Chapter 6

Summary and Conclusions

In this dissertation, I have described and evaluated a novel technology to facilitate the sharing of decision-support applications. Predicated on the belief that decision-support applications should access clinical databases directly and that clinical databases will remain heterogeneous in their organization and implementation, the TransFER technology facilitates sharing by *automatically translating database queries to a variety of equivalent formulations* so that decision-support applications can retrieve data directly from a variety of clinical databases. Although TransFER's query-translation methodology supports databases implemented in the *relational model* only, the methodology accommodates a large space of different relational database implementations. These implementations may vary in their relational representations of real-world objects, in their naming of tables and fields, in their coding of medical data, and in their dialects of relational query languages. The TransFER technology automatically resolves many of these heterogeneities so that developers can embed queries in decision-support applications without concerning themselves with the implementation details of existing clinical databases.

6.1 The Essence of TransFER

In Chapter 1, I described how heterogeneity among database implementations currently impedes the sharing of decision-support applications that retrieve data directly from clinical databases. The crux of the problem is that database queries embedded in decision-support applications must be customized at each site where the application is deployed, and this process is costly, time-consuming, and error-prone. In Chapter 2, I discussed previous research on standard data models and automated query-translation methods to reduce the costs and errors associated with customizing database queries. Although significant work has been done by distributed database developers, artificial intelligence researchers, and informatics-standards bodies, I concluded that no satisfactory integration of data-modeling and query-translation methods yet exists to support fully the sharing of decision-support applications.

In Chapters 3 and 4, I presented my own methodology for customizing database queries, called TransFER, which automatically translates queries based on the encoded specifications of various relational databases. The methodology consists of four formal components:

1. A semantic data model for representing the contents of clinical databases
2. A logic-based query language for encoding queries in a database-independent way
3. A database-mapping language based on a variation of the relational algebra
4. A query-translation algorithm based on the theory of attribute grammars

The data model and query language are sufficiently abstract that they allow developers of decision-support applications to encode queries in manner that *subsumes* the database schema and query language of any specific relational database implementation. The mapping language allows database experts at each site to encode declaratively the correspondence between the *global abstract schema* of clinical information (specified in the semantic data model) and the *local implemented schema* of clinical information (specified in the relational data model). The query-translation algorithm applies the encoded mappings at each site to automatically translate queries that appear in decision-support applications to equivalent relational queries that are specific to the database structure, clinical vocabulary, and query language in use at that site. The result of the translation is a database program consisting of SQL queries (which the local database engine can execute) and calls to arbitrary functions (which the local computing environment can process). The appropriate interleaving of SQL queries and function calls retrieves from the local database the clinical data requested by the decision-support application.

Although other methods exist for automatically translating database queries from a semantic data model to the relational data model, the TransFER methodology is novel in that it bases the query translation on *explicit* and *declarative* mappings between *individual constructs* of the semantic database schema and a relational database schema. The mapping of individual schema constructs in a declarative language simplifies the database-mapping process for human mappers by dividing the task into discrete modular steps and by supporting automated validation. The explicit specification of mappings facilitates the automated query-translation process by representing precisely which transformations are required to effect the translation, rather than requiring a compiler to search the space of possible transformations. The declarative representation of mappings in the extended relational algebra language supports the optimization of translated queries via equivalence-preserving transformations based on the formal properties of the relational algebra; I have defined and proven formally a set of such transformations. The explicit, declarative, and modular nature of the mappings allows queries that appear in decision-supported applications to be customized more easily, accurately, and efficiently than was possible with previous methods.

6.2 The Meaning of the Evaluation

In Chapter 5, I presented and discussed the results of an experiment I conducted to test the effectiveness of my methodology in practice by translating queries to three independently designed clinical databases. Three groups of volunteer subjects generated the relational databases based on a single clinical data set, and two physicians specified a set of representative queries with respect to the same clinical data set. The TransFER compiler subsequently translated the queries to three sets of database-specific programs based on mappings that I specified for each database implementation.

