1. Specific Goals of the Project

Two essential features of any large, modern database management system are the capability of performing updates and the capability of delivering limited access through views. The integration of these two aspects — the capability of supporting updates which are specified in views — has long been known to be a very difficult problem. The major problem is not one of existence; it is always possible to reflect an update to a view schema back to the main schema. Rather, the key issue is one of suitability; the reflection of the view update back into the base schema must be acceptable according to certain criteria.

For the most part, previous research on this problem has sought to develop strategies whereby a maximal set of view updates is allowable, with little attention paid to so-called *interface criteria* which impact the amount of knowledge which the user must possess in order to effect and understand a view update. Specifically, most of these existing strategies require that the user have knowledge of parts of the schema, and even of the data, which lie outside of the view. The research proposed herein, on the other hand, keeps central the idea that a view is to be an entity unto itself, and should appear to the user to be no different than a full database system. Such a view presentation is particularly important in the case that the user is not a database expert.

The goal of this project is to build upon the previous work of the author on methods of update support for views which emphasize interface criteria, the so-called *closed views,* by studying the complexity issues surrounding the support of updates via such views. More specifically, circumstances under which the management of view updates is computationally feasible, relative to the complexity of updating the base schema, will be investigated.

2. Overview of the Research Area

Open vs. closed views The topic of database updates via views has been the subject of investigation for at least a quarter century [DB78]. While some of the more recent work has moved from the classical relational model to other models, such as entity-relationship [LL92] and object-oriented [Bel00] models, virtually all investigations have been conducted within the context of open views.

The distinction between open and closed views was introduced in [Heg90a], and further elaborated in [Heg02] and [Heg03]. Roughly speaking, an *open view* admits the widest possible family of update operations, but requires that the user have knowledge of aspects of the overall schema and/or database which lie outside of the view. In general, open views involve anomalies such as *hidden triggers*, *hidden dynamic constraints,* and *irreversibility* [Heg03, 1.2].

A *closed view,* on the other hand, is functionally indistinguishable from the full schema of a database system. Roughly speaking, there are two principles which govern a closed strategy. First of all, the admissibility of an update to the view must depend only upon the state of the view; it must never be necessary to have further information about the state of the base schema in order to determine whether an update is to be allowed. Second, the family of allowed updates must form an equivalence relation; that is, it must be closed under reversibility, transitivity, and of course identity. These ideas are elaborated more fully in [Heg03, Sec. 1].

The constant-complement approach The seminal work on closed update strategies is the *constant-*

complement approach, developed more than twenty years ago by Bancilhon and Spyratos [BS81].[1] The idea is quite simple. To support updates to the view Γ of the main schema $\mathbf{D}$, a view Γ' which is complementary to Γ is identified; *i.e.*, $\{\Gamma, \Gamma'\}$ forms a lossless decomposition of $\mathbf{D}$. This decomposition may be very general and need not be based upon a join, even in the case that the schemata are relational. Then, the only updates to Γ which are allowed are those which hold the state of Γ' fixed. Intuitively, the changes are isolated in Γ; the rest of the schema, which is Γ', cannot change.

Fully commuting pairs It is a remarkable fact that every closed update strategy is definable via a constant-complement strategy [BS81, Thm. 7.3]. However, the converse is not the case. An explicit characterization of conditions under which the converse does hold was not pursued in the original work of Bancilhon and Spyratos. Rather, it was first identified in [Heg90a, 2.10], albeit without proof. The key condition for this converse characterization is that of a *fully commuting pair* [Heg03, 3.10]; that is, a pair of views whose underlying congruences commute. In the context of simple projections on a relational schema decomposed by a join dependency and further constrained by a set of full dependencies, this conditions is equivalent to the decomposition being both lossless and dependency preserving [Heg03, 2.17]. An example of a complement which does not yield an update strategy is provided by any relational decomposition into two projections which is lossless but not dependency preserving. This property of commuting congruences is also key to the study of a number of questions in database theory, and was thoroughly investigated in [Heg91] and [Heg93], where it is shown to be central in the generalization of the classical notion of acyclic join decomposition.

