Knowledge Integration to Overcome Ontological Heterogeneity: Challenges from Financial Information Systems

Aykut Firat, Stuart Madnick, Benjamin Grosof
MIT Sloan School of Management
Cambridge, MA USA
aykut@mit.edu, smadnick@mit.edu, bgrosof@mit.edu

Abstract

The shift towards global networking brings with it many opportunities and challenges. In this paper, we discuss key technologies in achieving global semantic interoperability among heterogeneous information systems, including both traditional and web data sources. In particular, we focus on the importance of this capability and technologies we have designed to overcome ontological heterogeneity, a common type of disparity in financial information systems.

Our approach to representing and reasoning with ontological heterogeneities in data sources is an extension of the Context Interchange (COIN) framework, a mediator-based approach for achieving semantic interoperability among heterogeneous sources and receivers. We also analyze the issue of ontological heterogeneity in the context of source-selection, and offer a declarative solution that combines symbolic solvers and mixed integer programming techniques in a constraint logic-programming framework. Finally, we discuss how these techniques can be coupled with emerging Semantic Web related technologies and standards such as Web-Services, DAML+OIL, and RuleML, to offer scalable solutions for global semantic interoperability.

We believe that the synergy of database integration and Semantic Web research can make significant contributions to the financial knowledge integration problem, which has implications in financial services, and many other e-business tasks.

Keywords: Database Integration, Semantic Web, Ontologies, and Source Selection

1. Introduction

The shift towards global networking brings with it the challenge of achieving global semantic interoperability among heterogeneous computer systems. In this paper, we discuss the importance of these capabilities and key technologies in achieving semantic interoperability among traditional and web data sources by referring to examples from financial information systems. In particular, we focus on technologies we have designed to overcome ontological heterogeneity, a common type of variation in financial information systems.

Our approach to representing and reasoning with ontological heterogeneities in data sources is an extension of the Context Interchange (COIN) framework, a mediator-based approach for achieving semantic interoperability among heterogeneous sources and receivers. The extended COIN (ECOIN) system's capability to handle both *data level* and *ontological* heterogeneities in a single framework makes it unique among other major database integration efforts. We will explain these two types of heterogeneities in the next section.

We also analyze the implications of ontological heterogeneity in the context of source selection, particularly in the problem of "answering queries using views"(Ullman 1997), which focuses on designing algorithms for realizing query rewriting with the goal of identifying the relevant information sources that must be accessed to answer a query. Our declarative solution to source selection with ontological heterogeneities combines symbolic solvers and mixed integer programming techniques in a constraint logic programming framework.

Finally, we discuss how these techniques can be coupled with emerging Semantic Web related technologies and standards such as Web-Services[1], DAML+OIL[2] and RuleML[3] to offer scalable solutions for global semantic interoperability.

In the next section, we start with a taxonomy of heterogeneities in financial data sources to motivate the financial knowledge integration problem. Before explaining our extended framework, we provide background information on COIN in section 3. Implications of ontological heterogeneities in the source selection problem follow the extended framework. Finally, we provide our vision of combining the ECOIN approach with Semantic Web technologies to facilitate even further the global interoperability among financial data sources.

2. Taxonomy of Heterogeneities in Financial Databases

After noticing puzzling differences in reported financial data belonging to same companies across different financial data sources, we conducted several investigations into the nature of these variations. We

[1] http://www.w3.org/2002/ws/
[2] http://www.daml.org -> DAML+OIL
[3] http://www.dfki.de/ruleml and http://www.mit.edu/~bgrosof/#RuleML

examined Primark Investment Research Center's Worldscope, DataStream and Disclosure databases and data definition manuals as well as Security Exchange Commission (SEC) company filings, and several other web-based financial sources (including Hoovers, Yahoo, Market Guide, Money Central, and Corporate Information.)

For example, we compared *Net Sales, Net Income, Total Assets, Number of Employees, and Five-Year Growth in Earnings per Share* accounting data items for a given company across these data sources and found significant variations. In Table 1, variations between Disclosure and Worldscope databases range from 4 to 92 percent for these five accounting data items for the same set of companies.

ACCOUNTING DATA ITEMS	% VARIATIONS
Net Sales	20
Net Income	20
Total Assets	4
Number of Employees	40
Five-Year Earnings Growth per Share	92

Table 1. Variations between Disclosure and Worldscope databases

We reviewed our findings with Primark representatives to discover that variations could be attributed to different reporting standards, namely data item definitions and representations, used by different databases. Different types of users prefer to view company financial data in different ways depending on their job functions. Based on those preferences, data providers usually provide financial data in one or more of the following ways: **As presented** by the company (data provided by SEC and similar foreign agencies)

1. **As reported** (data modified to fit a standard attribute naming convention)
2. **In local format** (data fits local accounting practices)
3. **Standardized** (data modified based on the knowledge of industry and extensive research in order to allow for meaningful performance analysis).

Fund managers, for instance, use "standardized data" to obtain a quick graphical representation of company's performance; and financial analysts often use "as presented data" for an in-depth analysis of a given company.

Because of these preferences, and local adaptations of data, several types of heterogeneity exist in financial data sources. Below, we elaborate on three types of heterogeneities: data-level, ontological and temporal, with most emphasis on ontological heterogeneities, which is the focus of this paper.

