Natural Sciences and Engineering Research Council of Canada | Conseil de recherches en sciences naturelles et en génie du Canada

Committee / Panel 1507

1507

Institutional Identifier

System-ID (for NSERC use only) 145578593

FORM 101
Application for a Grant
PART I

Date 2010/10/26

Family name of applicant	Given name	Initial(s) of all given names	Personal identification no. (PIN)
Pottinger	Rachel	RA	**Valid** 290625

Institution that will administer the grant: British Columbia

Language of application: [X] English [ ] French

Time (in hours per month) to be devoted to the proposed research / activity: 50

Type of grant applied for: Discovery Grants - Individual

For Strategic Projects, indicate the Target Area and the Research Topic; for Strategic Networks indicate the Target Area.

Title of proposal: Improving Schema Understandability for Semantic Integration

Provide a maximum of 10 key words that describe this proposal. Use commas to separate them.
Databases, Data Management, Schema Understandability, Schema Mapping

Research subject code(s)		Area of application code(s)	
Primary	Secondary	Primary	Secondary
2711		801	

CERTIFICATION/REQUIREMENTS

If this proposal involves any of the following, check the box(es) and submit the protocol to the university or college's certification committee.

Research involving : Humans [ ] Human pluripotent stem cells [ ] Animals [ ] Biohazards [ ]

Does any phase of the research described in this proposal a) take place outside an office or laboratory, or b) involve an undertaking as described in Part 1 of Appendix B?

[X] NO [ ] If YES to either question a) or b) – Appendices A and B must be completed

TOTAL AMOUNT REQUESTED FROM NSERC

Year 1	Year 2	Year 3	Year 4	Year 5
50,638	50,638	50,638	50,638	50,638

SIGNATURES (Refer to instructions "What do signatures mean?")

It is agreed that the general conditions governing grants as outlined in the NSERC *Program Guide for Professors* apply to any grant made pursuant to this application and are hereby accepted by the applicant and the applicant's employing institution.

Applicant

Applicant's department, institution, tel. and fax nos., and e-mail

Computer Science
British Columbia
Tel.: (604) 822-0436
FAX: (604) 822-5484
rap@cs.ubc.ca

Head of department

Dean of faculty

President of institution (or representative)

The information collected on this form and appendices will be stored in the Personal Information Bank for the appropriate program.

PROTECTED WHEN COMPLETED

Version française disponible

Canada

Personal identification no. (PIN)	Family name of applicant
Valid 290625	Pottinger

SUMMARY OF PROPOSAL FOR PUBLIC RELEASE (Use plain language.)

This plain language summary will be available to the public if your proposal is funded. Although it is not mandatory, you may choose to include your business telephone number and/or your e-mail address to facilitate contact with the public and the media about your research.

Business telephone no. (optional): 1 (604) 822-0436

E-mail address (optional): rap@cs.ubc.ca

Databases are very complex; a typical database can store data in hundreds of tables and can contain hundreds of references between tables. The layout of the data into tables and references between them is referred to as a schema. Learning the details of these complex schemas requires significant time. This problem is exacerbated by data sources being shared across organizations, and users attempting to make their applications do something for which they were not originally designed. Consider the following two scenarios:

(1) A user is integrating civil engineering data across a number of applications, including CAD models and project scheduling. The data in these applications can be exported via a number of different standards. In order to integrate the data, the user must first choose which standard to export from and then figure out how to understand how the data has been exported.

(2) An end user has access to a database application that generates reports. The end user asks the database programmer to change the application to access more information than was originally accessible. The programmer has never seen the application before, but now must extend the application to access new data. This will require the programmer to understand how the data in the back-end database is stored well enough to write the new queries.

Neither of these scenarios is adequately helped by existing approaches. I propose to improve schema understanding. The goal is to help users in a variety of scenarios including those above to make better use of the existing schemas that they are encountering for the first time, whether it is extending their understanding of a current application, trying to decide how to export their data to a standard schema (which is challenging since related data may be scattered across the standard schema, rendering it incomprehensible), or choosing a standard schema to which the data should adhere.