Uniqueness of update strategies in the context of order-based views Despite the title of the seminal paper [BS81], the theory of updates described there, and furthered in the work of the author, is not in any way tied to the relational model. Rather, it is as general as possible — database schemata are just sets (of legal databases) and database mappings (which define views) are just functions. Unfortunately, the original theory is a bit too general. In order to fix the way in which the view updates are reflected back to the main schema, it is necessary to choose a complement to be held constant. Except in the most trivial of cases, this choice is never unique [BS81, Thm. 4.4]; a complement must be selected in some way in order to fix the strategy by which view updates are reflected back to the main schema. This runs contrary to the natural intuition in common data models and their views, such as the relational model with views defined by projection, in which there is only one "natural" complement.

Past and current work of the author has addressed this uniqueness issue. In the early paper [Heg94], it is shown that by endowing the states of the database schema with a natural ordering (defined by relation-by-relation inclusion in the relational model), and by requiring that database mappings respect this order, it is possible guarantee that complements are unique. However, that paper was concerned with *direct* decompositions into views which are totally independent of one another, and thus its results are not immediately applicable to the project at hand, since complements for view update purposes are typically *subdirect*; that is, there is common knowledge shared by the views. Nonetheless, the idea of using the natural order on the states of a database schema is an appropriate one. In recent papers, it is shown that under such ordering constraints, while the complement itself may not be unique, all view updates which are *order based*; that is, which may be realized by a se-

[1] The terminology *closed update strategy* is due to the author, and is formalized in [Heg02, 3.1] and [Heg03, 3.1]. However, the formulation of a *translator* in [BS81, Def. 3.3], although outwardly different, is formally equivalent.

quence of insertions and deletions, have a unique reflection as updates into the main schema [Heg02, 4.2], [Heg03, 4.3]. In other words, this reflection is independent of the choice of complement. In the case that the set of admissible updates is sufficiently rich, the choice of complement itself is unique [Heg02, 4.3], [Heg03, 4.4]. In other words, under these order-based restrictions, which are met in practice by such ubiquitous examples as *select-project-join* (*SPJ*-) mappings in the relational setting, updates to closed views have a unique reflection to an update in the main schema.

Complexity results The research program described herein proposes to build upon the results developed in [Heg02] and [Heg03] with a systematic examination of complexity and algorithmic issues. Therefore, it is appropriate to provide an overview of previous research results which are relevant to this investigation. The key question in the complexity of updates is the so-called *maintenance problem* [WG92]. That is, given a state M of the schema which is known to satisfy the extant integrity constraints, and a new potential state M' which differs from M only slightly (say, by the insertion of a tuple), determine whether or not M' is also satisfies the extant constraints. Of course, this test may be conducted directly (by using, say, inference mechanisms such as the well-known chase procedure), but it is generally more efficient to employ algorithms which make use of the fact that M is known to satisfy the constraints. The only work which explicitly addresses such issues in the context of the constant-complement strategy is [CP84]. Working with views on relational schemata defined by projections and constrained by functional dependencies and join dependency for decomposition, the characterization of the computational complexity of deciding whether a proposed insertion of a tuple to the view is realizable at all while holding the complement constant is addressed. The principal results identify a very high worst-case complexity. However, at the time that [CP84] was written, the connection of the constant-complement strategy to fully commuting pairs, as described above, was not known. The examples of complementary pairs used in the proofs are not fully commuting, and so the results are not relevant to true constant-complement strategies as described in [BS81] and employed in extended fashion in the research program described here.

Unfortunately, in the more general context of the complexity of updates, most current work is focused upon logic databases, which are not central to the scope of this work.