2.1. Data-Level Heterogeneity

We observe this type of heterogeneity when databases reporting on the same *entity* adopt different representation choices, which are not fully specified in the *entity type definitions*. Financial databases reporting on the same companies, for example, often show differences in the way they represent financial items, usually in terms of units, scale factors, and/or formats although they may agree on the definitions of these items when they do not enforce a specific scale, unit or format. In Table 2, the Corporate Information web site reports *sales* data without any scaling, while the Market Guide web site reports *sales* data in millions. Similarly, currencies used by Corporate Information and other databases are also different. While Corporate Information reports sales data in the local currency of the company, the others always report the values in US dollars. We classify this type of heterogeneity that arises when the same entity is represented differently in different contexts as *data-level heterogeneity*.

COMPANY/ DATASOURCE	FIAT	DAIMLER CHRYSLER BENZ
Hoovers	48,741.0 (Dec99)	152,446.0
Yahoo	N/A	131.4B
Market Guide	45,871.5 (Dec99)	145,076.4
Money Central	49,274.6 ('99)	152.4 Bil
Corporate Information	57,603,000,000 ('00)	162,384,000,000 ('00)
World Scope	93,719,340,540 (99)	257,743,189 (98)
Disclosure	48,402,000 (99)	131,782,000 (98)
Primark Review	51,264 (99)	71354 (97)

Table 2. Sales data for Fiat and Daimler Chrysler across different data sources

2.2. Ontological Heterogeneity

We observe this type of heterogeneity when databases differ on *entity type definitions and/or relationships between entities*. In particular, the majority of definitional variations could be attributed to the inclusion or exclusion of various accounting items such as *Depreciation and Amortization, Excise Taxes, Earnings from Equity Interests,* and *Other Revenue*

from the financial data items. Similarly, variations in *Total number of Employees* could be attributed to inclusion or exclusion of *Temporary Employees*, *Employees of Subsidiaries* as well as the time of reporting. In addition, some of the variations in *5-Year Earnings Growth per Share* numbers could be attributed to the lack of accounting for fluctuations in foreign currency.

Despite having differing definitions, entities can usually be related to each other when one or more entities uniquely determine the value of one or more other entities. For example, for certain companies, the *Pretax Income* can be derived from *Pre-tax Profit* and *Assoc. Pre-tax Profit* attributes in another, as shown below:

"Worldscope. Pretax Income"= "Datastream. Pre-tax Profit" – "Datastream. Assoc. Pre-tax Profit"

More broadly, entities are not only related through a formula that produces one entity out of several other entities across or within a source, but also through more elaborate logic. For example when converting a financial figure in a database from local currency into US Dollars, its value may have to be derived by first figuring out which company it belongs to, which country the company is incorporated in, and which date corresponds to this financial figure, then using the appropriate exchange rate to perform the conversion.

Ontological heterogeneities also arise when ontologies have structural differences similar to cases studied under schema integration (Batini et. al. 1986), and when ontologies are expressed using different ontology languages. While our focus in this paper will be more on the definitional variations explained above, our framework can be used to address the structural and syntactical ontological heterogeneities as well.

2.3 Temporal Heterogeneity

Temporal heterogeneity arises when entity values or definitions belong to different times, or time intervals. In Table 2, for example, sales numbers for companies are reported for different years. Financial data are also aggregated over different time intervals, and often reported quarterly or annually. Temporal heterogeneity is usually orthogonal to data level and ontological heterogeneities and observed in mixed forms. Definitions of data terms, for example, may change over time as seen in the example below. The three-way dependency between the Worldscope, Disclosure, and SEC databases for Exxon is different before and after 1996.

For Exxon after 1996:

"Worldscope. Revenues" = "Disclosure. Net Sales" –" SEC. Earnings from Equity Interests and Other Revenue" –" SEC. Excise Taxes"

For Exxon before 1996:

"Worldscope. Revenues" = "Disclosure. Net Sales" – "SEC. Excise Taxes"

We have described three types of heterogeneities that are widespread in financial information sources. We do not claim that these three cover all types of heterogeneities that exist in financial databases, yet understanding the properties of these heterogeneities becomes critical when one attempts to integrate disparate financial databases. In the next section we describe the Context Interchange (COIN) framework, our core approach to financial data integration that successfully handles *data-level heterogeneities*. We then explain how we extend the COIN framework to handle *ontological heterogeneities*.

3. The Context Interchange Approach to Financial Data Integration

The COntext INterchange (COIN) framework is a mediator-based approach for achieving semantic interoperability among heterogeneous information sources. The approach has been detailed in (Goh et al. 1999) and a prototype has been developed to demonstrate the feasibility of the approach. The overall COIN approach includes not only the mediation infrastructure and services, but also wrapping technology and middleware services for accessing source information and facilitating the integration of the mediated results into end-users' applications.

