

Corporate Householding: Framework, Applications Areas and Knowledge Processor

Stuart Madnick
Richard Wang
Angela Xiang Xian

Working Paper CISL# 2003-03

June 2003

Composite Information Systems Laboratory (CISL)
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02142

This page is blank

Corporate Householding: Framework, Applications Areas and Knowledge Processor

Abstract: Previous research in Corporate Householding has defined categories of problems and application areas in various business domains. In this paper, we first describe our more recent results from studying corporate householding applications and knowledge exploration. Then we outline a technical approach to solve an important type of corporate householding problem – entity aggregation – using a motivational example in account consolidation. The proposed technical solution uses and expands the COntext INterchange (COIN) technology to manage and process corporate householding knowledge.

Keywords: Corporate Household, Corporate Householding, Data Quality, Context Mediation, COntext INterchange, Enterprise Knowledge Management, Database Interoperability

1 Introduction

Today’s business environment evolves rapidly. Corporate group structures and the relationships between corporate entities are becoming increasingly complex and difficult to understand. Effective use of knowledge about corporate structures and relationships has come to be an important issue in designing corporations’ strategies and performing business functions. Well-understood and managed information about a corporation’s internal structure as well as its relationships with other business entities can bring significant comparative advantage to its business. Analogous to a family household, a *Corporate Household* is defined as “a group of business units united or regarded united with the corporation, such as suppliers and customers whose relationships with the corporation must be captured, managed, and applied for various purposes” [8]. We also define *Corporate Household Knowledge* to be the actionable knowledge about organizations and related internal and external relationships. The process of capturing, analyzing, understanding and managing corporate household knowledge is known as *Corporate Householding* [9].

Context plays a large role in deciding how corporate household knowledge should be understood. For example, in order to answer the question “how many employees does IBM have?”, we have to consider the purpose of the question, in other words, the context in which the question is asked. Following common practices in their respective fields, an insurance company and IBM’s internal staff may come up with completely different answers. Most likely, this is because they have chosen to aggregate the employee counts from groups of entities within this giant corporation in different ways. If we can capture the corporate householding process for each context using a set of context-specific rules, we will be able to automate part of the process, and thus benefit organizations in cost reduction and in many other ways.

This paper is structured as follows. In Section 2, we review three common types of corporate householding problems, one of which we concentrate our discussion on in a later section. We then briefly go over some previous research findings in Section 3. After that, we present some of

the major application areas as well as examples from these fields in Section 4. In Section 5, we outline an engineering approach to model the corporate household knowledge domain, and to provide a technical solution to answer corporate householding questions based on an extension of the COntext mediation Interchange (COIN) technology. We conclude the paper in Section 6 and suggest future research directions.

2 Categories of Corporate Householding Problems

Corporate householding problems that businesses encounter come in various forms. Although ambiguities may exist in some cases, most of these problems can be categorized into the following three types.

- Entity identification. Part of the complexity that is involved in understanding corporate household data results from the multiple representations of the same entity. For instance, the corporate entity “IBM” can also be represented as “International Business Machines Corporation” or “I.B.M.”, though they all refer to exactly the same entity. One corporate entity can appear to be multiple entities in different data sources, and therefore can be difficult to identify correctly and efficiently.
- Entity aggregation. After we have identified that “IBM,” “International Business Machine Corporation,” and “I.B.M.” refer to the same entity, we need to determine what exactly that entity is. A large corporation’s corporate structure tree (or network) is usually very complex, including entities such as subsidiaries, branches, divisions, and joint ventures, normally with multiple layers. In one context, some parts of the corporate structure tree need to be taken into account in order to get a complete and correct view of the corporation; in another context, other parts of the tree may be considered. Therefore, when to aggregate which entities within a corporation to understand the precise meaning of the name depends on the task at hand.
- Transparency of inter-entity relationships. Relationships between corporate entities may involve multiple layers. For example, a seller can sell its products directly to its customers or through a broker. Knowing when the layers are important and when they are not poses another type of problem for corporate householding, which also has to be addressed depending on the context.

In most cases, when a corporate householding question is asked, the above three aspects all need to be considered in order to reach a correct answer. For example, if MIT wants to know how much it bought from IBM in the year 2002, it has to first identify all the instances of the same entity “IBM” in its records; then it needs to make sure to include purchases from entities that may not be directly related to IBM in their names but are in fact part of the corporation, such as its software subsidiaries Lotus and Rational; lastly, purchases of IBM products through brokers such as the local computer store CompuUSA may also be considered. In the later sections of this paper, we will primarily focus on the second type of corporate householding problems, that is, entity aggregation, and present a technical solution to aggregate corporate entities efficiently.

3 Previous Research

3.1 Research on Data Quality

Research efforts in data quality have been ongoing for many years. Organizations typically store a vast amount of data on distributed and heterogeneous systems for their internal and external activities. Therefore, well-managed and high-quality data is crucial to a company's success. Traditionally, "high quality" refers to the accuracy of data. In order to target the problems more precisely, research conducted at MIT Total Data Quality Management (TDQM) program [10, 11] has shown data quality as a multi-dimensional concept, which includes dimensions such as accessibility, timeliness, believability, relevance, and accuracy of data. Methods, models, tools, and techniques for managing data quality using the information product approach have also been proposed [11]. The approach includes a modeling technique to systematically represent the manufacture of an information product, methods to evaluate data quality, and capabilities to manage data quality. One research effort of the TDQM program is *corporate householding*, which aims at better understanding and utilizing corporate household data.

3.2 Research on Family Household

The conventional meaning of a household is "the people of a house collectively" [5]. The term "householding" has also been used increasingly in places such as an announcement sent out by the Security and Exchange Commission (SEC), which states: "*the Securities and Exchange Commission enacted a new rule that allows multiple shareowners residing at the same address the convenience of receiving a single copy of proxy and information statements, annual reports and prospectuses if they consent to do so. This is known as 'Householding'*" [5]. Traditional householding issues are similar to corporate householding problems. The structure of a family evolves over time, and may sometimes be very complex. Single mother or father families, families in which a husband and a wife have different last names and many other forms of families make it difficult to define and identify a family household. For instance, if a child goes to college in another city, will he/she be considered as part of the household? If two people live together but are not married, do they form a household? Similar to the corporate householding problems, these questions have no single "right" answers. We will need to consider the underlying purposes of the questions. In some areas such as customer data management, approaches for identifying family households have been implemented.

3.3 Commercial Approaches

In this section, we will explain some state-of-art practices used by some of the industry sponsors of the MIT TDQM Corporate Householding project, FirstLogic, Inc. and Dun & Bradstreet, to solve certain types of corporate householding problems.

FirstLogic uses the Subject Matter Experts (SME) approach to identify entities correctly and efficiently. This approach helps to identify and build hierarchical structures in order to represent relationships between two households (either family household or corporate household). Knowledgeable SMEs assist clients to establish the business rules that identify the entities in their own family structure (referred to as the "internal view"), as well as entities in the family structure of their business targets (referred to as the "external view"). The involvement of SMEs makes it possible to perform householding across task domains. The Firstlogic tools then allow

these rules to be applied across the company's database. The detailed steps follow: establish project goals, define applicable terms and gain cross-functional agreement, define the business rules that attain the goals, create and verify the rule matrix, create application parameters, check rules for cohesion, verify entire Match Sets and consolidation criteria, and finally run the householding process. Sometimes, rendering the householding process can be very time-consuming and costly [5].

