Coordinating Domain Heterogeneous Data

Overview

When data is shared among collaborators with related – but not identical – interests, users find that existing industrial and research techniques fall short of expectations. A number of ways exist for users to combine independently created data sources together. Because the data sources are independent, each source is likely to have its own schema (i.e., their own representation of concepts). To translate between the sources, a *mapping* relates concepts in one schema to the concepts in another. For example, "earthquake" in one schema may be "tremor" in another. Mappings allow easy translation of queries and data. This overall problem is called data exchange [1]. If there is a central authority, this generally is called *data integration* (see [2] for a recent survey) and all of the sources are mapped to that central authority. However, if there is no central authority (which is typical among scientists and other small communities), a common structure is the Peer Data Management System (PDMS) (e.g., [3]), in which each peer in a Peer To Peer (P2P) network is assumed to have its own database. However, even after 8 years of research PDMSs have not become popular. This is due to difficulties in deploying them. For example, it is currently difficult to aggregate answers from multiple sources. My goal in this proposed research is to make such sharing of domain heterogeneous data (i.e., data that comes from different domains) easier by (1) creating different aggregate mapping styles to allow changes of data to be reflected in other sources (2) improving the ways in which they handle updates and (3) improving the ease of creating and use of PDMSs so that they truly are a viable solution. This work can then be applied to other data exchange settings as well.

Previous Research Activities and Progress

For the past five years I have focused on managing heterogeneous data interactions, both in generic and specific, application-based settings.

ARTIFACT: Advanced Research, Techniques, and Informatics for Future Advantages in Construction Technology

For the past five years, I worked with Civil Engineering Researcher Dr. Staub-French, and Computer Supportive Collaborative Work researchers Dr. Booth and Dr. Tory on the ARTIFACT project. The overall goal of ARTIFACT is to improve the process of building a building. My part of the project focuses on the managing data used in designing and constructing a building. An initial study discovered that data flow inefficiencies make the task hard and the overall process inefficient. For example, suppose that the general contractor wants to save $50,000 by lowering the ceiling 3 cm. The contractor must coordinate with those in charge of the other systems (e.g., the duct work) to determine the impact of this change. This is because although the initial plans may come from the same source, they do not allow for easy querying of these kinds of details. Additionally, sub-contractors make their modifications in different, incompatible applications. Today this is a slow, tedious and error-prone process. Industrial consortiums have attempted to allow data to flow between applications by creating standard XML schemas (e.g., the construction schema, ifcXML), but these fail to meet their goals [4].

I have had four different students work on this project. The first, Jiemin Zhang, a MSc student, determined the basics of how to integrate and query the design data. Most data integration techniques assume that there is an easy, non-lossy way of extracting all of the information from a given source; that is not the case in ARTIFACT. For example, extracting ifcXML from Revit Building Systems, a popular CAD design tool, will enable many types of data to be extracted about the walls and the other structural material, but not information about duct works and relationships between them. Additionally, although ifcXML can model much of the structural information, it is very difficult to find out the answers to any queries that the user would like – for example, determining the location of a wall involves chasing down a series of approximately 10 idRefs per data point. This means that it is almost impossible for the user to understand how the data connects; a new conceptual model must be formed [5, 6]. As we verified by looking at research in other domains, similar problems arise. Specifically we found that (1) XML standards improve the facility of information exchange among applications (2) XML is a highly flexible data format, but this power of expressiveness comes at the cost of increased complexity and decreased usability, and (3) The existence of multiple XML domain standards significantly compromises usability as users must evaluate, compare, integrate and/or coordinate standards.

My second student, April Webster, is integrating the spatial data with the non-spatial data [7]. My third student, Michael Lawrence, is investigating how to link cost estimate data to design data so that it is easier to create the estimates initially and to update them as the design changes.