The set of Context Mediation Services comprises a Context Mediator, a Query Optimizer, and a Query Executioner. The Context Mediator is in charge of the identification and resolution of potential semantic conflicts induced by a query. This automatic detection and reconciliation of conflicts present in different information sources is made possible by general knowledge of the underlying application domain, as well as the informational content and implicit assumptions associated with the receivers and sources. These bodies of declarative knowledge are represented in the form of a domain model, a set of elevation axioms, and a set of context theories, respectively. The result of the mediation is a mediated query. To retrieve the data from the disparate information sources, the mediated query is transformed into a query execution plan, which is optimized, taking into account the topology of the network of sources and their capabilities. The plan is then executed to retrieve the

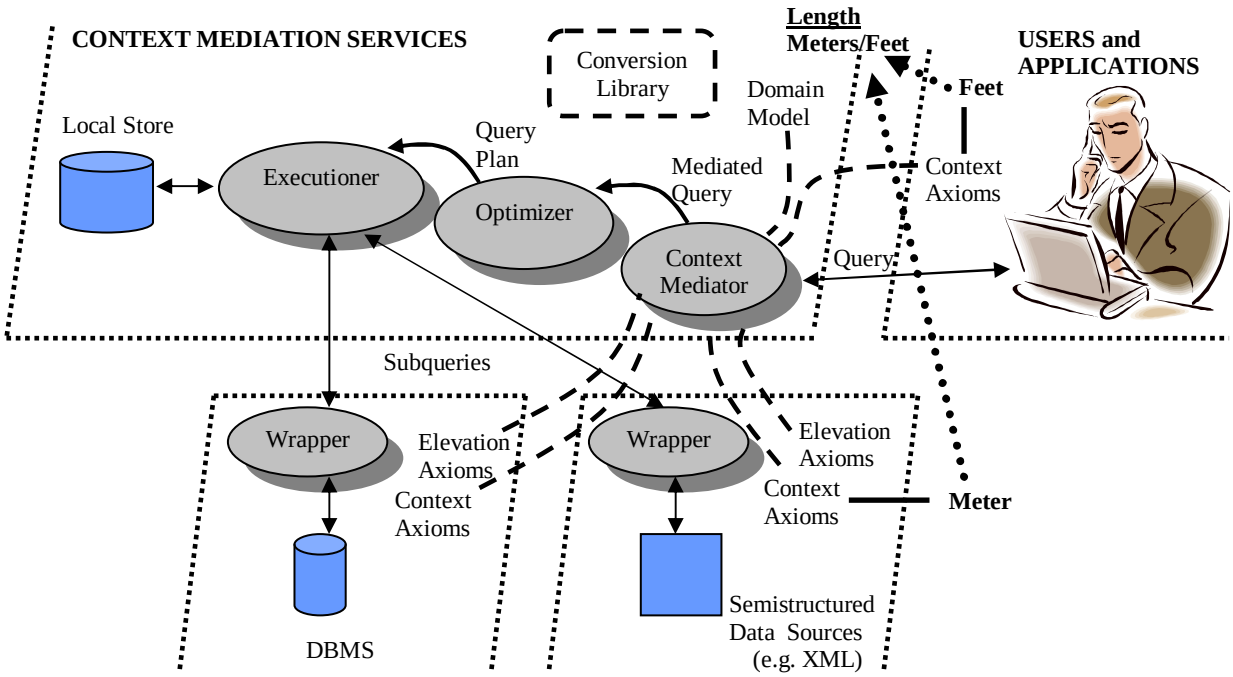

Figure 1. The Architecture Of The Context Interchange System

data from the various sources; results are composed as a message, and sent to the receiver.

The knowledge needed for integration is formally modeled in a COIN framework as shown in Figure 1. The COIN framework comprises a data model and a language, called COINL, of the Frame-Logic (F-Logic) family (Kifer et al. 1995; Dobbie and Topor 1995). The framework is used to define the different elements needed to implement the strategy in a given application:

- The Domain Model/Ontology is a collection of rich types (*semantic types, attributes, etc.*) and relationships (*is-a relationship*) defining the domain of discourse for the integration strategy.
- Elevation Axioms for each source identify the semantic objects (instances of semantic types) corresponding to source data elements and define integrity constraints specifying general properties of the sources;
- Context Definitions define the different interpretations of the semantic objects in the different sources or from a receiver's point of view. Special types of attributes, modifiers, are used to define the context of a data type. For example scale-factor and currency modifiers may define the context of objects of semantic type "profit" when they are instantiated in a context (i.e. scale-factor = 1000 & currency = USD)

Finally, there is a conversion library, which provides conversion functions for each *modifier* to define the resolution of potential conflicts. The conversion functions can be defined in COINL or can use external services or external procedures. The relevant conversion functions are gathered and composed during mediation to resolve the conflicts. No global or exhaustive pair-wise definition of the conflict resolution procedures is needed.

Both the query to be mediated and the COINL program are combined into a definite logic program (a set of Horn clauses)(Baral et al. 1994) where the translation of the query is a goal. The mediation is performed by an abductive procedure, which infers from the query and the COINL programs a reformulation of the initial query in the terms of the component sources. The abductive procedure makes use of the integrity constraints in a constraint propagation phase, to accomplish semantic query optimization. For instance, logically inconsistent rewritten queries are rejected, rewritten queries containing redundant information are simplified, and rewritten queries are augmented with auxiliary information. The procedure itself is inspired by the Abductive Logic Programming framework (Kakas et al. 1993) and can be qualified as an abduction procedure. One of the main advantages of the abductive

logic-programming framework is the simplicity in which it can be used to formally combine and to implement features of query processing, semantic query optimization and constraint programming.

COIN framework elegantly addresses *data-level* heterogeneities among data sources expressed in terms of context axioms. In the next section, we explain how we extend the COIN framework to handle *ontological heterogeneities* in addition to *data-level* ones, to enhance its capabilities, and to provide a more complete approach to knowledge integration.

