management process. Section 4 concludes the paper.