3. Description of the Project

Overview of the research program The overall goal of the research program is to investigate the issue of relative update complexity for closed views. The program will be divided into four steps, as follows.

Step 1: Investigation of the problem in the context of the traditional relational model, with simple SPJ-views.

Step 2: Extension of the results of Step 1 to the case in which foreign-key dependencies are also imposed.

Step 3: Investigation of the problem in the general context of order-based views, as developed in the previous work of the author [Heg02], [Heg03].

Step 4: Application of the results of Steps 1 and 2 to a more general data model: the Higher-Order Entity-Relational Model (HERM) [Tha00].

Steps 3 will be undertaken in parallel with Steps 1 and 2, while Step 4 will be taken after the others are well underway. The reason for initiating this investigation with parallel steps is that there does not exist as sufficiently rich body of knowledge about the relative complexity of updates on closed relational views to admit a pure investigation of the general case. On the other hand, it is anticipated that the results of Steps 1 and 2 will feed the development of Step 3.

Elaboration of Step 1 In the context of the relational model, the complexity of an update will be measured in the classical way; that is, via the complexity of determining whether a proposed new state satisfies the extant integrity constraints, given that the current state does. Although most recent work on consistency enforcement has focused upon formal specification in a programming-model environment (*e.g.*, [ST99], that approach is not a suitable fit for the questions at hand, since the principal interest is checking the suitability of a proposed update rather than considering how a given update might be modified to render it appropriate.

The classical *full implicational dependencies* [Fag82] , that is, dependencies which are expressed as Horn sentences with universal quantifiers, will be the initial focus. The specific measure of complexity of a set $\mathcal{F}$ of dependencies will be its *rank* [Hul84, Sec. 3]. Formally, the rank of a dependency is the number of distinct (universally quantified) variables which occur in it. Operationally, if $\mathcal{F}$ is a set of dependencies each of rank at most n, then the question of whether a given database M satisfies $\mathcal{F}$ may be answered by examining subsets of at most n tuples.[2]

Now let $\{\Gamma_1, \Gamma_2\}$ be a fully commuting pair of views of the base schema $\mathbf{D}$. Think of $\Gamma_1 = (\mathbf{V}_1, \gamma_1 : \mathbf{D} \to \mathbf{V}_1)$ as the view to be updated, and $\Gamma_2 = (\mathbf{V}_2, \gamma_2 : \mathbf{D} \to \mathbf{V}_2)$ as the complement to be held constant. The condition to be characterized is that of *relative update complexity*; that is, it should be no more complex to verify that a potential update to $\mathbf{V}_1$ satisfies the integrity constraints on $\mathbf{V}_1$ than it is to check that a potential update to $\mathbf{D}$ satisfies the integrity constraints on that schema. To formalize this notion, it is first necessary to recall that $\{\Gamma_1, \Gamma_2\}$ gives rise to a so-called *meet* view $\Gamma_1 \wedge \Gamma_2$ which recaptures the common information. $\Gamma_1 \wedge \Gamma_2$ may be regarded not only as a view of the main schema $\mathbf{D}$, but also of each of $\mathbf{V}_1$ and $\mathbf{V}_2$. For Γ_1 and Γ_2 projections of the same relation, the view $\Gamma_1 \wedge \Gamma_2$ is the projection on the common columns.

This meet is extremely important because when updating the view Γ_1 with constant complement Γ_2, it is precisely the updates which hold the state of $\Gamma_1 \wedge \Gamma_2$ constant which are allowed [Heg03, 3.9]. From an axiomatic (*i.e.*, integrity constraint) point of view, it is thus not necessary to axiomatize all of the view Γ_1; rather, it suffices to axiomatize it *relative to* $\Gamma_1 \wedge \Gamma_2$ [Heg90a, 3.5]. Call the view Γ_1 *relatively update tractable* with respect to Γ_2 if there is an axiomatization of $\mathbf{V}_1$, relative to $\Gamma_1 \wedge \Gamma_2$, which has rank no larger than that of the main schema $\mathbf{D}$. The main problem to be studied in this step of the research program may be stated as follows:

> **Problem 1:** For a given relational schema $\mathbf{D}$, characterize those complementary pairs $\{\Gamma_1, \Gamma_2\}$ for which Γ_1 is relatively update tractable with respect to Γ_2.