Dun & Bradstreet (D&B) has developed a representation of corporate structure to improve the understanding of their relationships. The Data Universal Numbering System (D-U-N-S) number is a unique nine-digit non-indicative identification number assigned to every business entity in D&B's databases. It is widely used for keeping track of millions of corporate group structures and their relationships worldwide. The D&B Family Tree is comprised of linkages and business relationships. It captures eight types of entities (single location subsidiary, headquarters, branch, division, subsidiary, parent, domestic ultimate, and global ultimate) and two types of relationships/linkages (branch to headquarters and subsidiary to parent). Each family member carries up to four D-U-N-S numbers, including its own number, the number of its next highest member in the family, and its domestic ultimate's and global ultimate's numbers. D&B's approach captures a significant amount of useful information about a corporation, but there are some limitations to it. For example, the corporate householding applications of a company can be much broader than what the D&B family tree covers. For example, any subsidiaries that are less than 50% owned by parents are not listed in the parents' family trees. The D-U-N-S numbers and the D&B family tree represent the major part of corporate structure data, but do not embed the corporate householding knowledge. Also, this system for identifying corporations may not capture the way data is organized in the corporations' internal databases.

4 Corporate Householding Application Areas

Building on the discussion about previous research in Section 3, we will further our understanding about corporate householding in this section by exploring a few common application areas with examples drawn from a review of the literature and interviews with subject experts. Many examples are not industry specific – any corporations may encounter similar problems in their business functions that relate to these areas.

4.1 Account Consolidation

Corporate householding is needed in the consolidation of financial statements. For example, consider a large organization like IBM¹, which has over a hundred directly or indirectly-owned subsidiaries -- how should it prepare its financial statements? Should its financial statements be consolidated with those of Lotus, a company acquired by IBM? While ambiguities might arise about how one should calculate total sales and expenses for a company, the Securities and Exchange Commission (SEC) has laid out ground rules concerning consolidation in Regulation S-X, Article 3A (210.3A-02). The commission presents several criteria for establishing the most meaningful presentation of a company's financial position in its year-end statements. To answer the last question, one needs to

¹ For a complete list of IBM's subsidiaries, please refer to IBM's annual report on SEC filings: <http://www.sec.gov/Archives/edgar/data/51143/000104746903008194/a2102367zex-21.htm>

evaluate Lotus’s relationship with IBM. According to the criteria set by the SEC, IBM should consolidate its accounts with Lotus if it has majority ownership, in other words, if IBM owns more than half of Lotus. Since IBM owns Lotus entirely, it should indeed consolidate its financial statements.

But what would happen if IBM only directly owns 40% of Lotus but also owns 20% of Lotus indirectly (e.g. through another subsidiary)? Should consolidation of financial statements still occur? Under the SEC regulations, the existence of a parent-subsidary relationship regardless of whether or not there is a majority ownership of voting stock still requires consolidation of accounts, given that the consolidation is necessary in presenting a fair view of IBM’s financial position. This decision can become quite complex because there are multiple layers of subsidiaries involved in the case of IBM.

The general rule then is that companies should consolidate financial statements when there is majority ownership, either direct or indirect. However, there are some rare situations in which companies can forgo consolidation with majority-owned subsidiaries, if it does not have a controlling financial interest. Other criteria that may cause exceptions include if IBM and Lotus differed substantially in their financial periods, if IBM is a bank-holding company, and if Lotus is a foreign subsidiary of IBM. We have summarized the rules in Figure 1 below.

Not only does the parent company IBM need to have and understand knowledge about its relationships with its subsidiaries – other companies dealing with IBM also need this information. The challenge for these other companies is how to collect, organize, and retrieve the data on IBM in an intelligent way depending on context. In Section 5, we will use a simplified scenario of retrieving IBM’s revenue to demonstrate a technical approach to corporate householding.

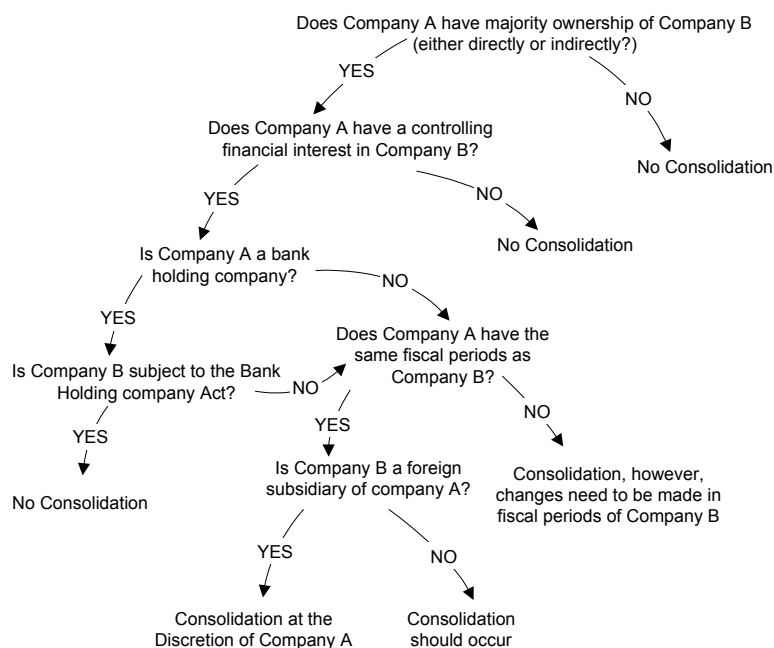


Figure 1: SEC Rules on Account Consolidation: Should Company A consolidate accounts with Company B?

4.2 Financial Risks

4.2.1 Credit Risk

One type of financial risk is credit risk, defined as “the risk associated with any kind of credit-linked events, such as: changes in credit quality (including downgrades or upgrades in credit ratings), variations of credit spreads, and the default event” [12]. Since credit is a crucial consideration in many financial transactions, corporate householding in the area of credit risk evaluation requires a significant amount of effort. The following are the types and examples of corporate householding activities that are related to credit risk management:

- Identifying multiple instances of the same entity. For example, a financial institution can extend credit to a number of different domestic as well as global organizations. Suppose CIBC is considering extending credit to IBM. To evaluate the overall risk involved, CIBC would need an aggregated report showing all branches of CIBC and their business relationships with all branches and subsidiaries of IBM. Given the complexities of both organizations, different CIBC branches may maintain information on IBM employing different ways of representation, such as using different names (IBM, I.B.M., or International Business Machines). Even if the names that the CIBC branches use are the same, the contact information of IBM that CIBC branches maintain is likely to differ by region.
- Understanding relationships between entities. For example, a firm planning to extend a large credit line to Hewlett Packard Puerto Rico may find it useful to know that Hewlett Packard only has a rating of AA, though Hewlett Packard Puerto Rico has a credit rating of AAA. In other words, when evaluating the credit risk of a subsidiary, its parent and other related entities (if any) should be considered as well.
- Considering the dynamics of corporations over time. For example, suppose a bond held by a bank a year ago had an investment grade rating, but since then the company's bonds have been downgraded to junk bond status. The bank should know about the change in status and re-evaluate the bond even though it is still the same bond with the same company.

