1. Introduction

Modern database schemata are very large and complex. Even the most modest often have hundreds of relations; the largest have thousands. Fortunately, it is often possible to model even the largest schemata as an interconnection of simple *components*. The natural model for a component is a *view* — a "window" schema which provides partial information about the database.

The research proposed here involves two projects on database systems, each involving components, modelled as views, in an essential way. The first addresses key conceptual and theoretical issues, while the second develops a key application of these ideas. Both are in collaboration with colleagues in the Information Systems Engineering group at the University of Kiel, Germany.

2. Two Research Projects on Updates to Database Components

2.1 Foundations of Update Support on Database Components

This research is a joint effort with Bernhard Thalheim. It brings together his ideas on database components [Tha03], [ST04], [Tha05] with those of the author on the foundations of view updates [Heg04], [Heg06a], [Heg06b]. In the component-based approach to modelling, large database schemata are regarded as being composed of a *components*, which are interconnected via *ports*. Components themselves may be combined to form larger components; the smallest components are termed *atomic*.

The focus of the proposed research is the development of systematic principles which govern *updates* to components; that is, changes to the state of the database which is embodied in a given component. The major difficulty in supporting updates to any sort of view is that the mapping from the main schema to the view schema is not injective. Therefore, there are many possibilities for *reflecting* the view update to the main schema. Most of these alternatives suffer from so-called *anomalies*, in that they either depend upon or make changes to a part of the state of the main schema which is not visible in the view. As argued in [Heg04], an update strategy is free of such anomalies precisely in the case that it is based upon the classical *constant-complement strategy* of Bancilhon and Spyratos [BS81]. In the proposed research, the constant-complement strategy will be used as the standard model for anomaly-free updates to a view. In this context, the chief issue is not one of uniqueness of a suitable complements, as addressed in the previous work of the author, since a natural complement always

exists — that formed by all components which are not contained in the component to be updated. Rather, the main problem is that it may not be possible to support the requested update via constant complement while working only within the given component. It may thus be necessary to expand the environment to a larger component to support the given update in an anomaly-free fashion.

Initial results on this problem are reported in [HT06]. First of all, it is shown that *pairwise definability* [Heg93, 2.1.1] is sufficient to guarantee that the resulting formal framework of components will support update via constant complement. Second, the notions of *universal insertion* and *co-universal deletion* are introduced, and it is shown that within the abstract order-based update framework developed in [Heg04], these provide precisely the means to determine how to expand a component to support a given update. Third, it is shown that logical frameworks whose constraints are Horn-like [Fag82] provide a natural concrete framework for these constructs. The proposed research will build upon these initial results in the following ways.

Coverage of general updates: The existing work applies only to insertions and deletions, and not to more general updates which involve both. Although these special cases are important in themselves, it is crucial that the work be expanded to cover the general case.

Coverage of data models with limited disjunction: Essentially, Horn clauses are logical sentences which do not admit positive disjunction. Although their utility in computer science is unquestionable [Mak87], it is important that this work address situations involving other types of constraints. Work on incorporating key aspects of *HERM* [Tha00] will be pursued in particular.

2.2 View-Based Modelling of Dialog Flow in Web-Based Information Systems

This research is being conducted jointly with Peggy Schmidt.

There are two distinct types of information flow which occur in interactive information systems, particularly Web-based information systems. Firstly, for each user-task pair there is a *dialog flow* which captures that which the user sees in the process of carrying out that task. Second, for each task there are one or more *document flows*, which describe how data objects are modified, by both users and by autonomous agents, during the execution of the task. Generally, a document flow involves multiple users and a dialog flow involves multiple documents.

The overall goal of the proposed research is to develop a formalism which models both forms of

flow explicitly. While approaches which separate these two have been developed [ADDW03] [RD03], they are focused upon process logics which are specifically oriented towards commerce, rather than upon database access issues for general applications. Classical models of *workflow* [AH02], of which document flow is a special case, address the formal aspects of dynamics, but not how database access rights define those dynamics.

Since a document flow in a Web-based information system consists of a sequence of updates to the underlying database, it is natural to seek an update-based model. The research on updates to components described in the previous section will be used as a conceptual and theoretical foundation. The key modelling idea is to associate a component Γ_Δ to each dialog flow Δ, so that executing a task T within Δ is tantamount to performing an update on Γ_Δ.

In general, to support this update, it will be necessary to extend the environment to a subsuming component Γ'. This update to Γ' may require privileges which the user associated with Δ does not possess. The task is then to assemble a group of users who collectively have the privileges to perform the update associated with T. Formally, this is equivalent to meeting the preconditions for firing an event in a more classical model of workflow based upon, for example, Petri nets. However, the model which is proposed here is a much more natural one for a designer of a Web-based information system to employ, since it is tied directly to the associated databases.