It is important to note that it does not suffice to consider measuring relative update complexity on an arbitrary view Γ_1 with respect to an empty meet. Indeed, it is well known that there exists a single-relation schema, constrained only by functional dependencies (and thus of rank two) which has a projection which is not of finite rank [Hul84, Lem. 4.1], [Heg90b]. There is, on the other hand, substantial evidence to suggest that such anomalies cannot occur when considering fully commuting

[2] Actually, for so-called *tuple-generating dependencies*, a check for an additional tuple in the database will need to be made, but this does not alter signifi cantly the overall notion of the complexity measure.

pairs. Indeed, for a pair of projections of a single relation constrained by full dependencies to be fully commuting, it must be both lossless and dependency preserving [Heg03, 2.17]. The pessimistic complexity results reported in [CP84] failed to take this into account, and thus do not represent the situation with regard to constant-complement update strategies as characterized in [BS81].

Although the above elaboration does not explicitly address the maintenance problem (as distinguished from the constraint satisfaction problem), because the characterization is of the relative complexity of updates on the main schema to updates on the view schema, it appears that any savings in computation will scale as well. However, this issue will need to be examined more closely as the research progresses.

Elaboration of Step 2 Foreign-key constraints are a central component of the SQL data-definition language, and so are supported by virtually all modern relational database systems. Therefore, it is crucial that they be incorporated in the current study. In this step of the research program, the results of Step 1 will be extended to encompass inclusion dependencies.

> **Problem 2:** Extend the results obtained in the solution of of Problem 1 to those cases in which the schemata may also be constrained by inclusion dependencies.

It is well known that the combination of functional dependencies and inclusion dependencies can lead to undecidable problems. While these decision problems do not directly impact the questions investigations proposed here (even in the context of undecidability for satisfiability, finite model checking is algorithmic), they do have the potential to contribute greatly to the inherent complexity. Fortunately, in the context of suitable yet practical restrictions, a much more accommodating theory may be obtained [LL99], [LL01], and it is within such contexts that the investigation will be conducted.

Elaboration of Step 3 In Step 3, the same questions will be pursued as in Steps 1 and 2, but in the context of a general framework rather than the specific relational model. In order to facilitate this investigation, the framework employed in [Heg02] and [Heg03] must be refined somewhat. In those references, a database schema $\mathbf{D}$ is characterized by its set of legal states $\mathsf{LDB}(\mathbf{D})$, together with an underlying partial order $\leq_{\mathbf{D}}$. The morphism γ of a view $\Gamma = (\mathbf{V},\gamma)$ is a function $\gamma : \mathsf{LDB}(\mathbf{D}) \to \mathsf{LDB}(\mathbf{V})$ which is an open surjection.

To provide an appropriate framework for the investigation of update complexity, this context must be augmented. First of all, to each schema $\mathbf{D}$ is now associated two sets of states, the set $\mathsf{DB}(\mathbf{D})$ of all states, and a subset $\mathsf{LDB}(\mathbf{D}) \subseteq \mathsf{DB}(\mathbf{D})$ of the *legal* states; *i.e.*, those which satisfy the integrity constraints. A view morphism γ for $\Gamma = (\mathbf{V},\gamma)$ now maps $\gamma : \mathsf{DB}(\mathbf{D}) \to \mathsf{DB}(\mathbf{D})$ with $\gamma(\mathsf{LDB}(\mathbf{D})) = \mathsf{LDB}(\mathbf{V})$. Furthermore, $\mathsf{DB}(\mathbf{D})$ is taken not only to be a partially ordered set, but a subset of the powerset $\mathbf{2}^X$ of some base set X. This generalizes the case of a single relation in the obvious way, with X the set of all possible tuples for that relation.