4. The Extended COIN Approach

The original COIN framework can handle data level heterogeneities, but did not provide a solution for ontological heterogeneities, which are quite common in financial databases (as mentioned in Section 2). Even the widely known concept "*profit*", for example, may take many meanings depending on whether it includes tax or not, whether one time items are included or excluded, etc. Before explaining our approach to handling ontological heterogeneities we briefly explain two approaches to modeling data sources by using a shared ontology, approaches which have direct relevance to the issue of ontological heterogeneity. (Levy 2000)

4.1. 'Global as View' and 'Local as View' Approaches to Ontological Heterogeneity

Both of these are ontology-based approaches to heterogeneous database integration problem. The major difference between the Global as View (GAV), and the Local as View (LAV) approaches is on how they relate sources and ontologies. The shared ontology is defined as a view over data sources in GAV. The reverse approach is taken in LAV, where data sources are defined as views over the shared schema. In Table 3, we illustrate, in prolog notation, how these two approaches would model the difference in the meaning of the term "*profit*" when it shows variations across sources based on the inclusion or exclusion of tax.

DATA SOURCE	GLOBAL AS VIEW	LOCAL AS VIEW
Edgar: profit means after-tax-profit	after-tax-profit(company, profit) :- edgar(company, profit).	edgar(company, profit):- after-tax-profit(company, profit).
Quicken: profit means before-tax-profit	after-tax-profit(company, profit) :- quicken(company, before_tax_profit), yahoo(company, tax), profit is before_tax_profit - tax.	quicken(company, before_tax_profit):- after-tax-profit(company, profit), tax(company, tax), before_tax_profit is profit + tax.

Table 3. How GAV and LAV would represent the ontological heterogeneity concerning "*profit*"

As shown in the first column, Edgar, a web based data source, provides after-tax-profits while the Quicken data source provides before-tax-profits. In both approaches it is assumed that the shared ontology has pre-defined semantics, and we arbitrarily decided that the *profit* concept in the shared ontology refers to after-tax-profit in our example.

In the GAV approach, the *profit* term in data source Edgar is easily mapped to the shared ontology concept *profit* since their meaning is the same. For Quicken, which provides before-tax profit, a third source, which can provide the tax amounts corresponding to each country name, is needed to define the mapping shown in Table 3 second row. There are, however, two important problems with finding a third source that provides the tax information. First, this source may not be internally available, thus may require a manual search to locate an external source. Second there may not be a single source that provides the tax information for all the companies in Quicken.

The LAV approach avoids both problems mentioned above for the GAV approach. As long as the terms in the shared ontology can be combined in a way that describes the source contents, there is no need to perform a manual external search to find data sources that provide tax information. Although the LAV approach avoids the problems of GAV, it suffers from problems related to computational tractability. For example, the problem of finding source combinations modeled as views over a shared schema to answer a given query (known as "answering queries using views") has been shown to be NP-complete. Algorithms that scale well for hundreds of sources, however, have been developed (Pottinger and Halevy 2001) and may be adequate in most practical cases. We are not aware, however, of any work done in "answering queries using views" that considers the issue of ontological heterogeneity and reasoning using equations that hold between ontology terms. The closest related work in this area can be found in (Duschka 1997), with a particular focus on functional dependencies. His analysis, however, is not extended to equations holding between heterogeneous ontological terms. (Levy 2000) also points out the need for more research on the query formulation problem in cases where the mediated schema and/or the data source

schemas are described using a richer knowledge representation language.

4.2 Extended COIN Approach to Ontological Heterogeneity

In the Extended COIN framework, unlike the two approaches explained above, we do not assume that the shared ontology has any pre-defined semantics independent of context, but instead allow the shared ontology to assume different meanings in different contexts. Thus, in Table 4, both Edgar and Quicken profit items map to the same ontology concept profit, but take different modifier values that affect the interpretation of profit in their contexts.

In ECOIN framework, we introduce a meta-ontology layer in order to allow ontology elements to assume different semantics in different contexts. The building blocks of the COIN ontology layer--*semantic types*, *attributes*, *modifiers*, and *is-a relationships*--become instances of the meta-ontology-layer types as shown in Figure 2. Every type in the meta-ontology layer inherits from the root-type Frame, which may be thought as an abstract type specification for ontology elements. *Modifiers* attached to Frames determine how a *semantic type*, *attribute*, *modifier*, and *is-a* relationship are represented and interpreted in a given context. In Figure 2, we show the *type modifier* attached to the Frame meta-semantic type. *Type modifier*, depending on the value it takes in a context, determines how Frame objects are to be interpreted in that context semantically. For example, type modifier for "profit" has an "after-tax" value in edgar context, and "before-tax" value in quicken context.