Given that the generated relational databases manifested significant heterogeneity, the experimental data was valid for evaluating the TransFER methodology. By assessing TransFER with respect to the five performance criteria that I defined in Section 5.2.4, I can now consider whether TransFER is superior to existing methods for sharing applications that access clinical databases. I address TransFER's performance with respect to each criterion in the following sections.

6.2.1 Latitude

The relational algebra mapping language handled heterogeneity in the *structured* representation of clinical data; this heterogeneity included table/field naming differences, coding differences, denormalization, and implicit versus explicit modeling. The mapping methodology could not accommodate certain unstructured representations, such as free-text fields that contained narrative clinical descriptions. However, no existing method for translating queries such that they can be executed automatically accommodates all narrative descriptions either. The latitude of TransFER is, therefore, equivalent to existing query-translation methods. When natural-language processing techniques become available to structure narrative clinical data, the extensibility of the TransFER mapping language via external-function calls will be able to exploit the technology to increase the latitute of the TransFER methodology.

6.2.2 Declarativeness

Less than 10 percent of the mappings across all of the schemas included external function calls. The Physician Workstation (PWS) environment, the only existing method that performs query translation automatically, requires procedural functions for every mapping. TransFER, therefore, provides significantly greater potential for query optimization than PWS because the declarative semantics of the relational mapping language support automated query transformation. I demonstrated this advantage by showing a 20 percent improvement in the efficiency of generated database programs when I implemented optimization rules based on the formal properties of the relational algebra.

6.2.3 Expressiveness

Although nearly a third of the queries generated by the physician experts could not be encoded completely in the ReFER query language, the missing query constructs were primarily aggregation and arithmetic-manipulation operators for temporal data (as opposed to data-retrieval operators). In practice, this deficiency is not a critical limitation because decision-support applications can perform aggregation or arithmetic manipulation of temporal data themselves after the TransFER interface has retrieved a superset of the required data. Furthermore, the standard SQL query language itself does not contain aggregation operators or arithmetic operators for temporal data; this implies that no query-translation method can generate SQL queries that capture the semantics of the information requests more fully than ReFER.

6.2.4 Correctness

The errors of correctness in the evaluation experiment resulted from missing data in the relational implementations or from faulty mapping specifications, rather than from any intrinsic flaws in the query-translation mechanism. If one assumes that the relational databases to which an FER schema is mapped contain all of the data represented in the FER schema and that the mappings for all entity types and relationship types in the FER schema are defined correctly, then the data from the experiment indicates that the TransFER methodology translates queries with 100 percent accuracy. This result confirms that the query-translation methodology, *per se*, is sound. Although, in practice, certain relational databases may not contain all of the information represented in the FER schema, these cases could be handled by extending the FER data model to include constructs that explicitly denote occurrences of missing data; this information would allow applications to respond to the absence of requested data in an appropriate manner (such as querying the user).

6.2.5 Efficiency

The data from the experiment clearly identify efficiency as the greatest current limitation of the TransFER methodology compared to existing manual query-translation methods. The four-fold difference in complexity between TransFER-generated database programs and manually generated database programs probably precludes the clinical use of TransFER at this time. However, the opportunity for optimization, both theoretical and realized, that TransFER's declarative mappings provide constitutes an improvement over any automated query-translation methods that currently exist. Further investigation and implementation of query-optimization techniques ultimately may make the efficiency of TransFER-generated query translations comparable to that of manually generated translations. Given that TransFER is already competitive with existing methods in terms of latitude, declarativeness, expressiveness, and correctness, significant advancements in efficiency may well establish TransFER as the preferred method for sharing decision-support applications that access clinical databases.

6.2.6 Generalizing the results

The experiment that I conducted provided an initial evaluation of the power and generality of the TransFER translation methodology. In designing the experiment, I took steps to create as realistic a set of problems as possible against which to test TransFER's capabilities. These steps included acquiring actual clinical data, transforming the clinical data to a representationally neutral form, taking care to not bias participants, and carefully ordering the sequence of data-modeling and query-formulation tasks. However, even accounting for these measures, the problems posed to the TransFER technology by this experiment were not *complete* with respect to the heterogeneity that exists among “real-world” clinical databases. To determine the meaning of the experiment and the generalizability of the results, one must characterize the nature and the extent of the differences between “real-world” database heterogeneity and the database heterogeneity encountered in the controlled experiment. Specifically, the experiment simplified the query-translation problem in at least five ways.