Now, it is a straightforward matter to generalize the notion of rank, at least for the case corresponding to equality-generating dependencies; it is only necessary to adapt the condition (*) of [Hul84, Sec. 3]. More specifically, a database schema has *constraint rank n* if for each $M \in \mathsf{DB}(\mathbf{D})$ and each $N \subseteq M$ with $\mathsf{Card}(N) \leq n$ (here $\mathsf{Card}(-)$ denotes cardinality), if there is a $P \in \mathsf{LDB}(\mathbf{D})$ with $N \subseteq P \subseteq M$, then $M \in \mathsf{LDB}(\mathbf{D})$.

The above notion generalizes all full dependencies in the relational case, including the so-called tuple-generating dependencies, such as join dependencies. It is also interesting to restrict attention to the situation corresponding to the so-called equality-generating dependencies, such as functional dependencies. In that case, the constraints only restrict the fields of existing tuples; they do not specify the existence of any new ones. Define a database schema to have *pure constraint rank n* if for each $M \in \mathsf{DB}(\mathbf{D})$, $M \in \mathsf{LDB}(\mathbf{D})$ whenever $N \in \mathsf{LDB}(\mathbf{D})$ for each subset $N \subseteq M$ with $\mathsf{Card}(N) \leq n$. Now, proceeding in a fashion similar to that described in the paragraph elaborating Step 1A above, the notions of Γ_1 being *relatively update tractable* (resp. *purely relatively update tractable*) with respect to Γ_2 for a fully commuting pair $\{\Gamma_1, \Gamma_2\}$ may be defined. The corresponding main problems to be studied in this step of the research program may be stated as follows:

Problem 3: For a given abstract schema **D** as defined above, characterize those complementary pairs $\{\Gamma_1, \Gamma_2\}$ for which Γ_1 is relatively update tractable and/or purely relatively update tractable with respect to Γ_2.

Elaboration of Step 4 Within Steps 1 through 3, the overall plan is to use results developed for the classical relational model (Steps 1 and 2) as the basis for a more general approach which is independent of any particular data model (Step 3). Step 4, the final step of this proposed research program, is designed to evaluate the utility of the results of Step 3 by applying them to a more general data model, HERM [Tha00].

Problem 4: Examine the way in which the results of Step 3 may be applied to HERM.

HERM is an attractive option because it attempts to recapture many of the dimensions of real-world data modelling for which the classical relational model is not well suited, while at the same time remaining independent of any particular inclination towards a particular model. Furthermore, HERM has an extensive, well-developed theory which should lend itself well to the envisioned theoretical results of Step 3.

4. Equipment

The research proposed herein will not require any special equipment beyond that which is normally available to the investigator at the Department of Computing Science at Umeå University.

5. International Cooperation

There is no specific international collaboration planned for this project. However, the principal investigator regularly participates in the FoIKS (Foundations of Information and Knowledge Systems) symposium series, both as a presenter and as a member of the program committee. It is anticipated that there will be substantial contact with the members of that group throughout the duration of the project.

6. Preliminary Results

The author has not addressed the complexity questions described here in any previous work However, he has carried out much previous research on the theory of views. In particular, the reference [Heg03] develops in great detail the foundations of updates to closed views upon which this entire research project is based.

7. Significance of the Proposed Research

While the research itself is of a theoretical nature, its relevance is most likely to be felt in the area of user-interface design for database systems. In many cases, support for updates via views is supported only in an *ad hoc* fashion, often requiring a substantial amount of expertise in database systems on the part of the user. The notion of a closed view is designed to overcome this limitation by providing a systematic, encapsulated way to view updates through views. It should thus make it easier to design user interfaces to database systems whereby users without computer expertise may work with a view interface which is simple and easy to understand, yet which supports a substantial family of updates.