4.2.2 Bankruptcy Risk

Bankruptcy risk is closely related to credit risk. Bankruptcy normally results either in “liquidation of debtor’s nonexempt property” or “debtor rehabilitation” or “reorganization of the debtor’s assets” [13]. When deciding whether or not to issue loans to a particular company, banks need to know who is responsible if the company bankrupts. For example, if a subsidiary goes bankrupt, how much liability (if any) does the parent company have? One concept that plays a significant role in the bankruptcy rules is affiliate. The definition of an “affiliate” covers parent corporations, subsidiaries of the debtor, and sister affiliates of the debtor, using a 20% stock ownership trigger. In particular, an affiliate is defined as: a) any entity that owns or controls 20% or more of the outstanding voting securities of the debtor; or b) any subsidiary for which the debtor parent corporation owns or controls more than 20% of the outstanding voting shares; or c) any sister subsidiary of the debtor for which a common parent corporation owns or controls 20% or more of the outstanding voting shares of both (exceptions for fiduciaries and

minority shareholders.) Depending on whether or not a corporation is an affiliate, different Bankruptcy Codes may apply. Affiliate status also determines if the corporation is an “insider” under the Bankruptcy Code. In addition, bankruptcy laws and regulations vary from country to country, increasing the need for Corporate Householding.

4.2.3 International Risk

As companies develop increasing global reach, risks caused by the differences in business protocols used in different parts of the world need to be considered. We name this risk “international risk.” When banks or credit rating agencies evaluate the level of risk associated with global companies, locations may cause ambiguity. For example, does the risk involved in a loan to a company depend on the location of that particular borrower or the location of the borrower’s owner? Consider a company that is located in Brazil but is also a division of a larger-sized American company. Or consider a company located in the US, whose parent company is in Japan, such as a Toyota manufacturing plant in the US. When should this plant be considered a “US company” and when should it be considered simply a subsidiary of a Japanese firm?

4.3 Legal Sector

There are many types of corporate householding activities in the legal domain as well. In this paper, we will focus our discussion on the topic of software licensing. A problem that many software vendors face is how to keep their information current regarding the consolidation of customers through mergers and acquisitions. It is a particular challenge for vendors of enterprise-wide solutions and those who sell enterprise-wide licenses. For example, a software vendor experienced this problem when two of its printer-publisher customers merged to form one of the world’s largest printer-publishers. After the merger the original two companies had to compare databases of software licenses to determine licensing overlap. A manual householding process was used to compare the two hierarchies of vendor -> product -> license -> restrictions. They could consolidate licenses where products overlapped, but licensing restrictions did not prevent usage at new facilities or “sites.”

Understanding corporate structures and relationships is valuable not only to the licensees, but also to the vendors. For example, a vendor that sells to a multi-national corporation that has a very extensive corporate family tree may find it beneficial to identify which entities in the family tree they have sold licenses to, and under what restrictions. Not only does this allow them to roll-up all revenues for that one global customer and produce a single customer view, but it also allows the vendor to propose the optimal licensing configuration. Obviously the vendor does not want to quote an enterprise-wide license that would essentially distribute its software for free to the many divisions of the global parent. However, the vendor also does not want to impose such restrictive terms as to place it at a disadvantage to its competition.

4.4 Business Management and Operations

Corporate householding issues also exist in areas such as customer relationship management, supply chain management, sales and marketing, and business intelligence. A few examples follow:

- Customer relationship management. Product vendors need to perform corporate householding in order to efficiently structure their customer records and to improve the

effectiveness of customer communications. The customer could be a retail customer or a corporate account. When the customer is a multi-national corporation, the vendor usually has hundreds of unique contact records (individuals) in its information systems. In many cases, vendors develop their own ways of keeping track of the information, such as assigning unique identifiers to business accounts.

- Sales and marketing. The sales and marketing functions of the corporation also need to manage customer information. There is a growing need for customer-identification systems that can provide integrated views of business-to-business customers to identify existing or high-potential customers, to assign resources to penetrate them, and to report on the performance of these efforts.
- Supply chain management. One of the examples in this area is from an information executive of a global manufacturing company. Her company is very interested in global sourcing to identify a manufacture site that could produce a particular product with the lowest costs (including the cost of manufacturing as well as transportation.) A large part of the manufacturing cost comes from raw material cost. Therefore, identifying and maintaining relationship with material vendors is critical in order to achieve cost reduction. However, due to localized information systems, two manufacturing sites of her company are highly likely to have two different, independent relationships/contracts with the same vendor for the same material. The situation becomes even more complicated when a vendor has different relationships with different corporate function areas, such as manufacturing, financial and accounting systems. Therefore, it becomes very hard to have a single and consistent view of a global vendor.

The inconsistencies among information systems that are maintained locally make it difficult for a company to understand its relationship with its business partners. In the case of this global manufacturing company, it is impossible for it to know how much raw materials are used globally. Additionally, the company cannot take advantage of the lowest price across all of its manufacturing sites from a particular vendor.

5 Corporate Householding Query Processor

Although corporate householding applies across all major business domains, most corporate householding problems can be classified into the three categories mentioned in Section 2: (1) entity identification, (2) entity aggregation, and (3) transparency of inter-entity relationships. In this section, we propose a new approach for solving the second type of corporate householding problems – entity aggregation. This approach is based on an extension of the COntext INterchange (COIN) technology developed at MIT.

5.1 Motivational Example

Let us consider the following example. Suppose Sally is a financial analyst and she would like to find out what IBM's total revenue is in fiscal year 2002 (IBM's fiscal period ends on December 31st). International Business Machines Corporation is a giant organization with about 100 years of history and numerous branches and offices around the globe with more than 100 subsidiaries²

² For a complete list of IBM's subsidiaries, see footnote 1.

directly or indirectly owned by this company. Although these subsidiaries are legally independent organizations, according to SEC’s accounting rules (summarized in Section 4.1), IBM should consolidate the revenues from all the majority-owned subsidiaries in its annual reports. Sally follows SEC’s rules, but the database that she gets her data from represents IBM’s revenue differently. The revenues of IBM and its subsidiaries are not consolidated. For illustration purposes, let’s assume the data source only considers a couple of subsidiaries of IBM – Lotus Development, IBM Far East Holdings B.V., International Information Products (Shenzhen) Co., Ltd and IBM International Treasury Services. Lotus is directly and wholly owned by IBM; International Information Products is owned 80% by IBM Far East Holdings B.V., which is a wholly-owned subsidiary of IBM; IBM International Treasury Services is owned by five different local branches of IBM in Europe, including IBM Germany and IBM France. The revenue number corresponding to “CorporateEntity = IBM” in the table “revenue1” in Figure 2 does not include the revenues of Lotus, IBM Far East Holdings, International Information Products and IBM International Treasury Services.