DATA SOURCE	ECOIN
Edgar: profit means after-tax-profit	profit(company, profit) :- edgar(company, profit). $modifier_{edgar}$(profit) = after-tax-profit
Quicken: profit means before-tax-profit	profit(company, profit) :- quicken(company, profit). $modifier_{quicken}$(profit) = before-tax-profit

Table 4. How ECOIN would represent the ontological heterogeneity concerning "*profit*"

This modifier based approach brings an important flexibility: the ability to deal with ontological conflicts without making changes to the existing ontology, for example by introducing new types and defining equational relationships between their values. Making changes to ontologies is likely to be a time-consuming and rigorous process, and is better avoided, or its time

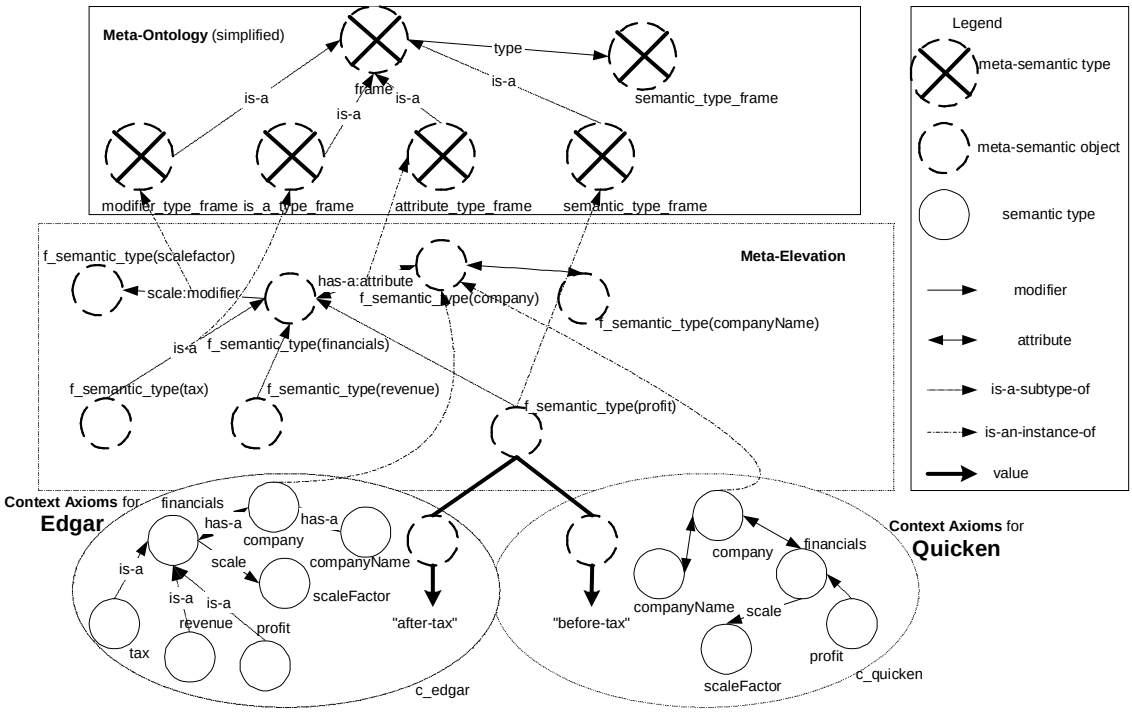

Figure 2. A Graphical Illustration of the Ontology Level Components of ECOIN

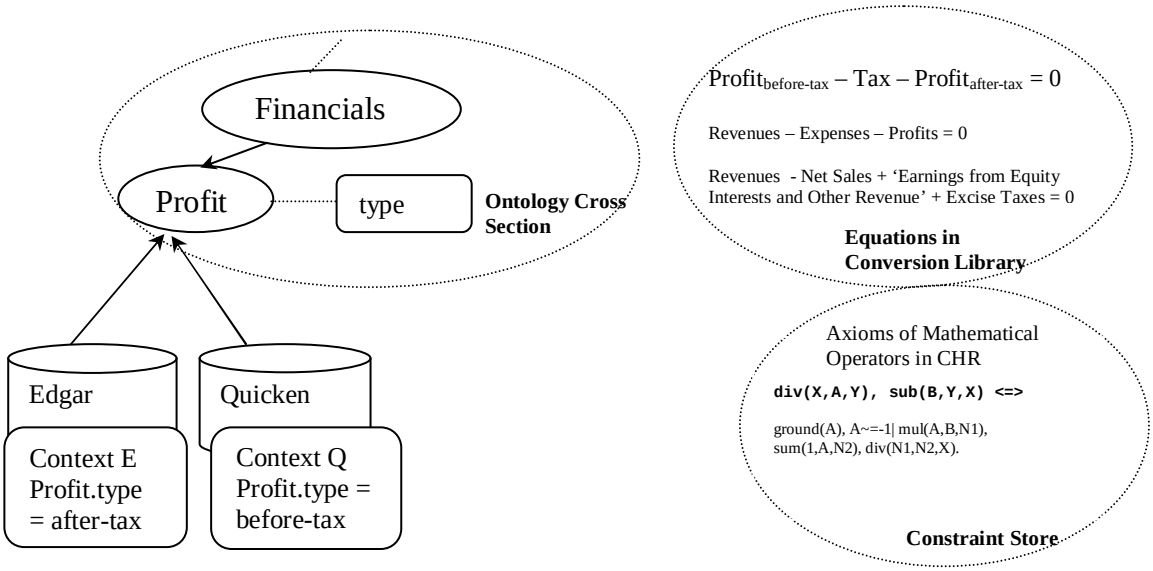

Figure 3. Ontological Conflict Handling in ECOIN

optimally determined.

In the ECOIN framework, difference in the interpretation of semantic types is automatically detected by comparing their modifier objects. Consequently, the detected conflicts are resolved by automatically applying appropriate conversion functions and symbolic equation solving techniques through the use of *constraint handling rules (CHR)*, a high-level language extension of *Constraint logic programming(CLP)*, especially designed for writing constraint solvers. CHR combines the advantages of logic programming and constraint solving by providing a declarative approach to solving problems, while at the same allowing users to employ special purpose algorithms in the sub problems. The constraint solver works by repeatedly applying constraint rules, rewriting constraints into simpler ones until they are solved. It has been used to encode a wide range of problems, including ones involving terminological and temporal reasoning.