Other Language Version of Summary (optional).

Personal identification no. (PIN)	Family name of applicant
Valid 290625	Pottinger

Before completing this section, **read the instructions** and consult the *Use of Grant Funds* section of the NSERC Program Guide for Professors concerning the eligibility of expenditures for the direct costs of research and the regulations governing the use of grant funds.

TOTAL PROPOSED EXPENDITURES (Include cash expenditures only)

	Year 1	Year 2	Year 3	Year 4	Year 5
1) Salaries and benefits					
a) Students	43,625	43,625	43,625	43,625	43,625
b) Postdoctoral fellows	0	0	0	0	0
c) Technical/professional assistants	799	799	799	799	799
d)	0	0	0	0	0
2) Equipment or facility					
a) Purchase or rental	2,000	2,000	2,000	2,000	2,000
b) Operation and maintenance costs	184	184	184	184	184
c) User fees	30	30	30	30	30
3) Materials and supplies	0	0	0	0	0
4) Travel					
a) Conferences	4,000	4,000	4,000	4,000	4,000
b) Field work	0	0	0	0	0
c) Collaboration/consultation	0	0	0	0	0
5) Dissemination costs					
a) Publication costs	0	0	0	0	0
b)	0	0	0	0	0
6) Other (specify)					
a)	0	0	0	0	0
b)	0	0	0	0	0
TOTAL PROPOSED EXPENDITURES	50,638	50,638	50,638	50,638	50,638
Total cash contribution from industry (if applicable)					
Total cash contribution from university (if applicable)					
Total cash contribution from other sources (if applicable)	0	0	0	0	0
TOTAL AMOUNT REQUESTED FROM NSERC (transfer to page 1)	50,638	50,638	50,638	50,638	50,638

Budget Justification

The budget presented on page 5 of form 101 can be broken down as follows:

- $38,000 = $19,000 a year for each of two the PhD students who will work on this project. The specific milestones that each student will work on are detailed in the proposal.
- $28,125 = $5,625 * 5 for one summer undergraduate per year to work on aspects of the above problems. $5,625 is the minimum than UBC will allow paying undergraduates for summer research. I hope to secure NSERC USRAs, as I have in the past, although I cannot count on this. Undergraduate Yun Lou successfully worked with me last year. Since he is only a second year student, I hope to have him work with me the next two summers. After that, I will find additional students.
- It is important for professional development for both me and my students to attend conferences. Thus, I anticipate one conference for me and one conference for each PhD student once a year; my costs will be higher due to the higher registration costs and because students can also get partial funding through student volunteer positions and other travel subsidies, so I have budgeted $2000 for a conference for me, and $1000 for a conference for each student.
- My computer is due for replacement in 2011. Because my computer is also used for teaching, the department offsets part of these costs. I expect that the part of the purchase price that is not offset by the department will come to $2000. Because laptops tend to last 3 years before becoming obsolete, I have also budgeted for a new computer in year 4 of the grant. My students can begin by making use of the computers funded by Professor Raymond Ng and Laks Lakshmanan's RTI grant. However, these computers will become obsolete over time as well, so I have budgeted for replacing them in years 2, 3, and 5.
- The UBC Department of Computer Science charges grants for the direct costs of technical support to research projects. This includes installation and support of equipment; technical support for researchers; print, file and network servers; printing and copying; and similar direct costs. The current charge is 0.06% for User Fees, 0.36% for Operation and Maintenance and 1.58% for Technical Professional Assistants.

