Bridging the Representational Heterogeneity of Clinical Databases

Walter Sujansky and Russ Altman

Section on Medical Informatics, MSOB x215
Stanford University School of Medicine, Stanford, CA 94305
sujansky@camis.stanford.edu, altman@camis.stanford.edu

Abstract

Clinical decision-support (CDS) applications are difficult to share among health care institutions because such applications require patient-specific data, and the electronic representation of clinical data varies widely across institutions. Our technology facilitates the sharing of CDS applications by providing a standard abstract database schema of clinical information and a means to automatically translate queries that are formulated with respect to the standard schema to equivalent queries that are valid with respect to arbitrary relational representations. We briefly describe our technology, discuss the inadequacies of extant methods that address the same problem, and present a set of formal evaluation criteria by which our technology can be shown superior to these methods. The evaluation criteria are applied in a formal controlled experiment we have designed based on the clinical ICU data set.

I. Addressing the heterogeneity problem

One of the major obstacles to the widespread integration of clinical decision-support (CDS) technologies in the daily practice of medicine is the heterogeneous nature of existing clinical database systems. This obstacle exists because most CDS applications require access to patient-specific data, which is electronically represented in heterogeneous ways across health care provider sites. This "representational heterogeneity" may take several forms, including data model heterogeneity, database schema heterogeneity, naming heterogeneity, level-of-abstraction heterogeneity, etc. [Sheth & Larson, 1990]. In order to share CDS applications, which are typically very expensive to develop, among provider sites, one must "bridge" between the representational models of clinical data that CDS applications expect and the models that operational clinical databases provide. For example, the porting of standard clinical alert rules across hospitals [Hripcsak, *et al.*, 1990] requires that local programmers *customize* the data-access component of each rule to correspond to the representational model of their clinical information system. Not only is this a laborious and costly process, because it requires a careful translation of the intended semantics of each rule to match the represented semantics of the local database, but it may result in inaccurate and inconsistent translations, because there currently exists no standard *domain model* of clinical data to ground the semantics of shared rules.

Our work addresses these problems by proposing a technology for reducing the costs of porting CDS applications that require patient-specific data and for increasing the consistency with which such applications can be implemented across provider sites. The key component of our methodology is a site-independent **reference schema** of clinical information that we specify using a semantic data model. CDS applications query the reference schema using a high level **query language.** A **translating compiler** translates the site-independent queries to semantically equivalent local queries, using a set of site-specific mappings. The mappings are specified in a declarative **mapping language** that we have designed. The advantage of our methodology is that local database administrators can *systematically* specify the mappings between the reference schema and a local "target" database; the compiler subsequently translates an arbitrary number of queries automatically (eliminating the effort required to port each query) and applies the same semantic mappings to each translation (ensuring that each query is translated consistently).

To deploy the methodology that we have defined, we obviously must address the technical issues of design and implementation. However, an equally important challenge is defining the criteria by which our methodology can be judged *effective* and to design an evaluation strategy that assesses our methodology with respect to these criteria. In this paper, we describe the technical content of our methodology briefly, and focus on the issues related to the definition of useful evaluation criteria and to the design of our evaluation experiment, which involves the ICU clinical data set provided.

II. The Methodology

Our "bridging" methodology consists of 4 components:

1. **A semantic data model**, called FER (Functional Entity-Relationship model) for modeling clinical data in an abstract way, i.e. independently of any site-specific database representation.

2. **A query language**, called ReFER, that corresponds to the FER data model and that allows developers of CDS applications to specify data retrieval requests with respect to an FER schema

3. **A mapping language**, called ERA (Extended Relational Algebra), that is based on the relational algebra [Codd, 1972] and that allows the constructs of an abstract FER schema to be mapped formally to equivalent constructs of a site-specific relational database schema.

4. **A query-translation module**, called TransFER, that applies ERA mappings to automatically translates ReFER queries into site-specific executable database commands.