This process is illustrated in a simplified drawing in Figure 3. On the right hand side, in Figure 3, the conversion library including the profit equations necessary to convert between different interpretations and the constraint store that handles symbolic equation solving are shown. Queries involving ontological conflicts are rewritten using the conversion libraries and the constraint handling rules specified in the constraint store.

Our approach not only deals with the semantic aspects of disparities, but also allows ontologies to be represented in different syntaxes with the definition of appropriate modifiers. This is quite useful, for example, when ontologies are interchanged using DAML+OIL or KIF, which are two different representation languages for ontologies. In the meta-ontology layer, we also have Frames that correspond to attribute, modifier, and is-a kind of relationships used in COIN framework ontologies, but we limited our discussion to semantic type Frames in this paper.

The ECOIN framework offers a complete solution to data integration problems concerning the two important aspects of data heterogeneities presented in section 2. A prototype has been developed that demonstrates the practicability of the ideas explained in the previous sections. Next we briefly discuss the issue of ontological heterogeneities in the context of source selection, before explaining how our extended framework can be used in the Semantic Web context.

5. Source Selection and Ontological Heterogeneities

In an emerging class of integration strategies (Levy et al. 1996; Ullman 1997), queries are formulated on ontologies without specifying a priori what information sources are relevant for the query. The information mediator undertakes the task of selecting sources that can satisfy a given query. This problem has been studied mainly from two distinct but related perspectives: finding a query formulation that can at least provide a partial answer to the original query (Levy 2000); and optimizing a certain criterion related to the query or data sources, such as the cost of executing a query or a threshold imposed on a set of data quality parameters

(Mihaila et al. 2001). In all of these approaches custom search algorithms have been developed including the Bucket Algorithm (Levy et al. 1996), the Inverse Rules Algorithm (Duschka 1997) and finally the MiniCon Algorithm (Pottinger and Halevy 2001).

None of these approaches, however, considers the source selection problem in the presence of ontological heterogeneities. We have decided to use constraint logic programming (CLP), a framework that elegantly combines Artificial Intelligence and Operations Research search techniques in a single platform. We represent the source selection problem as a mixed-integer programming model appended with logical constraints, whose solution provides source combinations that can answer a query formulated on an ontology. By using the flexibility of CLP, we integrate a symbolic solver into the problem solution, and reason with the equations that hold between ontology terms. The high level architecture of our mediator is shown in Figure 4.

The CLP framework also allows us to elegantly combine data quality and query reformulation perspectives in a single framework. Quality of data constraints are appended simply as additional constraints into our constraint solver. The combination of the two perspectives brings an additional benefit since it reduces the search space by increasing the number of constraints in the system.

Our approach to source selection is promising in that it translates the source selection problem into a mixed-integer programming problem; therefore we can use the techniques already developed in Operations Research. Because of space limitations, we will leave the details of the model to a more focused paper, and simply say that the model is dynamically generated for a given query, and it includes binary variables assigned to each source indicating whether the source is selected or not; a set of key variable constraints that are needed for joining data elements from various data sources; a set of capability and binding constraints on sources that determine what attributes have to be bound in a given query; a set of data quality constraints; and a set of logical constraints that checks whether the selected sources can satisfy the given query with the help of the symbolic solver that determines the term coverage of a combination of sources.

6. ECOIN and the Semantic Web

Semantic Web research addresses several issues that arose with the ubiquity of the Internet and is laying out some of the infrastructure for intelligent integration of information by adapting and introducing tools for knowledge representation (Berners-Lee et. al 2001). These tools, however, usually attract so much hype that they are perceived as a panacea to all existing IT problems. XML, for instance, has been touted as a complete cure for the data integration problem, whereas it faces many of the same challenges that plagued Electronic Data Interchange (EDI) and database integration efforts of the past (Madnick 2001).

Semantic Web related technologies and standards are promising steps toward achieving global semantic operability. These include in particular:

1) Web-Services (a set of standards for distributed computing),
2) DAML+OIL (an emerging ontology language for the web, based on Resource Description Framework (RDF)),
3) RuleML (an XML encoded knowledge

