MEDINFO 2007
HISA Ltd Australia
© 2007 The authors. All rights reserved.

Supporting Cooperative Updates of Clinical Guidelines

Alessio Bottrighi[b], Luca Anselma[a], Stefania Montani[b], Paolo Terenziani[b], Gianpaolo Molino[c], Mauro Torchio[c],

[a] *DI, Università di Torino, Corso Svizzera 185, 10149 Torino, Italy, E-mail: anselma@di.unito.it*
[b] *DI, Univ. del Piemonte Orientale "Amedeo Avogadro", Spalto Marengo 33, 15100 Alessandria, Italy, E-mail: {terenz,stefania,alessio}@mfn.unipmn.it*
[c] *Azienda Ospedaliera S. Giovanni Battista, Via Bramante 88, 10100 Torino, Italy*

Abstract

We propose an extension of the standard DB model in order to support cooperative updates to clinical guidelines, as needed both in the acquisition phase and when guidelines are modified to capture new therapies.

Introduction and background

Modelling guidelines is a complex task, which may require a cooperative effort of several participants. Alternative proposals may be generated, and a team of "supervisors" is needed in order to validate and choose between them. Maintaining the authors of each proposal and the history of the different versions of the guidelines is very important (e.g., to support the retrieval of past version of the guideline, to justify decision taken in the past). A temporal extension of the standard DB models is thus needed to support such a cooperative and dynamic environment.

Methods

In order to devise a modular, general, and system independent approach to face the above issues, we propose a three-layered architecture, consisting of:

1. a data model layer, defining the data model and providing the basic operations;
2. a query language layer, supporting an SQL-like high-level manipulation and query language, based on layer (1);
3. an interface layer, based on the previous ones, that provides users with high-level functionalities, accessed through a user-friendly graphical interface.

For the sake of brevity, in the following only the data model layer will be sketched. The data model must support temporal tables, in which both the *validity* time of data (i.e., the time when data are true) and its *transaction* time (i.e., the time when data are inserted\deleted from the DB) must be considered. Two levels of tables must be distinguished: (i) "supervisor" tables, containing data which have been validated (accepted) by supervisors, and "proposal" tables, concerning user proposals. Specifically, for each supervisor table r, we use three tables: (i) a table *propose_insert(r)* with the proposal of insertions in r, (ii) a table *propose_delete(r)* with the proposals of deletion, and (iii) a table *propose_update(r)* with the proposals of updates. While the first two tables contain standard tuples (plus their temporal part), *propose_update(r)* will contain pairs of tuples, the former representing the tuple to be modified, and the latter the new tuple to be entered.

Different manipulation operations are provided. Specifically, standard users can *propose* the *insertion, deletion* or *update* of tuples. Such operations will modify the proper proposal tables, while they will not affect the supervisor tables.

On the other hand, supervisors have the possibility to *accept* and to *reject* user proposals. The acceptance of a proposal will affect the corresponding supervisor table. For instance, the acceptance of a proposal of update will lead to the practical execution of the update on the supervisor table.

Subtle semantic issues need to be faced in the definition of operations. For instance, proposals of update concerning the same supervisor tuple must be managed as alternative proposals, in the sense that only one of them can be accepted (to resperct such an intended semantics, the acceptance of an update must modify the transaction time of each alternative update proposal to state that such proposals are no longer valid).

Results

We have defined an extended approach to temporal DB in order to support cooperative updates to a guideline.

Such an approach supports different levels of users (supervisors vs simple users) and supply proper operations for them, as required in guideline applications.

Acknowledgements

This work has been partially supported by Koine Sistemi.

Address for correspondence

Alessio Bottrighi, DI, Univ. Piemonte Orientale,
Via Bellini 25\g 15100 Alessandria, Italy
Email: alessio@mfn.unipmn.it
Ph: +39 0131 360338