Receiver: Sally

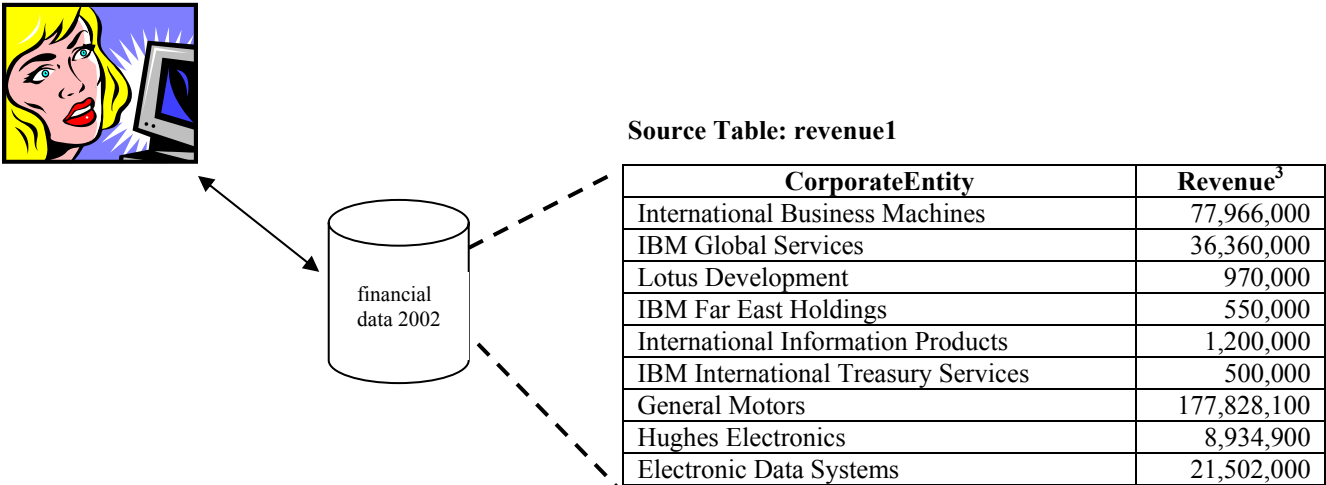


Figure 2. Motivational Example: Performing a Query for Total Revenue of IBM in Year 2002

The source database also includes revenue data on other corporate entities related to IBM, such as one of its divisions – the company’s consulting arm, IBM Global Services. However, because it is a division only (not an entity legally separated from IBM), its revenue is already consolidated in the revenue1 table and should not be double-counted. To illustrate Sally’s accounting rules (SEC’s rules) better, we summarize them in the decision tree in Figure 3. It is based on the diagram on account consolidation rules from Section 4.1. The tree can grow further beyond the current nodes, but in this example, we keep the simple version of it.

³ The revenue data is directly extracted or estimated from the revenue/sales data of IBM, Lotus, GM, Hughes, etc. See Appendix A, B and C.

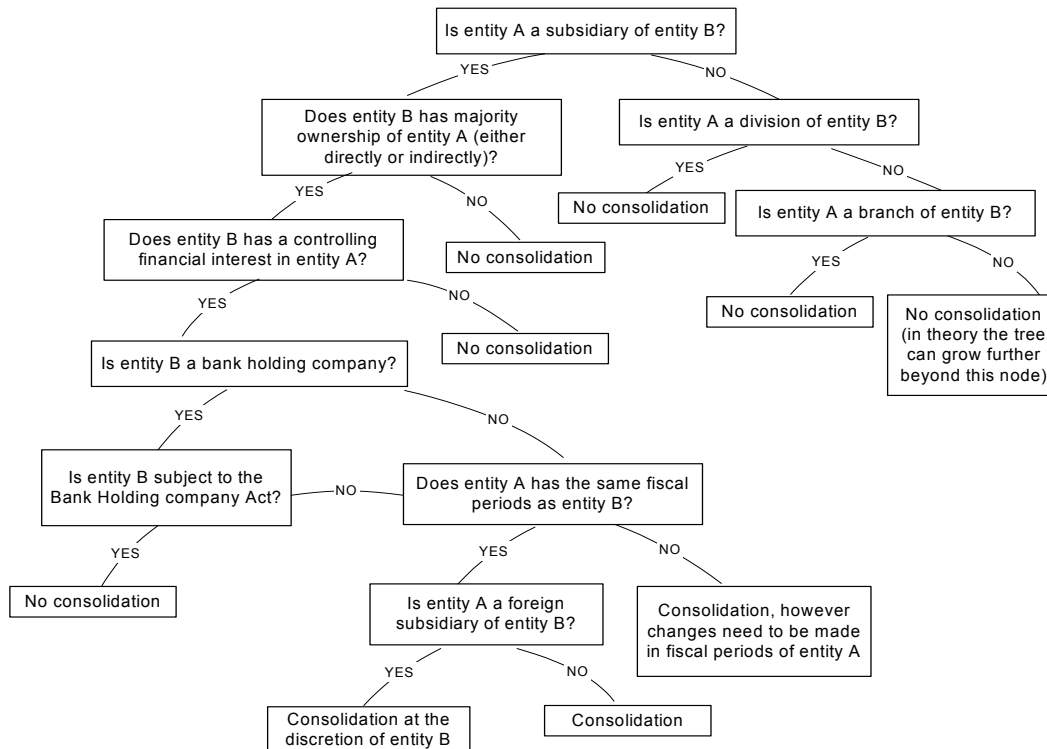


Figure 3: Decision Tree (simplified) that Represents the SEC’s Rules for Calculating Total Revenue

Given the disparities between Sally (the “receiver”) and the data source’s accounting principles, if Sally issues a simple query (using SQL query notation) –

*Select CorporateEntity, Revenue from revenue1
where CorporateEntity = “International Business Machines”*

on the source database directly, she will get back –

<i>CorporateEntity</i>	<i>Revenue</i>
<i>International Business Machines</i>	<i>77,966,000</i>

But this result is not the one that she is looking for. The total revenue of IBM, from Sally’s point of view, should include all the revenues of IBM’s subsidiaries with majority ownership. For this particular data source, the query should return the SUM of the revenues from International Business Machines, Lotus Development, IBM Far East Holdings, International Information Products, and IBM International Treasury Services. In other words, we should “aggregate” all these entities with their parent entity IBM, and hence, Sally’s problem is an entity aggregation problem in corporate householding.

Realizing that the simple query is not sufficient, what should Sally do to modify the query so that the desired result will be returned? In theory, she will need to consider each one of a few hundred entities in IBM’s corporate family using the decision tree in Figure 3, and to find out whether any particular entity should or should not be aggregated. Because all the leaves of the decision tree are mutually exclusive, any pair of entity A and entity B corresponds to only one path from the root node to a leaf, which indicates “consolidation” or “no consolidation”. In order to traverse the tree successfully, Sally has to rely on some auxiliary data sources that provide information on the entities within IBM’s corporate group, including ownership percentages, controlling financial interest, fiscal periods and other related information.