Finally, a great deal of information in construction projects is stored outside applications; for example, the meeting minutes are stored in PDF documents, which then must be manually integrated or laboriously searched by hand. An undergraduate student, Jamila Salari, started working on how to create an ontology from the PDF repository at hand[8]. This work will allow users to be able to have their queries be expanded or narrowed automatically if there are too many or two few answers – for example, if a query about “air conditioning” returns too few results, using the automatically generated ontology the system may discover that “HVAC” (which stands for Heating, Ventilating, and Air Conditioning) is a generalization of air conditioning, and may return to the user additional results. This problem is being looked at more generally with my colleague Laks Lakshmanan and our student Ali Moosavi.

JIIRP: Joint Infrastructure Interdependencies Research Project

I was also a member of a 13 investigator (from CS, Electrical Engineering, Civil Engineering, Business, Geography, and Health Care and Epidemiology) interdisciplinary project to improve disaster management. Part of this project was to help build a simulation system, including assessing the impact of a possible disaster (e.g., an earthquake) and a response when a disaster happens [9]. Additionally, my student, Jian Xu, and I investigated the specific data management aspects. One aspect of this is how to answer queries from multiple sources easily, particularly when (1) there is not enough time to create a global, mediated schema which all sources can use (2) the sources involved have very different domains – i.e., as we describe it they are *domain heterogeneous* and (3) there are many aggregation queries that need to be answered, which is not possible given current techniques. Jian and I have begun exploring how to extend PDMSs to account for these difficulties. Additionally, this project has illustrated many new research problems, including how to integrate relational data [10] with GIS data.

Objectives – short term and long term

A peer data management system (PDMS) is a great idea: users are able to query using their own schemas, and they can retrieve tuples from all sources in the network. However, PDMSs have existed for 8 years, and they have not yet achieved prominence. I posit that there are numerous issues preventing the spread of the PDMS, all of which stem from prohibitive setup and maintenance effort for the relative benefit. The benefit of a PDMS is that you can have your data integrated seamlessly without having to change the schema or application that the data is currently managed by. However, in practice, it is impossible to use existing tools to allow this to happen. My short and long term goals for this grant are to resolve these issues. Specifically, my short terms goals (described in more detail below) are to (S1) extend mappings between schemas to specify aggregation of values, and determine how to rewrite queries to aggregate over their decomposed components (S2) determine how to correctly evaluate the values that are returned by the aggregation, (S3) extend the mapping paradigm to handle domain heterogeneity and (S4) be able to coordinate data so that changes in one data source can also automatically be reflected in another data source. My longer term goals are to (L1) create a principled way to understand and extend schema standards so that they can be useful in the applications for which they are needed, (L2) figure out how to understand the data that has been exported from the application and how to use it.

Short Term: adding support for aggregation, domain heterogeneity, and updates

My short term goals are to focus on the problems raised in my now ending disaster management and civil engineering projects. Although JIIRP and ARTIFACT are very different, both require dealing with mappings between sources that require aggregation, when there may be few direct concepts shared between the sources. When dealing with sources of data that are curated independently, typically the different sources have different schemas – i.e., different representations of concepts – this is what is referred to as *semantic heterogeneity*. For example, the concept of "wall type" in one schema may include "material" and "fire rating" in a schema that is concerned about fire prevention but "thickness" and "composition" in a schema about earthquake recovery. In order to translate a query over one schema into a query over the other schema, a mapping must be provided between the schemas – e.g., "material" in the fire prevention schema may mean the same thing as "composition" in the earthquake recovery schema.

Currently, when managing data that comes from multiple schemas, it is assumed that there is an direct connection between elements in the two sources. The sources may have s*chema heterogeneity* – they may have a different representation of the same real-world object (e.g., "material" and "composition" above). *Domain heterogeneity* [11] (S4) on the other hand, relates to databases with different real-world object types (e.g., design data vs. cost estimates). Databases which are domain homogeneous but schema heterogeneous are discussed intensively in existing data integration applications e.g., the canonical examples of integrating bibliography records in different online publication databases. Disaster management data has strong domain heterogeneity – we need data that describes physical properties of land, buildings and critical facilities; we need damage assessment databases for various lifeline infrastructure including electricity, water, gas, communication; we need data that describe service information for security, public health and emergency response policies. JIIRP and ARTIFACT point to very similar underlying problems.