Relationship to Other Research

I currently have the following sources of research funding:

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

Participants: Staub-French, Booth, Pottinger, and Tory
Funding source: NSERC
Program name: Strategic Grant Program (November 2006 – November 2009)
Hours per month: 25
Budgetary overlap with present application: 0%
Summary: Construction of civil infrastructure depends heavily on *advanced communications and management of information*. Construction projects are characterized by complex one-of-a-kind facilities constructed in uncertain environments under intense schedule constraints. These unique challenges have led to a highly specialized industry that is fragmented, both vertically (between planning, design, and construction phases) and horizontally (across architects, engineers, and subcontractors). This leads to inefficiencies in the design process that have important economic consequences: overlooked design problems often require last-minute construction changes that can be costly and may disrupt the construction schedule. This project develops novel information integration, workflow capture, and interaction techniques to support coordination and communication between professionals in the construction industry. This grant was extended until November, 2010, and will end before this Discovery Grant will start. It is currently funding my PhD student Michael Lawrence. There is no budgetary overlap.

Requirements-Driven Data Warehousing: A Preliminary Proof-of-Concept Study

Participants: Kiringa, Pottinger and Consens
Funding source: NSERC CRD
Program name: Collaborative Research and Development Grant (2009-2010)
Hours per month: 20
Budgetary overlap with present application: 0%
Summary: This project aims at conducting a preliminary proof-of-concept study of a requirements-driven data warehouse construction. Two specific tasks are to be solved: first, design a conceptual integration modeling language for representing typical integration features such as data provenance, data access requirements for materialization, and mappings from the conceptual model to the underlying multidimensional model (which can be expressed as a data cube); second, build a prototype tool that uses models expressed in the conceptual integration modeling language to drive the design and the population of a data warehouse.
Conceptual and budgetary relationships of this project to the proposed research:
There is no conceptual or budgetary overlap with this proposal, and it will end in June of 2011. It is providing funding for one masters student, Charles Zhaohong Chen.

Business Intelligence Network (BIN)

Participants: Miller (Project leader), Pottinger, and 13 other PIs
Funding source: NSERC
Program name: Strategic Network (2009-2014)
Hours per month: 25
Budgetary overlap with present application: 0%
Summary: Complex organizations are in need of knowledge-management solutions that are more comprehensive than the existing patchwork of data and content management systems that these organizations deploy to date. Today, there is an overemphasis on data management and moving data

with the goal of providing meaningful knowledge extraction and integration from complex data. Complex data may be represented in a myriad of different forms within heterogeneous applications. These observations illustrate the need for a business knowledge management architecture where corporate objectives, processes, and data are linked through organization and operational objectives. This web of knowledge is not prescriptive. It is formed by the databases and the systems of an organization, and linked with documents and digital media that have implicit schemas and that can be mapped or linked to data sources to which they relate. The data sources, on the other hand should be able to expose some semantics related to their purpose, history, positioning, provenance, etc., so that meaningful connections can be made. To address these concerns in a meaningful way, we propose to organize our research efforts within the following four themes: strategy and policy management, capitalizing on document assets, adaptive data cleaning and supporting top-down business-driven data integration.

Conceptual and budgetary relationships of this project to the proposed research:

The students that I am funding on this grant are working on several problems including building a data warehouse bottom up and making sense of data that is stored either in large document repositories or on social networks. I am receiving approximately $50,000 per year on this grant; it is currently funding my MSc. students Tianyu Li, Ali Moosavi, and Dibesh Shakya.

Startup funding

Participants: Pottinger

Funding source: University of British Columbia

Budgetary overlap with present application: 0

Summary: This money was presented when I started my faculty position in 2004. I have enough money left to fund my PhD student, Michael Lawrence through the rest of his PhD along with additional TA support from the department.

Improving Schema Understandability for Semantic Integration

1 Overview

Data is overwhelming us at an ever increasing rate. A typical database schema (i.e., tables representing how the data is stored) can have hundreds of tables, and the world's data grows an astounding 60% annually [1]. The constant influx of data makes it hard for its true value to be realized – it is impossible to understand everything that the data can possibly show. The overwhelming nature of the flood of information is exacerbated when we consider that data may need to be combined with data from other sources for maximum usefulness. For example, overlaying the real estate listings for a house with tax, school, and crime data from other sources greatly improves the usefulness of the real estate listing. The growing popularity of mashups speaks to the power of combining data from heterogeneous sources.

