1. Introduction to the Problem

Updating a database through a user view, or window, is known to be a difficult problem at best. In general, there are many possible updates to the main schema which map to a single update on the view; the problem is then to choose which of those possible updates on the main schema, if any, is appropriate. The gold standard is the constant-complement strategy [BS81], which is precisely the approach which avoids all *update anomalies* [Heg04], *i.e.*, changes to the main schema which are not visible in the view. However, the constant-complement strategy is rather conservative with respect to the view updates which it allows, and has therefore been regarded as too restrictive by some [Lan90]. Extensions to the constant-complement strategy are in general rather *ad hoc*, and provide limited extension at best.

Even if one accepts that view update strategies which are more liberal than the constant-complement approach are to be allowed, the further issue of access rights must be addressed. Views are often used to limit access rights. Thus, if a view update is reflected to a change in the state of the main schema which is not entirely visible within the view, it is quite likely that the user of the view does not have the necessary privileges to make such an update, and so it must be disallowed.

In the proposed research, a new way of supporting updates to views will be investigated — *update by cooperation*. If a user of a view wishes to perform an update which does not respect the constant-complement strategy (that is, without side effects), then that user must enlist the cooperation of users of other views to complete the request. The agreed-upon update on the combined views of these users must respect the constant-complement strategy.

2. Completed Work

Update by cooperation requires a new way of modelling database schemata in which the users of distinct views can cooperate in a meaningful way. *Database components* [ST04] [Tha05] provide such a framework. Each component models the view of some user; the components may be interconnected via typed *channels*. The main schema is then modelled as such an interconnection. If a proposed update to some component does not satisfy the constant-complement strategy, then information about the additional updates which must be applied is communicated to neighboring components via the channels. Such information is then propagated further, as necessary. In a solution, users of all affected components must agree upon an update.

The database components of Thalheim [Tha05] are based upon the software components of Broy [Bro07], and are oriented primarily towards schema design. In particular, they combine the definition of the interconnected schema with the communication model. For the cooperative update problem, a model which separates these two is essential. To this end, an alternative model of database component, based only upon the core notions of schema and view, has been developed by the author [Heg07b]. This model is ideally suited for the cooperative update problem, and initial results in this direction have already been obtained [HS07].

A special case of the cooperative update problem occurs when the updates to neighboring components may be realized *canonically*; that is, when there exists a best (up to isomorphism) update to those components. In this case, the process of cooperation requires only approval, and not other choices, on the part of the neighboring components. This problem has been investigated by the author [Heg07a] and will be submitted to major conference during the summer of 2007.

3. The Proposed Research

The proposed research will be carried out in collaboration with Peggy Schmidt and Bernhard Thalheim of the Information Systems Engineering Group at Christian-Albrechts-Universität zu Kiel, Germany. It will build upon the foundational results reported in [Heg07b], [Heg07a], and [HS07] in the following ways.

Expanded models of cooperation and collaboration: The model of cooperative update which is presented in [HS07] is a first-step, proof-of-concept effort. To render it practical, this initial effort must be expanded substantially. The focus will be upon the following points.

Support for non-monotonic negotiation: In [HS07], the negotiation process is monotonic, in the sense that an update request, as it is propagated through the components, can only be refined. There is no facility which allows an actor to present a true counterproposal which conflicts with aspects of the existing proposal, and for the initial proposer then to agree to a modification. Needless to say, such capabilities are essential for real-world negotiations. In the proposed research, the model will be extended to support such extended, non-monotonic negotiation.

Actor-controlled models of communication: In [HS07], the model of task flow is completely opportunistic. Individual actors can only respond to update proposals by proposing themselves how such updates lift to their views; it is not possible for an actor to exert finer-grained control over the negotiation process, for example by first negotiating with Party A and then with Party B. This makes it difficult to model situations in which certain actors must make decisions not only about what is to be allowed, but also about how the negotiation process is to proceed. In the proposed research, a more extensive model of communication which allows actors to make decisions about the flow of negotiations will be developed.