The specific steps to be followed in carrying out this research are as follows.

Development of a formal notion of mapping access privileges to component-based views: The constant-complement update model provides a clear and formal notion of which updates are allowed to a given component view. However, this work requires that the converse idea be developed — a means of constructing a view based upon specified access privileges. The development of this notion will be the first step in the proposed work.

Development of formal associations to classical models: It will be shown formally how the view-based model which will be developed can be mapped to more conventional representations, such as those based upon Petri nets [AH02], [vdA01].

Integration of the formalism into design languages: Since a goal of this research is to provide a formalism which may be used for design, the resulting formalism will be integrated into existing design tools, such as SiteLang [TD01] and its descendants [ST05].

References

[AH02] van der Aalst, W. and van Hee, K., *Workflow Management: Models, Methods, and Systems*, MIT Press, 2002.

[vdA01] van der Aalst, W. M. P., "How to handle dynamic change and capture management information? An approach based on generic workflow models.," *Int. J. Comput. Syst. Sci. Eng.*, **16**(2001), pp. 295–318.

[ADDW03] Aberer, K., Datta, A., Despotovic, Z., and Wombacher, A., "Separating business process from user interaction in Web-based information systems," *Electronic Commerce Research*, **3**(2003), pp. 83–111.

[BS81] Bancilhon, F. and Spyratos, N., "Update semantics of relational views," *ACM Trans. Database Systems*, **6**(1981), pp. 557–575.

[Fag82] Fagin, R., "Horn clauses and database dependencies," *J. Assoc. Comp. Mach.*, **29**(1982), pp. 952–985.

[Heg93] Hegner, S. J., "Characterization of desirable properties of general database decompositions," *Ann. Math. Art. Intell.*, **7**(1993), pp. 129–195.

[Heg04] Hegner, S. J., "An order-based theory of updates for database views," *Ann. Math. Art. Intell.*, **40**(2004), pp. 63–125.

[Heg06a] Hegner, S. J., "The complexity of embedded axiomatization for a class of closed database views," *Ann. Math. Art. Intell.*, **46**(2006), pp. 38–97.

[Heg06b] Hegner, S. J., "Uniqueness of constant-complement update strategies for database views," 2006, submitted for journal publication.

[HT06] Hegner, S. J. and Thalheim, B., "Canonical expansion of database components for update support," 2006, submitted to ICDT 2007, the Eleventh International Conference on Database Theory.

[Mak87] Makowsky, J. A., "Why Horn formulas matter in computer science: Initial structures and generic examples," *J. Comput. System Sci.*, **34**(1987), pp. 266–292.

[RD03] Rodríguez, J. J. and Díaz, O., "Seamless integration of inquiry and transactional tasks in web applications," in: Meersman, R., Aberer, K., and Dillon, T. S., eds., *Semantic Issues in E-Commerce Systems, IFIP TC2/WG2.6 Ninth Working Conference on Database Semantics, April 25-28, 2001, Hong Kong*, pp. 105–119, Kluwer, 2003.

[ST05] Schewe, K.-D. and Thalheim, B., "Conceptual modelling of web information systems.," *Data Knowl. Eng.*, **54**(2005), pp. 147–188.

[ST04] Schmidt, P. and Thalheim, B., "Component-based modeling of huge databases," in: Benczúr, A., Demetrovics, J., and Gottlob, G., eds., *Advances in Databases and Information Systems: 8th East European Conference, ADBIS 2004, Budapest, Hungary, September 22-25, 2004. Proceedings*, pp. 113–128, Springer-Verlag, 2004.

[Tha00] Thalheim, B., *Entity-Relationship Modeling*, Springer-Verlag, 2000.

[Tha03] Thalheim, B., "Database component ware," in: *Database Technologies 2003, Proceedings of the 14th Australasian Database Conference, ADC 2003, Adelaide, South Australia, February 2003*, pp. 13–26, Australian Computer Society, 2003.

[Tha05] Thalheim, B., "Component development and construction for database design," *Data Knowl. Eng.*, **54**(2005), pp. 77–95.

[TD01] Thalheim, B. and Düsterhöft, A., "SiteLang: Conceptual modeling of Internet sites," in: *Conceptual Modeling - ER 2001, 20th International Conference on Conceptual Modeling, Yokohama, Japan, November 27-30, 2001, Proceedings*, pp. 179–192, Springer, 2001.