We designed the FER data model to be sufficiently general and expressive to subsume diverse clinical database implementations. To this end, FER combines the semantic data modeling features of the entity-relationship (ER) data model [Chen, 1976] and the functional data model [Shipman, 1981]. The ER model distinguishes between attributes and relationships, so that ER schemas commit to a particular relational representation that, in fact, may vary among implemented databases. The functional model restricts the information that may be represented regarding associations among database objects, so that functional model schemas cannot express conveniently the complete semantics that may be represented in implemented databases. The FER data model combines the ER and functional data models to remedy of the deficiencies of each model.

The syntax and semantics of the ReFER query language are based on the domain calculus [Pirotte, 1978] and closely resemble the syntax and semantics of the functional query language DAPLEX [Shipman, 1981] and the entity-relationship query language EERC [Hohenstein, 1989]. Unlike DAPLEX and EERC, each construct of the ReFER query language can be mapped explicitly to a relational algebra expression represented in ERA. The ERA expressions corresponding to each construct in a valid ReFER query can be combined (using formal composition rules that we have specified) into a single ERA expression that is semantically equivalent to the original ReFER query. A similar strategy is applied in the OODAPLEX query language (OODAPLEX [Dayal, 1989]), except that the relational mappings corresponding to OODAPLEX constructs are fixed and cannot accommodate diverse relational implementations, as they can in our methodology.

A query-translation module, TransFER, composes the ERA expressions corresponding to each construct of a ReFER query. TransFER does this by applying an attribute grammar [Knuth, 1968] that formally defines the relational semantics of each syntactic construct in the ReFER language, such as primitive entities and logical connectives. The result is a single ERA expression equivalent to the input ReFER query, which TransFER subsequently optimizes and translates into the site-specific variant of SQL. The translated ReFER query is then ready to be executed locally. The details of the design and implementation of the TransFER methodology are discussed in [Sujansky, 1992].

At a high level, then, we can summarize the steps required to bridge representational heterogeneity:

1. A site-independent reference schema of clinical data is specified, using the FER semantic data model, which represents the medical concepts and associations among concepts typically stored in clinical information systems.

2. Based on the FER reference schema, local database administrators use the ERA mapping language to specify mappings between their local database schemata and the global FER schema.

3. Application developers (for example, the authors of clinical alert rules) write ReFER queries against the FER reference schema to retrieve patient-specific data.

4. Applications, which include ReFER queries, are distributed in source form to local database sites, where TransFER compilers automatically translates the ReFER components into site-specific queries that retrieve the requested data from the local clinical database.

III. Previous research

Our methodology addresses the need to efficiently and reliably share CDS applications across sites that have heterogeneous clinical database implementations. Previous research in expert systems and in database interoperability have produced other general methods that address this need:

Method 1. Manual interpretation and execution of queries. In this method, used by Mycin [Buchanan & Shortliffe, 1984] and QMR [Miller & Masarie, 1990], the physician user personally interprets the CDS application's information requests and responds to them manually by entering patient-specific data that the user knows or that is documented in the medical record. Effectively, the user bridges the representational heterogeneity himself each time the application makes an information request.

Method 2. Manual translation of queries. In this strategy, used by the Arden medical knowledge representation language [Hripcsak et al., 1990], local database programmers manually translate each query that appears in an Arden program to a query that is consistent with the local clinical database implementation.

Method 3. Manual translation of data. This strategy, used by several clinical research databases, such as ARAMIS [Blum, 1981], requires that medical transcriptionists manually abstract clinical data from primary patient records into a uniform format that is then electronically uploaded into a centralized

databank. The "normalized" data then is queried by CDS and research applications.

Method 4. Automated translation of data via HL7. This strategy, used by several commercial clinical information systems, entails that clinical data is automatically uploaded from local "feeder" systems into a normalized clinical data repository, using the standard HL7 medical data interchange format [Rishel, 1988].

Method 5. Automated translation of queries via procedural mappings. This strategy, used by the Physician Workstation environment [Annevelink, *et al.*, 1992], automatically translates queries issued against a reference data model to site-specific operations required to retrieve data from underlying databases.

Each of these methods has deficiencies with respect to at least one of three desirable properties: **Automation**, **efficiency**, and **latitude**.