For the simplified “revenue1” table in our example, entity A can take values such as Lotus Development, International Information Products and IBM Global Services; but entity B is reserved for International Business Machines. So Sally only need to consider five pairs of entities. The steps of reasoning for {entity A = Lotus Development, entity B = International Business Machines} are illustrated in part (A) of Figure 4 below. Because Lotus Development Corporation is wholly owned by IBM, IBM has “majority ownership” of Lotus, and has “a controlling financial interest” in Lotus. IBM is not a bank holding company⁴, and it has the same fiscal periods as Lotus. Both of them are incorporated in the US. Based on the above information, Sally can conclude that Lotus’s revenue should be consolidated with IBM’s total revenue. Similarly, the revenues of IBM Far East Holdings, International Information Products and IBM International Treasury Services should also be consolidated (assume IBM consolidates revenues from its foreign subsidiaries.) On the other hand, when entity A = IBM Global Services and entity B = International Business Machines (steps of reasoning shown in part (B) of Figure 4), because IBM Global Services is a division of IBM, no consolidation should occur to avoid double counting.

After reasoning through the source data using her accounting rules, Sally knows that she should actually issue the following query on the database:

*Select “IBM” as CorporateEntity, SUM(Revenue) as Revenue from revenue1
where CorporateEntity in (“International Business Machines”, “Lotus Development”, “IBM Far East Holdings”, “International Information Products”, “IBM International Treasury Services”)*

And she gets back

<u>CorporateEntity</u>	<u>Revenue</u>
IBM	81,186,000

This is the correct result, because “77,966,000 + 970,000 + 550,000 + 1,200,000 + 500,000 = 81,186,000.”

Now let us suppose Sally’s accounting rule has changed. According to the new rules, she consolidates only the revenues from the wholly-owned subsidiaries, but not those that are only partially owned by IBM. Since International Information Products is only 80% (indirectly) owned by IBM, its revenue should not be consolidated with IBM’s total revenue, whereas all the other subsidiaries in our example should be, because they are 100% owned by IBM. Therefore, Sally’s query on the table revenue1 should look as follows.

*Select “IBM” as CorporateEntity, SUM(Revenue) as Revenue from revenue1
where CorporateEntity in (“International Business Machines”, “Lotus Development Corp”, “IBM Far East Holdings”, “IBM International Treasury Services”)*

The result she gets back is

<u>CorporateEntity</u>	<u>Revenue</u>
IBM	79,986,000

⁴ According to the Bank Holding Company Act, a Bank Holding Company is “any company [that] has control over any bank or over any company that is or becomes a bank holding company by virtue of this Act.” “Any company has control over a bank or over any company if: (A) the company directly or indirectly or acting through one or more other persons owns, controls, or has power to vote 25 per centum or more of any class of voting securities of the bank or company; (B) the company controls in any manner the election of a majority of the directors or trustees of the bank or company.

The above example captures the essences of entity aggregation problems. For any entity aggregation problem, whether its purpose is account consolidation or credit risk evaluation or sales and marketing, we want to find out what entities in the corporate family should be considered and should contribute to the final result, given the purpose. We perform corporate householding using the corporate group structure data, rules and regulations specific to the purpose, and other related information. The reasoning process described above can get very tedious and costly if the ‘revenue1’ table Sally is querying on has all of the subsidiaries’ data separated out from the parent, for she will have to consider the subsidiaries, divisions and branches one by one. However, it is probably closer to reality than the sample table in Figure 2. It would be valuable to have a system that will capture the differences in aggregation rules between the source and the receiver of a query, test the entities recursively using these rules, and mediate the query according to the receiver’s expectation to achieve the desired entity aggregation. That way Sally would not have to perform the corporate householding task manually when she searches for the total revenue of IBM. The COntext INterchange technology presents a unique approach to capture the differences between contexts (semantics of data sources and the receiver), to resolve those conflicts, and to output correct data in a fast and efficient way. The COntext INterchange (COIN) system, with some extensions, can be used to store and process corporate householding knowledge. In particular, it can help to solve entity aggregation problems such as the challenges that the motivational example poses. In the following section, we will describe the COIN system in detail.

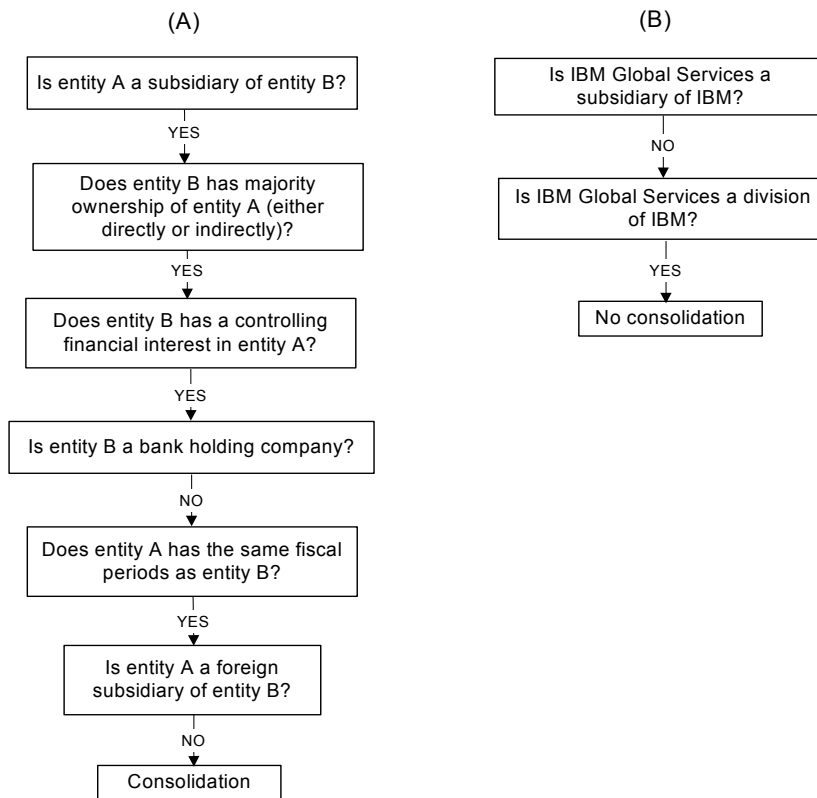


Figure 4: Steps of Reasoning Using the Decision Tree in Figure 3

(Entity A = Lotus Development in (A) and IBM Global Services in (B); Entity B = International Business Machines)

5.2 Overview of the Context Interchange Technology

Context Interchange (COIN) technology is a mediation approach for semantic integration of disparate (heterogeneous and distributed) information sources in order to achieve semantic interoperability and logical connectivity. The COIN architecture consists of three major components: (1) client processes such as applications that perform queries on multiple databases; (2) server processes including database gateways and wrappers; and (3) the mediator process, which is the core of the entire system. The Context Mediator rewrites queries in the receiver's "context" into a set of mediated queries where all conflicts are automatically detected and explicitly resolved. The process is based on an abduction procedure that determines what data are needed to answer the query and how conflicts should be resolved by using the axioms associated with the contexts. Automatic identification and reconciliation of conflicts are made possible by general knowledge of the application domain and the implicit assumptions associated to the sources and receivers. The three components of the COIN architecture work together to enable efficient and meaningful use of heterogeneous data, when the data sources and potential receivers have semantic differences. The modularity of the design keeps the system scalable with increasing numbers of sources and receivers, extensible to local changes in the system, and accessible to end-users.