The connection between of cooperation and workflow: The classical topic of *workflow* involves the systematic modelling of processes which require the coordinated interaction of several actors [AH02]. It has long been known that such models play an important rôle in the context of information systems [FGHW88] [AAA*96] [RS95]. The model of cooperative update clearly involves a workflow, as the negotiation process proceeds and requests are passed from actor to actor. In general, a cooperative update request will not involve a fixed workflow, but rather will impose constraints on the family of workflows which will lead to its realization. The goal of this part of the proposed research is to make this association explicit, as a theory of *query-based workflow construction*.

The natural scope of general updates: The theory of canonical cooperative updates which is reported in [Heg07a] is restricted to insertions and deletions. In that context, it is possible to order such updates by inclusion, thus providing the necessary notions of minimality. For more general types of updates, obtaining a satisfactory notion of minimality is much more challenging. Although general notions of minimal update have been forwarded, largely for updates to logic databases [MT99], they have never proven to be satisfactory.

The method of defining canonical updates in [Heg07a] uses a universal construction (in the spirit of category theory [HS73]) to give a precise meaning to the notion of canonical. The great utility of category theory is that when it is known how to realize a concept in familiar territory (in this case insertions and deletions) using universal constructions, then that same construction may often be applied to unfamiliar territory (in this case general updates) to obtain the needed generalization. This is precisely the approach which will be taken in the proposed work.

References

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

[AAA*96] Gustavo Alonso, Divyakant Agrawal, Amr El Abbadi, Mohan Kamath, Roger Günthör, and C. Mohan, "Advanced transaction models in workflow contexts," in: Stanley Y. W. Su, ed., *Proceedings of the Twelfth International Conference on Data Engineering, February 26 - March 1, 1996, New Orleans, Louisiana*, pp. 574–581, IEEE Computer Society, 1996.

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

[Bro07] Manfred Broy, "Model-driven architecture-centric engineering of (embedded) software intensive systems: modeling theories and architectural milestones," *Innovations Syst. Softw. Eng.*, **3**(2007), pp. 75–102.

[FGHW88] Fernando Flores, Michael Graves, Brad Hartfield, and Terry Winograd, "Computer systems and the design of organizational interaction.," *ACM Trans. Inf. Syst.*, **6**(1988), pp. 153–172.

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

[Heg07a] Stephen J. Hegner, "Charactertizing the natural scope of updates to database components via free extensions," Technical Report, Christian-Albrechts-Universität zu Kiel, 2007.

[Heg07b] Stephen J. Hegner, "A model of database components and their interconnection based upon communicating views," in: Hannu Jakkola, Yashushi Kiyoki, and Takehiro Tokuda, eds., *Information Modelling and Knowledge Systems XXIV*, Frontiers in Artificial Intelligence and Applications, IOS Press, 2007, in press.

[HS07] Stephen J. Hegner and Peggy Schmidt, "Update support for database views via cooperation," in: Yannis Ioannis and Boris Novikov, eds., *Advances in Database and Information Systems, 11th Eastern European Conference, ADBIS 2007 Varna, Bulgaria, September 29 - October 3, 2007, Proceedings*, Lecture Notes in Computer Science, Springer-Verlag, 2007, in press.

[HS73] Horst Herrlich and George. E. Strecker, *Category Theory*, Allyn and Bacon, 1973.

[Lan90] Rom Langerak, "View updates in relational databases with an independent scheme," *ACM Trans. Database Systems*, **15**(1990), pp. 40–66.

[MT99] Enric Mayol and Ernest Teniente, "A survey of current methods for integrity constraint maintenance and view updating," in: *Proc. ER '99 Workshops, Paris, Nov. 15-18, 1999*, Volume 1727 of Springer LNCS, Springer-Verlag, 1999.

[RS95] Marek Rusinkiewicz and Amit P. Sheth, "Specification and execution of transactional workflows," in: Won Kim, ed., *Modern Database Systems*, pp. 592–620, ACM Press and Addison-Wesley, 1995.

[ST04] Peggy Schmidt and Bernhard Thalheim, "Component-based modeling of huge databases," in: András Benczúr, János Demetrovics, and Georg Gottlob, eds., *Advances in Databases and Information Systems: 8th East European Conference, ADBIS 2004, Budapest, Hungary, September 22-25, 2004, Proceedings*, pp. 113–128, Volume 3255 of Lecture Notes in Computer Science, Springer-Verlag, 2004.

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