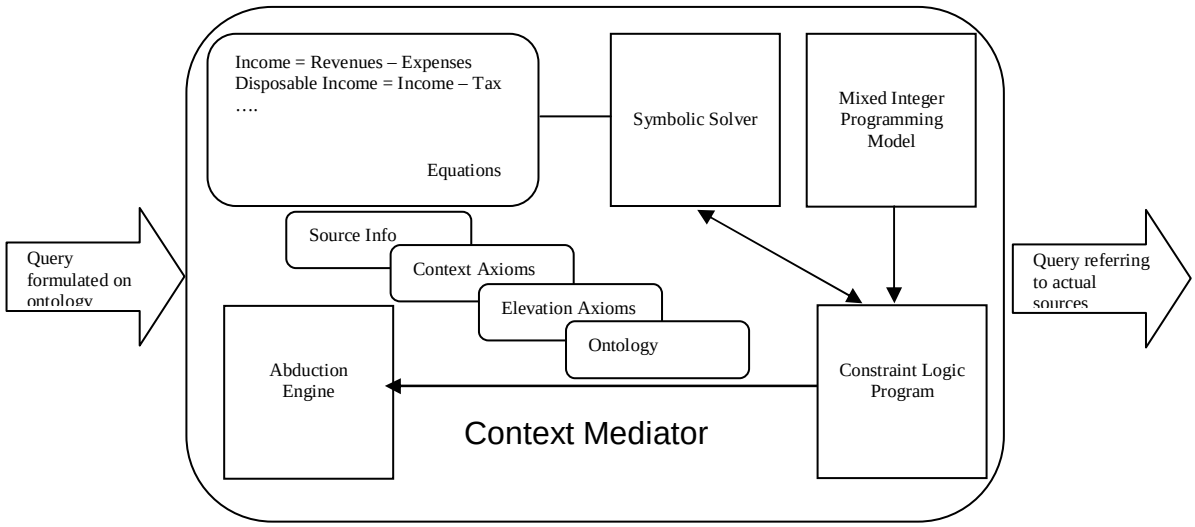

Figure 4. High Level Architecture of Context Mediator with Source Selection Capability

representation for rules).

We believe that the experience and insights gained in database integration research can find direct application in the Semantic Web domain. In this section we provide our architecture for enhancing the ECOIN framework to use these three Semantic Web technologies.

In this architecture, shown in Figure 5, we define each component of a context mediation system (see Figure 1) as a web service. By doing that we build a distributed architecture, with each component having a well defined programming interface in accordance with web services specifications. In this architecture, Context Mediation Registry Services plays an essential role of keeping information about the sources and their relevant axioms. This is similar to the Universal Description, Discovery and Integration (UDDI) registry, but instead stores or provides pointers to context, elevation, ontology axioms, etc. We use XSLT to transform axioms in local format to RuleML, which is used to exchange these axioms between different web services. RuleML is appropriate for this task not only because these axioms can easily be represented in terms of rules, but also because RuleML provides a neutral syntax that can be used both by forward and backward chaining implementations of ECOIN services. This makes it easier for programmers to develop context mediation services by separating knowledge from implementation details. We expect that DAML+OIL will be one of the multiple ontology representation frameworks, along with others, such as KIF. The ECOIN framework will play a crucial role in reasoning with syntactically and semantically disparate ontologies. Our architecture also accommodates legacy systems, or sources not complying with the ECOIN Mediation Services standards with the external provision of required axioms. Conversion libraries are adopted as web services, which may be handled by specialized third parties.

7. Conclusions

Our research into the nature of heterogeneities in financial information systems reveals that *data-level, ontological, and temporal* conflicts are quite common in these systems. Information technologies that can detect