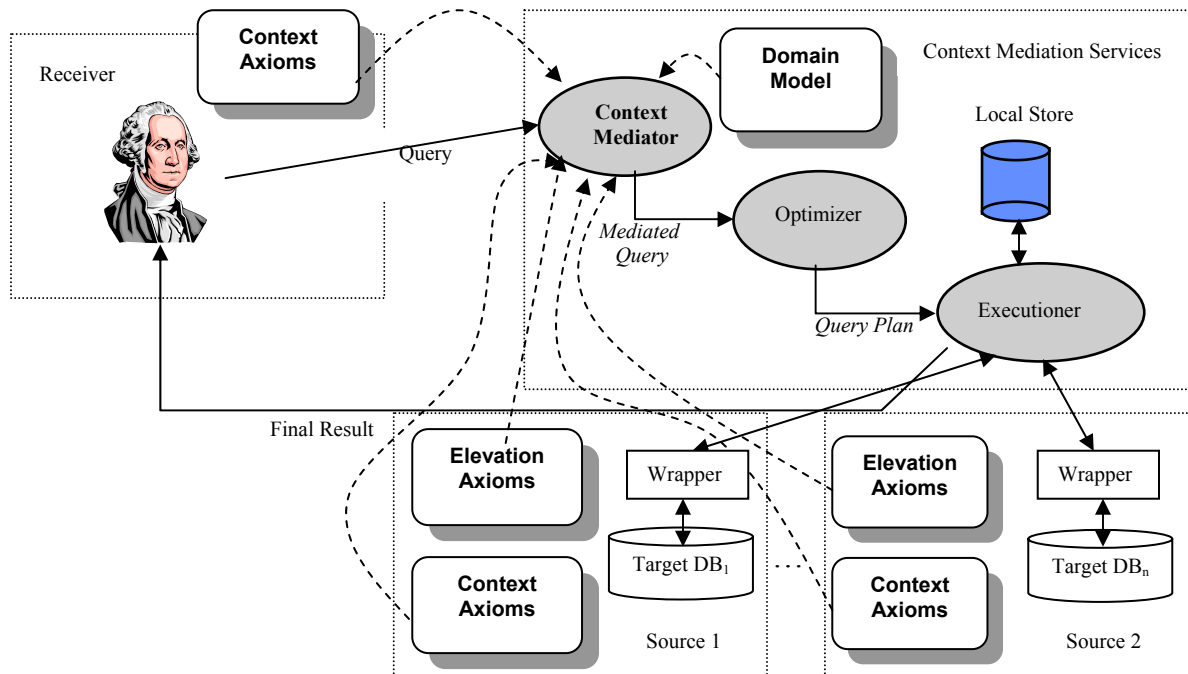


Figure 5. Architectural Overview of the Context Interchange (COIN) System

Figure 5 illustrates the components of COIN. The COIN framework consists of a data model and a logical language, COINL, which are used to describe a Domain Model that can represent the source, the receiver and the contexts associated with them. A Domain Model specifies the semantics of the "types" of information units, which constitute a vocabulary used in capturing the semantics of data in disparate sources. Three kinds of relationships are expressed: inheritance, attributes and modifiers; the values of modifiers vary depending on the context. Together they define the ontology that will be used. The Elevation Axioms identify the correspondences between attributes in the

source and semantic types in the Domain Model. The Context Axioms define alternative interpretations of the semantic objects in different contexts. These three components enable the Context Mediator to generate the correct mediated query from the original user query. Besides the context mediator, the mediation services also include a query optimizer and a query executioner to enhance performance. The result of the query execution is reformulated into the receiver's context.

The research in Context Interchange and the COIN framework has been ongoing for some time, and a prototype system has been developed to validate the method. Applications that perform queries on disparate data sources (such as financial databases and on-line shopping sites) have been built and thoroughly tested. Also, a user interface aimed to ease the development of COIN applications has been developed, such that application developers need to specify only the ontology model, contexts (with elevations, conversion functions between contexts, and modifier values for every defined context), and data sources (relations) to build an application based on the COIN system.

Corporate householding problems, especially entity aggregation problems, are very similar to traditional COIN applications in the sense that entity aggregation also involves different source and receiver contexts. Under different contexts, an entity may or may not need to be aggregated. For instance, as discussed in the motivational example, for the purpose of account consolidation using SEC's regulations, Lotus Development should be considered as part of IBM. However, in the source context, Lotus's revenue is not consolidated with IBM's. These differences in contexts can be captured in the COIN system. In Section 5.3, we will define the components of the corporate householding query mediator in detail and illustrate how context mediation can be used to solve the entity aggregation problems in the motivational example.

5.3 Corporate Householding Query Mediation Example

5.3.1 Ontology Framework

The first thing we need to do is to specify the domain model (ontology framework) for the sample corporate householding problem, as shown in Figure 6. The semantic types are divided into two categories – corporate structure related and task related. Corporate structure related semantic types represent common concepts in corporate group structure and entity aggregation, and thus are useful in any entity aggregation problems; the task related semantic types shown here are specific to the account consolidation example we are considering. This ontology can be extended easily to accommodate entity aggregation problems in other application areas by substituting the current task related semantic types with a set of new task related types and setting appropriate relationships across the two categories of semantic types.

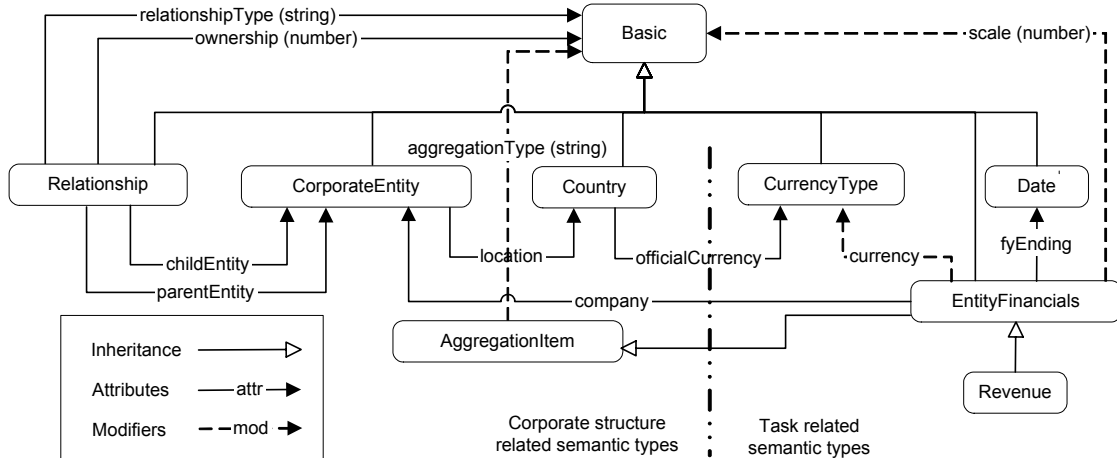


Figure 6: Ontology for the Motivational Example on Account Consolidation

Three kinds of arrows in Figure 6 represent the “inheritance”, “attributes” and “modifiers” relationships respectively.