8. Personnel in the Research Group

	Current position	Period of appointment	Working hours within research in total	All sources of funding for salary	Period of funding for salary	Personnel relevant to the project (marked with X in the column)
Main applicant	Senior lecturer and docent	permanent	50%	100%	permanent	X

9. Other Financing of this Research

No other external funding for this research has been sought. Currently, it is funded by the Faculty of Science and Technology of Umeå University.

References

All of the papers listed below which are by the author of this proposal are available in PDF format at the web site `http://www.cs.umu.se/~hegner`.

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

[Bel00] Bellahsène, Z., "Updates and object-generating views in ODBS," *Data & Knowledge Engr.*, **34**(2000), pp. 125–163.

[CP84] Cosmadakis, S. and Papadimitriou, C., "Updates of relational views," *J. Assoc. Comp. Mach.*, **31**(1984), pp. 742–760.

[DB78] Dayal, U. and Bernstein, P. A., "On the updatability of relational views," in: *Proceedings of the Fourth International Conference on Very Large Data Bases*, pp. 368–474, 1978.

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

[Heg90a] Hegner, S. J., "Foundations of canonical update support for closed database views," in: Abiteboul, S. and Kanellakis, P. C., eds., *ICDT'90, Third International Conference on Database Theory, Paris, France, December 1990*, pp. 422–436, Springer-Verlag, 1990.

[Heg90b] Hegner, S. J., "Some open problems on view axiomatization," *Bulletin of the EATCS*, **40**(1990), pp. 496–498.

[Heg91] Hegner, S. J., "Pairwise-definable subdirect decompositions of general database schemata," in: Thalheim, B., Demetrovics, J., and Gerhardt, H.-D., eds., *MFDBS91, Third Symposium on Mathematical Fundamentals of Database and Knowledge Base Systems, Rostock, Germany, May 1991*, pp. 243–257, Springer-Verlag, 1991.

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

[Heg94] Hegner, S. J., "Unique complements and decompositions of database schemata," *J. Comput. System Sci.*, **48**(1994), pp. 9–57.

[Heg02] Hegner, S. J., "Uniqueness of update strategies for database views," in: *Foundations of Information and Knowledge Systems: Second International Symposium, FoIKS 2002, Salzau Castle, Germany, February 2002, Proceedings*, pp. 230–249, Springer-Verlag, 2002.

[Heg03] Hegner, S. J., "An order-based theory of updates for database views," *Ann. Math. Art. Intell.*, **39**(2003), , in press.

[Hul84] Hull, R., "Finitely specifiable implicational dependency families," *J. Assoc. Comp. Mach.*, **31**(1984), pp. 210–226.

[LL99] Levene, M. and Loizou, G., "How to prevent interaction of functional and inclusion dependencies," *Info. Process. Lett.*, **71**(1999), pp. 115–125.

[LL01] Levene, M. and Loizou, G., "Guaranteeing no interaction between functional dependencies and tree-like inclusion dependencies," *Theoret. Comput. Sci.*, **254**(2001), pp. 683–690.

[LL92] Ling, T. W. and Lee, M.-L., "A theory for entity-relationship view updates," in: Pernul, G. and Tjoa, A. M., eds., *Entity-Relationship Approach - ER'92, 11th International Conference on the Entity-Relationship Approach, Karlsruhe, Germany, October 7-9, 1992, Proceedings*, pp. 262–279, Springer-Verlag, 1992.

[ST99] Schewe, K.-D. and Thalheim, B., "Towards a theory of consistency enforcement," *Acta Informatica*, **36**(1999), pp. 97–141.

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

[WG92] Wang, K. and Graham, M. H., "Constant-time maintainability: A generalization of independence," *ACM Trans. Database Systems*, **17**(1992), pp. 201–246.