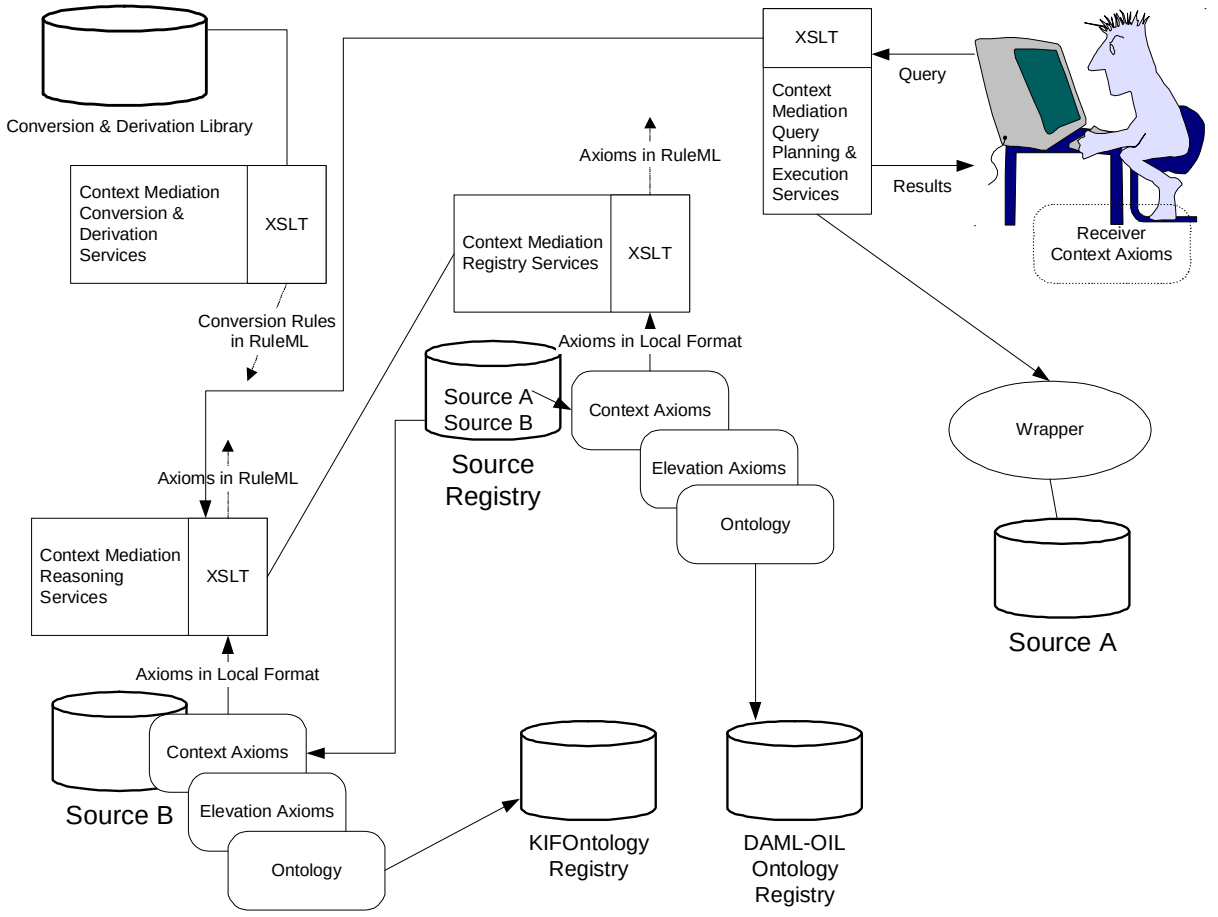

Figure 5. COIN Architecture Applied to Semantic Web

and reconcile these conflicts will be crucial in achieving global semantic interoperability among heterogeneous financial information systems. Our ECOIN framework provides an elegant solution to both data-level and ontological conflicts with its logic-based declarative framework. The use of modifiers is a novel way of representing ontological heterogeneities and eliminates the need of making frequent changes to the ontology, which is known to be a tedious and costly process. ECOIN also provides a clean framework for the source selection problem in the presence of ontological heterogeneities, and equations that relate heterogeneous terms to each other. It combines constraint logic programming framework with mixed integer programming models and symbolic equation solvers, to address the source-selection problem in a single constraint based framework. Our prototype is available in our web site (http://context2.mit.edu/coin/) with several interesting applications. We believe that database integration research has important insights to offer to the parallel Semantic Web research and are currently in the process of adapting our technology into the Semantic Web domain.

Acknowledgements

Work reported herein was supported, in part, by MITRE Corp., Suruga Bank, and Singapore-MIT Alliance.

References

Ambite, J. L., Knoblock, C., Muslea, I., and Philpot, A. "Compiling Source Descriptions for Efficient and Flexible Information Integration," *Journal of Intelligent Information Systems* (16:2), 2001, pp. 149-187.

Baral, C., and Gelfond, M. "Logic Programming and Knowledge Representation," *Journal of Logic Programming* (19), 1994, pp. 74-148.

Batini, C., Lenzerin, M., and Navathe, S. B. "A Comparative Analysis of Methodologies for Database Schema Integration," *ACM Computing Surveys* (18:December), 1986, pp. 323-364.

Berners-Lee T, Hendler, J. and Lassila, O. "The Semantic Web," Scientific American(284:5), 2001, pp. 34-43.

Dobbie, G. and Topor, R. "On the declarative and procedural semantics of deductive object-oriented systems," *Journal of Intelligent Information Systems* (4), pp. 193-219.

Duschka, O.M. "Query Planning and Optimization in Information Integration," STAN-CS-TR-97-1598, *Stanford University Computer Science Technical Report*, Stanford, CA, 1997.

Goh, C. H., Bressan, S., Madnick, S., and Siegel, M., "Context Interchange: New Features and Formalisms for the Intelligent Integration of Information," *ACM Transactions on Office Information Systems*(17:3), 1999, pp. 270-293.

Grosof, B.N., Labrou, Y., and Chan, H.Y. "A Declarative Approach to Business Rules in Contracts: Courteous Logic Programs in XML," in *Proceedings of 1st ACM Conf. on E-Commerce*, Denver, Colorado, 1999, pp. 68-77.

Kakas, A.C., Kowalski, R.A., and Toni, F. "Abductive logic programming," *Journal of Logic and Computation* (2:6), 1993, pp. 719-770.

Kifer, M., Lausen, G., and Wu, J. "Logical foundations of object-oriented and frame-based languages," *Journal of the ACM* (4), 1995, pp. 741-843.

Levy, A.Y. Logic-Based Techniques in Data Integration. *in Logic Based Artificial Intelligence*, Jack Minker (ed.)., Kluwer Academic Publishers, Boston/Dordrecht/London, 2000, pp. 575-595.

Levy, A., Rajaraman, A, Ordille, J. "Querying Heterogeneous Information Sources Using Source Descriptions," *Proceedings of the Twenty-second International Conference on Very Large Databases*, Los Altos, CA, 1996, pp. 252-262.

Madnick, S. "The Misguided Silver Bullet: What XML will and will NOT do to help Information Integration," *Proceedings of the Third International Conference on Information Integration and Web-based Applications and Services*, Linz, Austria, 2001, pp. 61-72.

Mihaila, G., Raschid, L., Vidal., M. E. "Source Selection and Ranking in the WebSemantics Architecture Using Quality of Data Metadata", *Advances in Computers* (55), 2001, pp. 87-118.

Pottinger R., Halevy, A. Y. "MiniCon: A Scalable Algorithm for Answering Queries Using Views," *VLDB Journal*(10), 2001, pp. 182-198.

Reeves, D.M., Wellman, M.P., and Grosof, B.N. "Automated Negotiation from Declarative Contract Descriptions," *Proceedings of the Fifth International Conference on Autonomous Agents*, Montreal, Quebec, Canada, 2001 , pp. 51-58.

Ullman, J. D. "Information integration using logical views," in *Proceedings of the 6th International Conference on Database Theory*, Delphi, Greece, 1997, pp. 19–40.