When heterogeneous databases are combined, they typically have different schemas. For information to be shared between these databases, there must be some way for differences in representation to be resolved: if information about a neighborhood is stored one way in schema A, and another way in schema B, then for information from both schemas to be combined, there must be some way to describe how data in schema A relates to the data in schema B; this is referred to as *semantic heterogeneity*. Combining these heterogeneous sources so that they can be queried uniformly is known as *semantic integration*, and forms the basis of my research program. There are many aspects to semantic integration, including how to create the underlying system that allows queries to be processed to allowing the user to understand the overpowering amount of data available.

My overall research goal is to focus on increasing data's usefulness through semantic integration. In Section 2, I describe how this is informed by my recent work on semantic heterogeneity across many diverse application types and scenarios, including construction, business, and disaster management.

My current and short term focus is to make the current flood of potentially integrated information more comprehensible and less bewildering. As I describe in Section 3, my goal in this grant is to ameliorate a key problem found in many areas: the schemas that users are trying to integrate are often so complex as to be incomprehensible; users cannot understand how to make their schemas interact. This is true whether the user is trying to extend understanding of a current application, deciding how to export data to a standard schema (which is challenging since related data may be scattered across the standard schema, rendering it incomprehensible), or choosing a standard schema to which the data should adhere.

2 Previous Research Activities and Progress

For the past five years my research focused on managing heterogeneous data interactions, in both generic and specific application-based settings. Focusing on some application-based settings has helped ensure that I am focusing on useful problems that allow the user to make the most of their data.

My primary goal for the previous phase of my ongoing research to resolve semantic heterogeneity was to solve representational issues preventing users from semantically integrating their data – users could not integrate as needed because the tools that they needed were either lacking in formalization or in the ability to handle the complexity for current work. I took motivation from two different interdisciplinary projects where the users had typical data management needs:

- **ARTIFACT** is an ending Strategic Project where computer scientists and civil engineers improved managing the construction of a building. Data flow inefficiencies make the task hard and the overall process inefficient. For example, suppose that a general contractor could save $50,000 by lowering the ceiling 3 cm; the contractor must coordinate with others (e.g., electricians) to determine the impact of this change. Today this is slow, and tedious and error-prone. Industrial consortiums attempted to improve data flow between applications by creating standard XML schemas, but these failed to meet their goals [2.ZWL+11]. This limits the effectiveness of data integration – the standard schemas that are supposed to allow the sources to interact are so complex that they are unusable.

- **JIIRP:** I was also an investigator on an interdisciplinary project to improve disaster management. Part of this project built a simulation system for assessing the impact of a possible disaster (e.g., an earthquake) and assisting in responding when a disaster happens [2.MSV+06].

These two wildly divergent applications revealed similar problems: existing data integration solutions are insufficient to overcome applications' semantic heterogeneity. Below I describe some specific innovations we proposed to address these shortcomings; they are described more in my form 100.

- Existing schema mapping solutions are insufficient for the rich ontologies used in applications. My colleagues, students and I worked with anatomy ontologies [2.MPB04], as well as domain ontologies from building design [2.NSFZ+08], and developed new mapping solutions that handle rich ontological relationships [2.WP08]. We showed how these richer mappings help in integrating (merging) schemas [2.MPB04,2.PB08,2.PB09].
- Peer Data Management Systems (PDMSs) are ad-hoc networks of independent peer databases; each source has its own schema. Student Jian Xu and I showed how to choose which peers to create mappings between to maximize the improvement on query performance [2.XP]; this allows faster creation of data sharing communities, which is crucial to allowing data to flow quickly in situations where time is of the essence, e.g. earthquakes. Additionally, we showed how to process aggregations efficiently using a novel three-role structure answers queries without the user having to know that aggregations are even required [2.XP11a,2.XP11b]. This opens up new possibilities for sharing data without having to understand how the data is organized at other sources.
- Schemas and databases are not static. Current solutions do not help in managing integrations that may be dynamic and evolve over time. Student Michael Lawrence and I, along with civil engineer Sheryl Staub-French, developed a novel system that reflect changes in one database to another database that depends on it [2.LPSF10,2.LPSF11]. This allows users to explore changes that impact both databases more easily, since the coordination is automated.
- Finally, a big unknown is how to manage users' understanding of the schemas that they are trying to interact with and integrate; this is in line with the current trend of making data systems usable [2]. Though current CAD systems allow users to export data to a standard XML schema, we showed that the export is complex enough to be incomprehensible [2.LP07, 2.ZWL+11] – users simply cannot understand their own data. The civil engineers decided that they could not comprehend the representation at all, and that the way to solve this was to create an *ontology* – a representation of the concepts in a domain and the relationships between the concepts – and then create an application that mapped between the sources and the ontology and query based on the ontology [2.NSFZ+08, 2.NZW+09]. Because our work allows users to get the data in the form that they understand rather than the form in which the standard schema exposes it, users can create better buildings because they can explore alternative building designs more easily. This last finding spurred me to my latest direction: increasing the understandability of schemas so that they can be integrated.