Automation of query translation

Methods 1 – 3 all involve labor-intensive translations. Method 1 requires that users interpret and manually respond to all queries; method 2 requires that programmers manually translate all queries; method 3 requires that medical records personnel manually abstract all data of interest that is captured in the medical record. The goal of automation is to relieve users, programmers, and medical records personnel of tedious and error-prone labor. The manual translation task is especially onerous when it must be performed frequently, and methods 1 – 3 require human intervention each time an existing query is issued, each time a new query is added, and each time a source database is updated, respectively.

An important assumption of the automation property is that the automatic translations of queries and of data are semantically *correct*. Correct translations preserve the semantic equivalence of source and target queries. Although it may seem obvious that automation without correctness is not useful, certain current efforts for automating the translation task suffer from errors of correctness. For example, Method 4 specifies that disparate information systems may share clinical data by automatically translating the data to and from a standard message format (HL7). However, because the message format is not based upon a standard *reference model* of clinical data, the semantics of data transmitted in HL7 messages cannot be formally specified and may be inconsistently translated by the sending or the receiving application [McLinden, *et al.*, 1990].

Efficiency of translated queries

When heterogeneous representations are reconciled via the translation of queries, either manually or automatically, the target queries should be as efficient as possible. In some cases, the run-time costs of evaluating queries is critical to the objectives of the application. In general, the costs of executing a query depend on where the data resides, how the data is represented, how the query is decomposed into sub-parts, which agents execute which sub-parts, and in which order the sub-parts are executed [Ullman, 1989].

Method 1, which requires physician users to manually "execute" queries by entering data—possibly after looking it up in the patient record—is clearly the least efficient. Method 2, which relies upon programmers to manually translate queries that are subsequently executed by the database, takes advantage of DBMS query facilities, but relies upon the ability of programmers to formulate efficient queries. Sometimes, the optimal specification of queries may depend upon knowledge that the programmers do not have or that is subject to change. Method 5 (methods 3 and 4 do not apply) requires that programmers *procedurally* specify the mappings between the data representation expected by decision-support applications and the data representation existing in clinical databases. The query engine subsequently uses the procedural mappings to import data from the clinical database into the CDS application. However, this process cannot be automatically optimized, because procedural mappings cannot be decomposed, reordered, and recombined in more efficient ways [Annevelink et al., 1992]. To maximize the potential for optimization, a *declarative* representation of mappings is necessary, because a query optimizer can inspect and manipulate declarative mappings [Genesereth & Nilsson, 1987].

Latitude of portability

A large variety of legacy clinical databases exists. The objective of any method for sharing clinical queries is to support the mapping of queries to as many of these databases as possible. We refer to the proportion of legacy database implementations that a method can support as the method's *latitude*. The latitudes of methods 1 – 5 actually are quite large, but the methods achieve broad latitude by sacrificing automation, efficiency, or correctness.

In general, one must reduce the latitude of a "bridging" method in order to increase its automation or efficiency. This is because computational methods to automatically and efficiently translate queries require formal models for describing data representation, and these models place constraints on the representations that may exist in legacy databases. In some cases, the constraints may be excessive, resulting in a bridging method that applies to only a small proportion of legacy databases. Few application developers will conform to such a method. However, superior bridging methods might increase automation or efficiency while sacrificing much less latitude. A method that applies to 80% of the existing clinical database implementations might be adopted as a standard by CDS application developers. The remaining 20% of the database sites could modify their implementations to conform to the bridging method, reaping the benefits of automation and efficiency for their troubles, or they could default to one of the current methods, remaining no worse off than they are today. Therefore, although a new method for sharing clinical queries may result in a reduction in latitude, the method might nevertheless, constitute an improvement over the status

quo.. One must consider the extent of the trade-off between latitude and automation/efficiency when evaluating new methods for bridging representational heterogeneity.