6.2.6.1 A single clinical data set

I distributed to each of the database-design groups the same clinical data set that I used to design the FER schema. Because I instructed each database-design group to include in its relational database implementation *all* of the data represented in the clinical data set, almost all data elements modeled in the FER schema appeared in each relational schema. This situation is unrealistic, because certain “real-world” legacy databases may be missing data elements that appear in the FER schema; missing data may make the mapping problem more difficult and may compromise the correctness of applications that use the TransFER model. However, the issue of FER data elements missing from relational database implementations is not central to the evaluation of the TransFER translation methodology, *per se*, because the translation methodology assumes that all target databases contain the same information as the FER schema. I can test the validity of this assumption in subsequent evaluation experiments and develop methods to inform applications that information is missing, if necessary.

6.2.6.2 A circumscribed clinical data set

The scope of the clinical data set that I distributed to the database-design groups was significantly smaller than the scope of data sets that are modeled in real-world clinical databases. A larger number of data elements increases the complexity of the relational database designs and results in greater representational heterogeneity among the designs. For example, the data set used in the experiment included only ICU patients, whereas a real-world data set might also include outpatients, ward inpatients, etc., resulting in a more complex relational schema. However, although the clinical data set that I used did not generate a complete set of representational heterogeneities, it was sufficiently large and complex to provide an opportunity to address several important issues of representational heterogeneity. As described in Section 5.5.1, representational heterogeneities of several kinds appeared among the three relational databases that were generated for the experiment. These heterogeneities provided a set of realistic and non-trivial mapping problems for the TransFER methodology.

6.2.6.3 A limited number of target databases

Realistically, TransFER will need to map the FER query model to clinical databases at *hundreds* of provider sites. The experiment that I conducted required TransFER to map the FER query model to only a handful of clinical databases. A greater number of target databases undoubtedly will create a greater degree of representational heterogeneity to be resolved by the TransFER methodology. However, the degree of heterogeneity probably is not proportional to the number of clinical database sites: There may be only a finite number of ways in which typical clinical data is modeled relationally, and certain sites may share the same relational implementation because they have purchased commercial clinical databases from the same vendor. Although a handful of clinical database implementations did not exhaustively test TransFER's ability to resolve representational heterogeneities, it did indicate whether the TransFER methodology is a *feasible* means of resolving the heterogeneities that typically arise in the representation of clinical data.

6.2.6.4 A limited number of clinical queries

The FER clinical data schema and the TransFER query-translation methodology are intended to support the data-access needs of numerous and diverse decision-support applications. Although the physicians who generated the set of test queries for the experiment attempted to simulate these data-access needs as completely as possible, the full range of queries required by medical decision-support applications could not be represented by the relatively modest set of queries that the physicians formulated. However, the query authors were encouraged to produce and had the background to produce a diversity of data-access requests related to patient care, clinical research, outcomes assessment, and clinical practice guidelines. Although the queries were not exhaustive, they were representative in terms of content, complexity, and diversity.

6.2.6.5 A single relational DBMS

All of the subject groups implemented the relational databases that they designed using the Sybase relational DBMS. Realistically, health care institutions implement clinical databases using a variety of relational DBMS platforms, such as Sybase, Oracle, Informix, IBM DB2, etc. Each of these platforms differs slightly in the dialect of SQL that it provides as a query interface. By requiring all of the database-design groups to use Sybase, I avoided the added complication of translating ReFER queries to multiple dialects of SQL. However this simplification does not detract appreciably from the validity of the evaluation because the objective of the TransFER methodology is to resolve *representational heterogeneity* rather than *access heterogeneity* among clinical databases [Sheth and Larson 1990]. Access heterogeneity entails primarily differences in network connections to database servers, application programming interfaces to DBMSs, and syntactic details of SQL. Several commercial vendors already have addressed the problem of access heterogeneity among relational DBMSs [Burleson 1995], and I could incorporate these technologies into future versions of TransFER to provide access to various database-system platforms.