- **Inheritance:** the classic type of ‘is-a’ relationship. All semantic types root from one semantic type – “Basic”, which includes system native types such as integers, strings, and real numbers. If type B inherits from type A, B is a sub-type of A and inherits all A’s properties and attributes. For example, in Figure 6, “Revenue” inherits from “EntityFinancials”; thus it automatically has modifiers, such as “currency”(which represents which currency the financial data is in) and attributes, such as “fyEnding”(which represents the ending date of the fiscal period associated with this piece of financial data).
- **Attributes:** used to represent the structural properties of semantic types. In other words, they define relationships between objects of corresponding semantic types. For example, the semantic type “CorporateEntity” has an attribute called “location” of type Country. This attribute represents the country of incorporation of a corporate entity (if it is a legally independent entity) or the location of an entity (if it is not legally independent). The semantic type “Relationship” has attributes “parentEntity”, “childEntity”, “relationshipType”, and “ownership”. The “parentEntity” (of type CorporateEntity) owns the “childEntity” (also of type CorporateEntity) with ownership percentage equal to the value of “ownership”. The types of relationship between child and parent entities include subsidiary, branch and division, and they are captured by the attribute “relationshipType”.
- **Modifiers:** special attributes whose values vary depending on the context and whose values determine the interpretations of data. Modifiers are used in conflict detection during query mediation. For instance, the modifier “currency” has value “USD” in a US based context, and value “GBP” in a UK based context. The modifier “aggregationType” has value “division+branch” in an unconsolidated revenue context, and value “subsidiary+division+branch” in a consolidated revenue context based on the SEC rules.

The following sections will explain in detail the semantic types, attributes and modifiers in the ontology model.

5.3.2 Corporate Structure Related Semantic Types and Data

5.3.2.1 Semantic Types

The semantic types in this category are closely associated with representations of entity aggregation, corporate group structure and relationships between corporate entities:

- **CorporateEntity**: inherits from Basic. This semantic type has the attribute *location* of type Country, which specifies the corporate entity’s country of incorporation or its location. Some sample values that CorporateEntity may take are “Johnson & Johnson” and “Citibank Canada”; some sample values for “Country” are “USA” and “Canada”.
- **AggregationItem**: inherits from Basic. It is a super-type of any specific item that is being aggregated. These specific items are semantic types in the task-related domain, such as EntityFinancials in the current ontology. Other subtypes of AggregationItem may include Employee, Customer, or CreditRisk, depending on the task at hand. The modifier *aggregationType* specifies how the items should be aggregated. Suppose that the aggregation rule in the context concerned is to aggregate all divisions, branches and wholly owned subsidiaries with their parents. The value of aggregationType here is therefore “whollyownedsubsidiary+division+branch.”
- **Country**: inherits from Basic and has an attribute *officialCurrency* of type CurrencyType, which captures the official currency type of the country concerned.
- **Relationship**: inherits from Basic and has attributes *relationshipType* of type Basic, *ownership* of type Basic, and *parentEntity* and *childEntity* of type CorporateEntity. For instance, we know that Lotus is a subsidiary 100% owned by IBM. This relationship can be represented as “childEntity = Lotus, parentEntity = IBM, relationshipType = Subsidiary, ownership = 100.”

5.3.2.2 Data

In order to perform corporate householding we need information about corporate structures. Part of the desired relation (“structure”⁵) is shown in Table 1:

ChildEntity	ParentEntity	RelationshipType	Ownership
IBM Credit Corp.	International Business Machines	Subsidiary	100
Lotus Development	International Business Machines	Subsidiary	100
IBM Far East Holdings B. V.	International Business Machines	Subsidiary	100
International Information Products	IBM Far East Holdings B. V.	Subsidiary	80
IBM Global Services	International Business Machines	Division	100
IBM Germany	International Business Machines	Branch	100
IBM France	International Business Machines	Branch	100
IBM Finland	International Business Machines	Branch	100
IBM Denmark	International Business Machines	Branch	100
IBM Switzerland	International Business Machines	Branch	100
IBM International Treasury Services	IBM Germany	Subsidiary	33
IBM International Treasury Services	IBM France	Subsidiary	14
IBM International Treasury Services	IBM Finland	Subsidiary	10
IBM International Treasury Services	IBM Denmark	Subsidiary	18

⁵ The data in this table is extracted from Exhibit 20.01 in IBM’s annual report for the year ending Dec.31st, 2002. See footnote 1.

IBM International Treasury Services	IBM Switzerland	Subsidiary	25
Hughes Electronics	General Motors	Subsidiary	100
...

Table 1: Relation “structure”: Pairs of Related Corporate Entities and Details about Their Relationship

The columns in the above table are self-explanatory. For example, IBM Far East Holdings B. V. is a wholly owned subsidiary of IBM, and International Information Products is 80% owned by IBM Far East Holdings. The “ownership” column describes the percentage ownership of the “ParentEntity” on the “ChildEntity”, and it can take values up to 100 (wholly-owned subsidiaries or divisions or branches). For demonstration purposes, we have implemented the structure table as an inline database table, that is, a set of rule statements in the abduction code. Only a minor change in the code is needed if database calls are implemented and added in the future. A sample statement that defines one row of the above table follows:

rule(structure("Lotus Development", "International Business Machines", "Subsidiary", 100), (true)).

Another piece of information we need is the country of incorporation or location table. We name this relation “country,” and part of this relation looks as follows:

CorporateEntity	Country
International Business Machines	USA
Lotus Development	USA
IBM Far East Holdings B. V.	Netherlands
International Information Products	China
IBM Germany	Germany
IBM France	France
IBM Finland	Finland
IBM Denmark	Denmark
IBM Switzerland	Switzerland
IBM International Treasury Services	Ireland
General Motors	USA
Hughes Electronics	USA
...	...

Table 2: Relation “country”: Country of Incorporated or Location of Corporate Entities

The above two relations – “structure” and “country” – are generic across all contexts; in other words, no matter what the purpose of the query is, the data from these two tables will be used and they will not change. We may also need other data sources to derive answers to the questions asked in the decision rules that are context-specific. For example, in the context of “c_majorityowned_revenue”, according to the decision tree in Section 5.1, we will need to know information on (1) a company’s controlling financial interest on the other, (2) if a company is a bank holding company and if it is subject to the Bank Holding company Act, and (3) if two entities have the same fiscal period. Here, we have simply assumed that IBM has controlling financial interest on its subsidiaries, it is not a bank holding company, and they share the same fiscal period end date. Therefore, the types of relationships and ownership percentages are what determine the aggregation between entities.