Another issue common to both scenarios and necessary to make PDMSs useful is that of aggregation (Goals S1 and S2). In JIIRP, a "Division" is defined as a functional infrastructure (e.g., a hospital) which consists of several buildings. Seismic damage assessment for buildings is collected by earthquake engineers. The user of the division database wants to compute the structural damage and monetary loss of *divisions* using the damage assessment on the *buildings*. As a simplified example, the damage of a division is estimated by taking the average damage to the consisting buildings. Currently, the division database is populated by manual aggregation queries written for each individual division. This is especially difficult because people working with the division schema are not familiar with the schema of the building damage assessment. Because the information for each division must be populated manually, users are loath to define new divisions or update the division damage records. Similar problems occur when trying to map a building design estimate to a CAD model in ARTIFACT – for example, an estimation for the unit cost of a particular type of wall may require examining the total length of such walls grouped by height. Handling aggregation queries on domain heterogeneous sources is not something that is feasible in traditional data exchange work. Cohen et al. [12], discuss rewriting aggregation queries using views and provides a theoretical framework to unify previous works on rewriting queries with aggregation functions such as in [13, 14]. While the above works mainly focus on using aggregation views for query rewriting, our approach differs because we focus on discovering aggregations and the grouping functions to transform a query to aggregations. In a recent study [15], the authors discuss aggregation query answering in data exchange. The setting used differs because those works are looking at all of the possible worlds that might exist in the target. In this problem, however, a target instance already exists. Therefore, the techniques discussed in [15] do not trivially apply.

A final short term goal (S3) is how to coordinate data so that changes made to one data source may also be reflected in another related data source – which we describe as *data coordination*. In this way, the manager of a data source can ensure it is up to date and consistent with the latest data provided by related sources upon which it depends. For example, in ARTIFACT, a building's cost estimate depends in precise ways upon its design (e.g., walls above a certain height require completely different construction procedures). Previous work (e.g., Orchestra [16]) operates from the perspective of change (i.e., how a change to data source A corresponds to a change to data source B.) This is a sensible approach for domain homogeneous, schema heterogeneous sources, as each event has a corresponding action (e.g., an insertion in A corresponds to an insertion in B.) However, when we have heterogeneous domains and relationships which involve aggregation (goals S1 and S2), there are many different events which can result in the same action.

The goal of data coordination is to solve precisely the kinds of what-if queries that have been raised in the ceiling lowering example above. If we can support this kind of analysis, then people in many domains – not just civil engineering and disaster management – will be able to explore their options more thoroughly rather than being stuck at a local minimum. Decreasing the hysteresis in a system will allow for better design to be fully explored more quickly than current techniques permit.

I anticipate that coming to the root of these problems including determining how to formalize the necessary mappings and how to answer queries and coordinate data will take approximately three years from the beginning of this grant's tenure. By now we have a good understanding of the details of the problems and the steps we want to take. One of my students has begun working on how to rewrite user queries into aggregation over the sources in a PDMS [11], however this work has just started. Another is in the process of defining data coordination in heterogeneous domains[17], however he has much work left to go as well.

Longer Term Goal: increasing usability of data sharing systems

Once data sharing has been extended as mentioned above, there still exist many problems with the usability, as we have seen in our work thus far. Increasing the usability of the systems is necessary to get people from different domains – not just database experts – to use data sharing systems such as PDMSs. One big problem is exporting data from non-DBMS sources, e.g., CAD data. This problem has been made easier by the existence of XML – many applications now export their data into XML standards that have been created by domain experts. However, these standards are conflicting, and it is difficult to choose which standard to export to, and it may be possible that none of the existing standards can handle all of the data. For example, exporting a simple building design can be done using any of three different XML standards, each of which is designed for a different purpose. Additionally, of the three standards the most comprehensive still only provided complete or partial support for 45 of the 57 concepts that the construction practitioners needed [18]. I have several goals to help solve this problem: (L1a) helping the user to decide how to understand which standard is most compatible with their work and (L1b) helping the user to see how to extend the standard in a principled fashion to allow their data to be fully expressed and shared. Since the schemas in these projects, as in many real world applications, are imposed on us, they do not adhere to the recommendations on the literature on understanding schemas (e.g., [19-21]). By seeing how the types of schemas that are used in practice differ from those in the literature, we will learn how these advanced schemas differ from those created for novice users.