3 Planned Research: Approach, Methodology, and Related Work

As motivated by my current and previous work, I will next focus on ensuring that semantic integration is not hampered by incomprehensibility of large schemas. The following scenarios that have arisen in my research illustrate some representative scenarios:

- An end user of a database application asks a programmer to modify the application to access information in the database, but to which the existing application did not provide an interface. The programmer must be able to understand the database schema sufficiently to write the new queries.
- NIEM [3] is a massive XML schema that is supposed to allow law enforcement agencies to better share data that was used in JIIRP. However, the size of the schema and the amount of repetition means that users cannot understand which part of the schema they should map their data to. As a result, many of the benefits of having a standard schema are lost – related data may be stored in so many ways across different organizations that it is impossible to find all of it at once.

- A user is integrating data across applications, e.g., CAD models and project scheduling data. The data can be exported to a number of different standard schemas. To integrate the data, the user must first choose which standard schema to export to and then understand how the data has been exported.

I propose to work on improving schema understanding. The goal is to help users in scenarios similar to those above to better use existing schemas, whether it is extending a current application, deciding how to export data to a standard schema, or choosing a standard schema to which the data should adhere.

These goals are inadequately supported by existing approaches. Some proposed approaches allow users to query without knowing the schema (e.g., [4]). However, as in the above scenarios, many applications require answering semantically deep queries consistently; that will not happen without understanding the schema. Other emerging work is on providing a summary of a schema [5, 6]. While this is helpful for understanding where to begin, it is inadequate for those who need to understand the schema in sufficient depth to write detailed queries. This research consists of five specific objectives:

Objective 1: Assist the user to create a schema or ontology representing the user's understanding of the data (the idealized schema). This idealized schema represents the way that the *user* thinks about the data. If the user is querying an unfamiliar database, this allows the user to express the schema that he or she wants. If the user is trying to understand his or her own data, the idealized schema allows the user to express that as well. If the user is choosing a standard schema to export the data to, the idealized schema allows the user to describe the representation to which the user would like to have that data adhere. The idealized schema does *not* have to describe the entire actual schema; it only describes the user's current interests. This idealized schema can be expanded to solve other needs as they expand. Because the goal is to be data model (e.g., XML, relational, UML) neutral, this must be done generically, so existing tools must be extended. Objective 1 has the following specific milestones:[1]

A. Create a representation of the user schema, ontology, or UML diagram for the idealized schema as well as a storage representation that allows the user to translate between the idealized schema and the actual schema – i.e., an internal matching representation. I am an expert in generic representations, i.e., those coming from Model Management [7, 8]. I will use this knowledge to decide on the representation of the actual and idealized schemas.

B. Develop a tool that allows users to create a schema, ontology or UML to adhere to their representation. Existing tools (see [9] for an overview) need to be extended to work with multiple data models and our generic representation. The focus is on the representation rather than the visualization, so we will use existing techniques for the visualization (see [10] for a survey). Finally, we will also need to create a tool to assist the user to change the idealized schema while ensuring that the underlying representation is updated or that the changes are rejected.

