December 2001

Derivation of an Append-Only Data Model

William Leigh
University of Central Florida

Mario Paz
University of Louisville

Noemi Paz
Science Applications International Corporation

DERIVATION OF AN APPEND-ONLY DATA MODEL

William Leigh
University of Central Florida
leigh@pegasus.cc.ucf.edu

Mario Paz
University of Louisville

Noemi Paz
Science Applications
International Corporation

Abstract

Title 21 of the Code of Federal Regulation – "Food and Drug Administration" mandates that all records which are part of the test procedures of a new drug be maintained as part of an integral audit trail. One interpretation of this requirement is that the involved database system must be append-only, that is, that there may be no Update or Delete transactions, only Insertions. We outline a method for deriving a data model to support such a system.

Overview

The proposed method proceeds through the systems transformation approach to design. Relational database normalization is a well-known design method also based on systems transformation. This proposed method proceeds in the same spirit as relational database normalization, though at a different information system level. And, like database normalization, this process is guided by a semantic understanding of the domain of the application system, with alternative design choices being possible at each juncture.

In the proposed method, the original system is represented in a matrix, and this matrix is changed to new configurations having desirable qualities missing from the original version. The subsequent versions of the system maintain their degree of "truth" as a solution to the original problem. The form of the information system that is the objective of the method has only insertion (no update or delete) transactions affecting the database. This append-only form has desirable qualities including the presence of an integral audit trail, simplified security, and a highly structured interface with client-side processing components.

In the method proposed in this paper, the first transformation of the system representation, to a first standard form, achieves a direct correspondence between each cell in the matrix and a state transition of the real-world objects of the original system. The second transformation, to a second standard form, adds data objects to the information system to achieve a 1-to-1 correspondence between the state transitions and the final data object set of the transformed system.

System Representation

Programs in transaction processing systems display forms for data entry and then take that data and apply it to data objects as transactions, which are comprised of row inserts, updates, and deletes. Application logic and programmer ingenuity dictate the order of these operations. A matrix, which is similar to the object interaction diagram employed in many object-oriented design methods, relates transaction events (and their associated data entry forms), data objects, and the processing operations (Insert, Update, and Delete.)

First Standard Form

In the "first standard form" of the matrix representing the system:

- Each row of data operations in the object is implemented as a single database transaction;
- There is only a single Insert, Update, or Delete in each cell;

- The topmost operation appearing in each column is an Insert;
- The column may contain any number of Updates below the Insert;
- A single Delete may appear below the Updates.

Second Standard Form

To achieve the "second standard form", all instances of the update and delete operation are removed. Now every row and column contains one and only one Insert. A complete audit trail exists for the system, as nothing is updated-in-place and nothing is deleted.

State Transitions

Note that in first standard form there exists a direct correspondence between a matrix column and the state-transition-diagram for the data object entity associated with that column. Each cell of the matrix columns corresponds to a state transition.

In second standard form, an object is added to the database to correspond to each transition in the state transition diagrams. This makes possible to accommodate the restriction to insert-only without loss of processing functionality. Insertions into the new data objects denote the changes of state for the associate objects.

Summary and Conclusion

This proposed normative reverse engineering process prompts the discovery of additional events and data entry forms, and of additional data objects. In the specific example shown (in the presentation at the conference), three data entry forms (which might be only one multi-function form in many legacy implementations) becomes five, and one data object becomes five. Like the addition of entity types to the database design in the standardization process, the addition of forms and objects here results in a system, and a user interface, in particular, which is more easily implemented, is more complete, is more naturally modular, and is inherently more controllable.

The insert-only characteristic of the resulting design is a primary device to enhance controllability. That the client side only retrieves and inserts data to the server side, and does not update or delete, allows all data entered to be saved on the server, becoming a complete and secure audit trail. That the client-side adheres to this protocol may be insured through use of standard database management system security mechanisms.