Once users have picked a standard, they must export their data. There are two concrete goals: (L2a) understanding how the data has been exported and (L2b) determining how to map data to another source. One step along this path is to allow the users to draw a diagram of what they believe their data should look like – using UML or some similar modeling tool – and then extending schema mapping techniques to map the user's understanding of the data to how it is actually represented in the database.

However, while it is clear how to use existing techniques to help map the data, it is *not* clear how to use this information to allow the users to understand their data. This is the work that we would focus on.

Addressing these long term goals will allow users to better share their data and to make better, quicker decisions, since they will be able to explore options more thoroughly in a shorter amount of time. By being able to more easily understand and integrate data from multiple sources, the non-database experts will be more willing to use systems such as PDMSs, which will prevent them from having to spend time redoing the efforts of others or paying for additional experts to make this happen for them.

This process of making better, more informed, decisions is complimentary to the Business Intelligence Strategic Network (BIN) of which I am a member. BIN's goal is to enable users to make more informed business decisions, and the work proposed here dovetails nicely with those goals. The projects that I am proposing here will complement the work that I am doing in BIN, such as enabling a data warehouse to be designed top-down instead of bottom up. All of these goals are about the same thing: how can we make the best, most efficient use of the influx of data that we have today and help users to integrate the information from different sources. Solving these problems will allow Canadians to make the best decisions possible, which will help them to live more productive lives.

Training to Take Place Through the Proposal

My work is heavily dependent on the students with whom I work. In return, those students gain skills that help them to be productive members of society. My seven MSc students who have graduated have all found employment within the field or continued on in graduate school. Michael DiBernardo, Andrew Carbonetto, Xun Sun, and Shuan Wang are continuing to work in traditional software development at Novell, MDA, and Microsoft. Jiemin Zhang, who worked on ARTIFACT as a graduate student shows that the work that she did with CAD models transfers to skills necessary to analyze financial documents. Jie Zhao has recently completed an MBA in order to better understand the needs of business. Ting Wang is working on a PhD at the Georgia Institute of Technology. All of these students are using the skills that they learned through this research to better society.

I anticipate that for the first two years of the proposal I will fund my PhD student, Jian Xu, on this grant. I anticipate that for the first four months I will hire an undergraduate to help Jian with the development of a PDMS that runs on WestGrid [22]. I anticipate funding my PhD student, Michael Lawrence, for 1.5 years, since the ARTIFACT grant will run out in November after this proposal's tenure begin. After that I will fund two new PhD students to work on the longer term vision of increasing the comprehensibility of making data sharing systems, including PDMSs, easier to use.

In addition to the one summer undergraduate I have proposed, I will continue to work with undergraduates through independent studies, as I have with five previous undergraduates.

I also bring my research into the classroom, both at the undergraduate and graduate level. When teaching the introduction to relational databases course for undergraduates, I always describe the current research that I am doing. I find that the students are always attentive and appreciative, even though I explicitly state that it will not be on the final. At the graduate level, I integrate the research that I doing into the curriculum through both papers that we read and class discussions. I have had a number of students who were not previously interested in data management research become interested through this class, and a number of other students incorporate the ideas that they have learned through their classes into their own projects. This is one reason why I am or have been a member of four PhD committees (from Computer Science, Electrical and Computer Engineering, and Civil Engineering) and of an additional five MSc students' committees. I anticipate continuing with this trend; thus this grant will contribute to the training of more than the students explicitly mentioned in this proposal.