Objective 2: Semi-automatically create a mapping between the idealized and actual schemas. Querying the actual schema through the idealized schema involves creating first a *schema matching* – the set of correspondences between the elements in the two schemas – and then a *schema mapping* – precise relationships necessary to translate data and queries. Both schema matching and schema mapping algorithms must be extended. Because the idealized schema may be an XML schema, relational schema, UML diagram, or ontology (as in the ARTIFACT project above), the system must create matchings over many different data models. Current schema matching is typically done within a single data model, e.g., relational [11], XML, or ontologies [12, 13] . COMA++ [14] is one of the few systems that matches across data models and scales to large schemas. Hence we will build on that work. Existing schema matching techniques generally assume that the input schemas are roughly the same size. E.g., while COMA++ can match schemas of different sizes, their methods are unlikely to generalize well to schemas of wildly varying sizes because it splits larger schemas into fragments; this would inhibit matches across

[1] Note that all objectives include verifying that the objectives are met and that the individual algorithms do as promised, allow users to efficiently execute their tasks, and scale well.

fragment boundaries. There is a strong possibility that the idealized schema will map as a whole to different places in the actual schema (e.g., the classic example of an "address" being mapped to either a "shipping address" or a "billing address"), as well as that the idealized schema will quite possibly be related to elements from a larger part of the actual schema than is typically assumed in schema matching work. Because of this we will focus on the way that users interact with the system. To build the schema matching we intend to build on systems such as Clio [15] and our own HePToX [2.BCH+10]. Objective 2 consists of the following specific milestones:

A. Create a representation of the mapping. We will extend current mapping languages such as the source-to-target tuples generating dependencies used in Clio, and the Datalog-style mappings used in HePToX to our setting. This will mean incorporating ontological relationships into the mapping and ensuring we can capture all features of our generic ideal schema representation
B. Create an algorithm that points out the possible mappings between the actual and idealized schema. This will be an extension of current mapping discovery algorithms.
C. Create a system that allows user interaction to guide which of the possible mappings to choose, possibly building on the work of [16] in choosing which schema merge results to use. The evaluation will be on both real schemas (from the domains of my collaborators including civil engineering) and also on synthetically generated schemas to show the limits of the approach.

Objective 3: Create a method to modify the mapping and/or idealized schemas until they encompass the concepts necessary to answer the required queries. This will involve extending existing techniques for schema evolution and schema mapping. Schema mapping algorithms generally assume that schema mapping is a one shot activity, which is not true in our case. It may be that as the idealized schema grows, elements in the idealized schema are better suited to other matches. As well, the user may want to customize the schema for a different application. For this to happen, the system needs to balance between the "best" fit and wildly veering in different directions, which would confuse the user. This objective includes the following specific milestones:

A. Create a schema repository which maintains the existing schemas and mappings. Otherwise, we will be unable to compare the new mapping with the existing mapping. My ongoing collaborations with scientists across several domains will help to ensure realistic schema repositories.
B. Allow mappings to be extended for new queries to be answered. Existing approaches for evolving mappings react to changes in the matchings or schemas, e.g., Clio's work on updating mappings in [17, 18]), and PROMPT [12], which focuses on user interaction. [19] builds on my MiniCon Algorithm [20] to compose pre-updated mappings between ontologies and relational sources. In contrast, we wish to study how to evolve a mapping when the user requirements lead to new concepts within the ideal schema. We believe past work will be useful but insufficient for this new application.