IV. The evaluation

Given the design and implementation of our methodology, we must determine how to evaluate its effectiveness. In general, new computer applications should be judged by their ability to perform a useful function that could not be previously performed, to perform a function more effectively than it could be previously performed, or to perform a function more efficiently than it could be previously performed. By some criterion, a new application should constitute an *improvement* over the status quo. .

We have designed an experiment to evaluate whether the TransFER methodology is an improvement over the existing methods described in Section 3. The experiment entails five steps:

1. Three groups, each consisting of one clinician and one database expert, design a relational database schema based on the same clinical data set, specifically the ICU data set provided for this conference.

2. We design an abstract reference schema, using the FER model, that represents the same ICU clinical data set.

3. A small group of physicians informally formulate a set of queries that request clinically relevant information available in the data set and relevant to decision support. The physicians distribute the queries to the three database-design groups, which manually encode them in SQL in a manner consistent with their particular relational schema. We also encode the queries, using ReFER, in a manner consistent with the FER schema we defined.

4. We map the FER schema to each relational schema using the ERA mapping language.

5. The TransFER compiler translates the ReFER queries to three sets of SQL queries such that each set is consistent with one of the relational database schemata.

The hypotheses of the experiment are 1) that our methodology effectively can map a reference schema of clinical data to three independently designed relational database schemata, all of which represent the same data set, and 2) that our methodology automatically can translate queries formulated against the single FER reference model to queries that can be executed against all three relational schemata.

To systematically evaluate whether our methodology is more effective than the methods described in Section 3, we have specified a set of criteria by which our methodology should be judged *inferior* to the existing methods. The rationale for these "failure criteria" is that satisfaction of any of them indicates that our methodology no better than existing methods for sharing clinical queries. In this sense, the failure criteria constitute an experimental "null hypothesis" for the evaluation of our methodology. The failure criteria are:

1. **Latitude failure.** The TransFER mapping language cannot fully map the FER schema of clinical data to most legacy relational databases, and therefore the TransFER compiler cannot translate queries automatically. Latitude failure results in a return to the manual translation of queries that is characterized by method 2.

2. **Declarativeness failure.** The TransFER mapping language can map the FER schema to most legacy relational databases, but only by resorting to the liberal use of procedural functions. Procedural functions are encapsulated and cannot be manipulated by the TransFER query optimizer (whereas combinations of ERA operators can be inspected and reordered by the optimizer). Also, the evaluation of procedural functions takes place in the host environment rather than in the database server, so it cannot benefit from the indexes and other query optimization features of commercial relational DBMSs. Declarativeness failure results in a return to the procedural specification of mappings and the limited optimization of queries that are characterized by method 5.

3. **Semantics failure.** The TransFER mapping language can map the FER schema to most relational databases using declarative ERA mappings, but the TransFER compiler generates target queries that are semantically inconsistent with the input queries. The source of error may be either the quality of the database-specific mappings or the algorithms used by the compiler. In either case, semantic failure indicates that the TransFER method is unreliable and should not replace existing methods.

4. **Efficiency failure.** TransFER can map the FER schema to most legacy databases using declarative mappings and TransFER can generate semantically correct target queries, but the TransFER query optimizer cannot generate queries that are efficient enough for most client applications. For example, the performance of the queries is too poor for them to be executed as background processes in an integrated clinical alerting system, such as the HELP system [Kuperman, *et al.*, 1991]. Efficiency failure

results in a return to the manual translation and optimization of queries that is characterized by method 2 or in a return to the translation of data that is characterized by methods 3 and 4.

5. **Model failure.** TransFER can map the FER schema to most databases declaratively and can generate semantically correct and efficient target queries, but the FER data model and the ReFER query language are not sufficiently expressive to represent the semantics of clinical queries that are typically required by decision-support applications. Model failure means that decision-support applications must perform certain data operations themselves that could be performed more efficiently by the underlying DBMS if those operations could be expressed in the ReFER query language. Model failure precludes the optimal allocation of operations between the application and the DBMS because the operations are not known to the TransFER optimizer nor the TransFER query engine, and therefore results in sub-optimal query plans, as characterized by method 5.