Because of the factors described in Sections 6.2.6.1–6.2.6.5, the problems that I presented to the TransFER methodology in the evaluation experiment were not *complete* with respect to the representational heterogeneities that exist among real-world clinical databases. However, the problems were *representative* of those heterogeneities. Because the experiment provided a realistic, representative, and non-trivial test of TransFER's capabilities, the favorable results indicate that TransFER is a significant novel technology with potential utility for sharing decision-support applications.

The results of the experiment also indicate that additional research and evaluation is required before one can deploy TransFER as a general technology for translating database queries automatically across heterogeneous database sites.

6.3 Future Work

Although the TransFER methodology shows promise as an effective model for sharing decision-support applications that include database queries, more work is required before TransFER can be applied to solve the real-world problems of application portability. Outstanding research issues include:

- Development of a global FER schema of clinical information that is sufficiently comprehensive to support a variety of clinical decision-support applications yet sufficiently constrained to represent the contents of most clinical databases

- Development and implementation of additional optimization techniques to reduce further the complexity of relational queries generated by the TransFER compiler

- Development of application programmer interfaces (APIs) to TransFER from a number of programming languages, so that ReFER queries can be embedded in and integrated with the code of decision-support applications

- Development of a sound method to represent that a certain datum is not available in a clinical database and to notify decision-support applications of this fact at the time the datum is requested

- Design and execution of further evaluation experiments that involve actual clinical databases and decision-support applications

Work in these areas is necessary before the concept of TransFER described in this dissertation can be realized in practice. Several extensions to the TransFER model are also worth investigating:

- Development of additional query-translation algorithms that can translate ReFER queries to the query languages of *nonrelational* legacy database systems, such as object-oriented databases and hierarchical databases

- Enhancement of the TransFER query-translation algorithm to aggregate data from *multiple* heterogeneous databases, so that TransFER is more valuable for multisite research and decision-support applications

- Combining the FER data model with the type system of a programming language to create a *database programming language* [Atkinson and Buneman 1987] , which could be used to build decision-support applications that efficiently combine data access with knowledge-based inference.

6.4 Conclusions

Computerized decision-support tools may well become a critical resource in a complex healthcare system that faces pressure to maintain quality while reducing costs. The effectiveness and acceptance of these tools depends, at least partially, on their ability to access up-to-the-minute patient-specific data in an accurate, timely, and unobtrusive manner. As electronic clinical databases increase in prevalence and sophistication, they offer fabulous potential to provide data to decision-support tools quickly and automatically, allowing the tools to be integrated smoothly into the healthcare-delivery process. This desirable model of application–database interoperability, however, is significantly complicated by the extensive and multifaceted heterogeneity that exists and will continue to exist among clinical databases. Database heterogeneity makes it exceedingly difficult to deploy an effective and proven decision-support tool across multiple healthcare-provider sites because the tool must be painstakingly customized to the highly variable database specifications at each site. Until now, no technology has been available to automate the customization process in a principled and efficient manner.

The TransFER query-translation methodology constitutes an important step towards such a technology. The formal data model and query language allow the developers of decision-support applications to specify queries precisely enough to be unambiguous, yet abstractly enough to be portable across heterogeneous databases. The modular, powerful, and declarative database-mapping language allows database programmers to specify schema-to-schema transformations for a large variety of relational databases in a manner that is intuitive and that supports efficient query translation and automated query optimization. The attribute-grammar-based translation model allows the TransFER compiler to use a fixed set of mapping expressions to translate an arbitrary number of queries such that the resulting database programs are semantically equivalent and locally executable. Although the computational efficiency of automatically translated queries may be currently insufficient to meet the demands of many decision-support applications, the formal elements of the query-translation model support the development of many further optimization techniques. Additionally, the extensibility of the mapping language via external-function calls allows TransFER to incorporate future technologies for resolving database heterogeneity into the current query-translation model. Given the demonstrated soundness of the basic query-translation model, more powerful optimization techniques and more advanced database-mapping technologies may well complete the link provided by TransFER between decision-support applications and heterogeneous relational databases.