Objective 4: Create views (i.e., stored queries) to allow the user to query over the idealized schema. There are many new aspects in this objective, including that the idealized schema can be in many data models, and users may not know conventional query languages. I intend to build on model management [7], which focuses on managing metadata in a data-model-independent fashion and has recently yielded the richer mappings necessary to answer queries [21, 22]. This objective has the following milestones:

A. Design and implement a mechanism/query language to allow users to query without knowing a complex query language, ideally without restricting the user to a single data model. This will extend existing work on querying without knowing query languages to decide on, e.g., DISCOVER [23], or MSQ queries [24], though those works assume a single data model. This may build on the work in the NSERC Strategic Network on Business Intelligence that I am involved in. This milestone could easily be sufficient for an additional PhD thesis, in which case fully exploring this milestone would depend on my securing additional funding.
B. Querying will be over the idealized schema. Data will be in the actual schemas. Whether it will require something like view expansion or answering queries using views [25] or a combination of the

two depends on the mapping language in Objective 2. Even if it is expansion only (which is computationally easier), this is non-trivial for non-relational sources. There are algorithms for answering queries using views for relational databases (including my previous work on MiniCon [20] which remains the best algorithm for the problem), and some work has been done on answering queries for subsets of the XPath query language for XML (e.g., [26]) to build on, but answering queries for ontologies and UML requires more work.

Objective 5: Allow users to decide which standard schema better fits their needs. Once users can understand how different actual schemas fit their understanding of the data, they may need guidance to choose an appropriate standard schema. This objective consists of the following milestones:

A. Extend the system in Objective 1 to reuse mapping work in multiple target schemas. This may build on work on matching in schema corpuses [27]). It may also require understanding how the various possible target schemas are related to each other.

B. Allow users to compare the complexity of the mappings, since this may suggest whether the target schema is a better natural fit or not. This will require creating a metric for explaining the complexity of a mapping and showing how the mapping adheres to that metric.

C. Ideally, allow the user to see how data in the idealized schema is treated in candidate schemas.

D. Compare coverage between the possible schemas.

E. Compare complexity of the possible standard schemas. Note that some applications may want more complex schemas than others, so the simplest schema is not always the right solution.

4 Relationship to Other Funding

This grant is one part of my larger program to help users integrate their data and understand the process and the results. My primary additional source of funding is an NSERC Strategic Network Grant on Business Intelligence (BIN). In BIN I am focusing on other aspects of increasing understandability when integrating data: (1) students Ali Moosavi and Tianyu Li and colleague Laks V.S. Lakshmanan and I are working on creating an ontology from social network data (e.g., Delicious) and text data so that the data within can be better understood and shared, and (2) colleague Iluju Kiringa, postdoc Flavio Rizzolo, and a number of students and I are working on how to create a data warehouse top-down [2.RKPW11]. This enables users to understand the representation of the data that they are integrating more effectively than existing bottom-up techniques which build up the representations based on the source schemas.

5 Anticipated Significance of the Work

The problems being addressed in my research have occurred in several variations a number of times in actual practice as mentioned in the motivating scenarios. Solving these objectives would allow users to access more data more flexibly, which would allow them to make sure that they get better access to *all* their data – not just in the manner that they originally anticipated that they would need.

The sample scenarios allow users to get the most out of their data and share it more effectively; there are many more benefits. Without a solution of the type that I propose, data management will be accessible only to the very highly trained. As the amount of data grows, training professionals to re-implement these solutions over and over again (while assuring job security for me) is simply unscalable.

6 Training to Take Place Through the Proposal

The proposed research will provide dissertations for two PhD students. I anticipate that one PhD student will guide the overall representation, and the techniques to express, create, and evaluate the mappings (Milestones 1A, 2A-C, and 3B). A second PhD student will concentrate on the areas more related to user interaction, starting with tools to create, store, and visualize the schema and mapping (Milestones 1B and 3A), then work on the query language (Milestones 4A-B), and will finally work on allowing the user to compare schemas (Milestones 5A-E). Additionally, smaller problems that arise will be investigated as projects for supervised master's students, undergraduates, or students taking my graduate database classes who will benefit from working on real projects.

References Note: references of the form [2.x] are found in the contributions section of my form 100.

[1] Anonymous "All Too Much," *The Economist: Data, Data Everywhere: A Special Report on Managing Information,* pp. 6, 2010.