V. Conclusions

We have described a novel methodology that addresses a significant obstacle to the widespread implementation and use of clinical decision-support systems: the representational heterogeneity of clinical databases. The methodology draws on previous research in semantic data modeling, high-level query-language design, and grammar-directed translation. Recognizing that competing methods exists for bridging representational heterogeneity, we have defined three important criteria with respect to which all such methods should be evaluated: Automation, efficiency, and latitude. To evaluate the methodology and to compare it to the existing methods, we have designed a formal controlled experiment (currently underway) that involves the design of relational databases for the clinical data set provided. We have also specified a set of "failure criteria" that constitutes a null hypothesis for the experiment. We assert that, although passing the failure criteria does not prove that our methodology is generally applicable, it strongly suggests it is superior to existing methods and that more exhaustive evaluations are warranted.

Bibliography

Annevelink, J., and Young, C. Y (1992). Heterogeneous database integration in a physician workstation. In P. D. Clayton (Ed.), *Proceedings of the Fifteenth Symposium on Computer Applications in Medical Care*. New York: McGraw-Hill, pp. 368-372.

Blum, R. L. (1981). Displaying clinical data from a time-oriented database. *Computers in Biomedical Research*, 11: 197.

Buchanan, B. G.& Shortliffe, E. H. (1984). *Rule-Based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project*. Reading, MA: Addison-Wesley.

Chen, P. P.-S. (1976). The Entity-Relationship Model—Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1(1): 9-36.

Codd, E. F. (1972). Relational completeness of data base sub-languages. In R. Rustin (Ed.), *Data Base Systems* . New York: Prentice-Hall.

Dayal, U. (1989). Queries and views in an object-oriented data model. *Proceedings of the Second Workshop on Database Programming Languages.*

Genesereth, M. R.& Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. Los Altos, CA: Morgan Kaufmann.

Hohenstein, U.& Gogolla, M. (1989). A calculus for an extended entity-relationship model incorporating arbitrary data operations and aggregate functions. In C. Batini (Ed.), *Proceedings of the Seventh International Conference on Entity-Relationship Approach*, pp. 129-148. Rome, Italy: Elsevier Science Publishing, Inc.

Hripcsak, G., Clayton, P. D., Pryor, T. A., Haug, P., Wigertz, O. B.& Van der lei, J. (1990). The Arden Syntax for Medical Logic Modules. In R. Miller (Ed.), *Proceedings of the Fourteenth Symposium on Computer Applications in Medical Care* . Washington, D.C.: , pp. 200-204.

Knuth, D. E. (1968). The semantics of context-free languages. *Mathematical Systems Theory*, 2.

Kuperman, G. J., Gardner, R. M.& Pryor, T. A. (1991). *HELP: A Dynamic Hospital Information System*. New York: Springer-Verlag.

McLinden, S., D'Ascenzo Carlos, G.& Oleson, C. E. (1990). The evolution of a standard for patient record communication: a case study. *Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care*: pp. 239-243.

Miller, R. A.& Masarie, F. E. (1990). Quick Medical Reference (QMR): A microcomputer-based diagnostic decision support system for general internal medicine. In *Proceedings of the Symposium on Computer Applications in Medical Care* . Washington, D.C.: IEEE Press, pp. 986-988.

Pirotte, A. (1978). High Level Data Base Query Languages. In H. Gallaire & J. Minker (Ed.), *Logic and Data Bases*. Plenum Press, pp. 409-433.

Rishel, W. (1988). Pragmatic Considerations in the design of the HL7 protocol. *Proceedings of the Symposium on Computer Applications in Medical Care*: pp. 687-690.

Sheth, A. P.& Larson, J. A. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3): 183-236.

Shipman, D. (1981). The functional data model and the data language DAPLEX. *ACM Transactions on Database Systems*, 6(1): 140-173.

Sujansky, W. (1992). *An Extended Relational Algebra for Bridging Representational Heterogeneity among Relational Databases. Technical* Report, Section on Medical Informatics, Stanford University.

Ullman, J. D. (1989). *Principles of Database and Knowledge-base Systems*. Rockville, MD: Computer Science Press.