[2] H. V. Jagadish, A. Chapman, A. Elkiss, M. Jayapandian, Y. Li, A. Nandi and C. Yu, "Making database systems usable," in *SIGMOD,* 2007, pp. 13-24.

[3] Department of Human Services and Department of Justice. National information exchange model (NIEM). Available: http://www.niem.gov.

[4] A. Nandi and H. V. Jagadish, "Qunits: Queried units for database search," in *CIDR,* 2009.

[5] X. Yang, C. Procopiuc and D. Srivastava, "Summarizing relational databases," in *VLDB,* 2009.

[6] C. Yu and H. V. Jagadish, "Schema summarization," in *VLDB,* 2006, pp. 319-330.

[7] P. A. Bernstein, A. Y. Halevy and R. Pottinger, "A Vision of Management of Complex Models," *SIGMOD Record,* vol. 29, pp. 55-63, 2000.

[8] R. Pottinger and P. Bernstein, "Merging models based on given correspondences," in *VLDB,* 2003.

[9] A. Borgida and J. Mylopoulos, "Conceptual schema design," in *Encyclopedia of Database Systems,* 2009, pp. 111-124.

[10] T. Catarci, M. F. Costabile and S. B. Levialdi C., "Visual Query Systems for Databases: A Survey," *Journal of Visual Languages and Computing,* vol. 8, pp. 215-260, 1997.

[11] J. Kang and J. F. Naughton, "On schema matching with opaque column names and data values," in *SIGMOD,* 2003, pp. 205-216.

[12] N. Noy and M. Musen, "PROMPT: Algorithm and tool for ontology merging and alignment," in *AAAI,* 2000, pp. 450-455.

[13] A. Doan, J. Madhavan, P. Domingos and A. Halevy, "Learning to map between ontologies on the semantic web," in *WWW,* 2002, pp. 662-673.

[14] H. H. Do and E. Rahm, "Matching large schemas: Approaches and evaluation," *Information Systems,* vol. 32, pp. 857-885, 2007.

[15] R. Fagin, L. Haas, M. Hernández, R. Miller, L. Popa and Y. Velegrakis, "Clio: Schema mapping creation and data exchange," in *Conceptual Modeling: Foundations and App.*, 2009, pp. 198-236.

[16] L. Chiticariu, P. G. Kolaitis and L. Popa, "Interactive generation of integrated schemas," in *SIGMOD,* 2008, pp. 833-846.

[17] C. Yu and Popa, "Semantic adaptation of schema mappings when schemas evolve," in *VLDB,* 2005.

[18] Y. Velegrakis, R. Miller and Popa, "Mapping adaptation under evolving schemas," in *VLDB,* 2003.

[19] H. Kondylakis and D. Plexousakis, "Enabling ontology evolution in data integration," in *EDBT/ICDT PhD Workshop,* 2010.

[20] R. Pottinger and A. Levy, "A scalable algorithm for answering queries using views," in *VLDB,* 2000, pp. 484-495.

[21] P. A. Bernstein and S. Melnik, "Model management 2.0: Manipulating richer mappings," in *SIGMOD,* 2007, pp. 1-12.

[22] S. Melnik, A. Adya and P. A. Bernstein, "Compiling mappings to bridge applications and databases," in *SIGMOD,* 2007, pp. 461-472.

[23] V. Hristidis and Y. Papakonstantinou, "DISCOVER: Keyword search in relational databases," in *VLDB,* 2002, pp. 670-681.

[24] C. Yu and H. V. Jagadish, "Querying complex structured databases," in *VLDB,* 2007.

[25] A. Y. Halevy, "Answering queries using views: A survey," *The VLDB Journal,* vol. 10, pp. 270-294, December, 2001.

[26] F. Afrati, R. Chirkova, M. Gergatsoulis, B. Kimelfeld, V. Pavlaki and Y. Sagiv, "On rewriting XPath queries using views," in *EDBT,* 2009, pp. 168-179.

[27] J. Madhavan, P. Bernstein, A. Doan and A. Halevy, "Corpus-based schema matching," in *ICDE,* 2005.