5.3.3 Task Related Semantic Types and Data

5.3.3.1 Semantic Types

The task related part of the ontology includes the semantic types related to a specific task, i.e., one of the corporate householding application areas. Different kinds of task related components could be added onto the current ontology model when different problem domains are considered. Here, we include only some of the semantic types in the Company Financials domain, those that are closely related to our “total revenue” example.

- **EntityFinancials**: inherits from **AggregationItem**, and encapsulates the representations of a corporate entity’s financial information. It has attribute *fyEnding*, as well as modifiers *company*, *currency*, and *scale*. For example, the fact that Entity A’s fiscal year ends on Dec. 31 and its financial data is in thousands USD is represented by “entity = Entity A, fyEnding = 12/31, currency = USD, scale = 1000.” Because EntityFinancial is a subtype of **AggregationItem**, it inherits the modifier *aggregationType*.
- **Revenue**: inherits from **EntityFinancials**, and thus inherits all its attributes and modifiers by default. It captures a corporate entity’s revenue data.
- **CurrencyType** and **Date**: inherit from **Basic**. They help to define **EntityFinancials**, and could be shared by other problem domains.

In the following sections, we will explain how the **Context Interchange** technology and the ontology framework for the corporate householding problem described in previous sections can help mediate (i.e., rewrite) the query to better meet users’ needs. The purpose of this example is to demonstrate the working of the system, which we believe is scalable and flexible enough to be extended to more realistic cases with a reasonable amount of add-ons and more specification.

5.3.3.2 Data

We have already presented the key needed task relation in Figure 2, that is, the “revenue1” (in the *c_unconsolidated_revenue* context) table. In reality, this table could be the result of a join on multiple data sources. We assume: (1) the “Revenue” corresponding to “CorporateEntity = IBM” does not include the revenue from four of IBM’s subsidiaries - Lotus Development, International Information Products, IBM Far East Holdings B. V., and IBM International Treasury Services, but includes revenues from all divisions, branches and other subsidiaries; (2) all the entities have the same fiscal periods ending on December 31 and the data is for year 2002; (3) IBM consolidates the revenues from its foreign subsidiaries.

5.3.4 The Contexts and Rules

Recall the example described in Section 5.1. Sally wants to find out IBM’s total revenue in fiscal year 2002. Here, we don’t care which organization Sally belongs to. She could be from the sales department of a large organization that does business with IBM and its branches, subsidiaries, etc. Because of the size of her company, its sales data may be scattered across the company’s databases around the country (or even the world) that have differences in semantics and data representations. Or, she could be from an organization that needs IBM’s financial data, in which case she is querying on outside data sources and trying to figure out how much total revenue IBM has accomplished in fiscal year 2002.

We name the context of the data source Sally uses “c_unconsolidated_revenue”, and Sally’s own context “c_majorityowned_revenue” (because Sally uses accounting rules from SEC, which require consolidation of majority owned subsidiaries). Another possible context, that we will consider, is “c_whollyowned_revenue”, which requires account consolidation of only wholly owned subsidiaries. The following table compares the differences and similarities among these contexts:

Context Name	Revenue (of any corporate entity)	scale	currency	aggregationType
c_unconsolidated_revenue	Includes revenues of its divisions and branches only	1000	USD	division+branch
c_majorityowned_revenue	Includes revenues of its divisions, branches, and majority owned subsidiaries	1000	USD	subsidiary+division+branch
c_whollyowned_revenue	Includes revenues of its divisions, branches, and wholly owned subsidiaries	1,000,000	USD	whollyownedsubsidiary+division+branch

Table 3: Context Comparison in the Motivational Example

Similar to the modifier values as shown above, rules (decision trees) are defined per context as well. For instance, if Sally is in the c_majorityowned_revenue context, the purpose of Sally’s query is to find out the “total revenue” of a company using SEC’s consolidation rules, and the tree that represents the rules in this context is shown in Figure 3 of Section 5.1. Here, the example is simplified such that the value of the modifier “aggregationType” captures the rules in different contexts.

5.3.5 Modifiers

Table 1 summarizes the values that the modifiers take in the three contexts. For example, the modifier *scale* is 1000 in the first two contexts, but is 1,000,000 in the c_whollyowned_revenue context. This means that the actual revenue figures in different contexts may differ by a factor of 1000. The definitions of these modifiers in COINL⁶ look like follows (using the c_unconsolidated_revenue context as an example):

```
modifier(
  'EntityFinancials', Object, aggregationType, c_unconsolidated_revenue, Modifier),
(cste(basic, Modifier, c_unconsolidated_revenue, "division+branch"));
```

```
modifier(
  'EntityFinancials', Object, currency, c_unconsolidated_revenue, Modifier),
(cste(CurrencyType, Modifier, c_unconsolidated_revenue, "USD"));
```

```
modifier(
  'EntityFinancials', Object, scale, c_unconsolidated_revenue, Modifier),
(cste(basic, Modifier, c_unconsolidated_revenue, 1000)).
```

Every modifier corresponds to a potential conflict that may occur between the context “c_unconsolidated_revenue” and some other context. For example, the above clause states that the modifier “scale” for the object *Object* of type EntityFinancials in the “c_unconsolidated_revenue” context is the object *Modifier* where *Modifier* is a constant (cste) of type Basic and value 1000 in this context.

⁶ COINL is a logical programming language based on F-logic, which is based on Prolog.

5.3.6 Conversion Functions

Conversion functions define how modifier values change between different contexts. In most cases, they are defined independent of any specific source or receiver context. During query mediation, the Context Mediator decides whether or not a conversion should be used. For example, the following is a conversion function between scales in different contexts:

```
cvt(EntityFinancials, _O, scale, Ctxt, Mvs, Vs, Mvt, Vt):-  
    Ratio is Mvs / Mvt,  
    Vt is Vs * Ratio,
```

where *scale* is a modifier of semantic type *EntityFinancials*, and has value *Mvs* in the source context and value *Mvt* in the target receiver context. The value of *scale* for an object *_O* of type *EntityFinancials* in the receiver context (*Vt*) is equal to the value of *scale* for *_O* in the source context (*Vs*) multiplied by the *Ratio* of the modifier value in the source context to the modifier value in the receiver context. In our example, the ratio might be 1000/1000, 1,000,000/1000, or 1000/1,000,000 – depending upon the contexts being considered.

The conversion functions that take care of the modifier *aggregationType* encapsulate the reasoning part of the aggregation process according to relationship types and ownership percentages. The conversion function between the *c_unconsolidated_revenue* context and the *c_majorityowned_revenue* context (the source and receiver contexts in our example) look as follows:

```
cvt(commutative,'EntityFinancials', O, aggregationType, Ctxt, "division+branch", Vs, "subsidiary+division+branch",  
Vt):-
```

```
(attr(O, company, C),  
attr(C,subsidiary,SL),  
flatten(SL,TFSL),  
length(TFSL,Length),  
if(Length < 1, VLSA is 0,  
    (attr(C,percentsubsidiary,PL),  
    maplist(attr2(company),SL,EOL),  
    maplist(value_reverse2(Ctxt),EOL,VL),  
    maplist(value_reverse2(Ctxt),PL,PLV),  
    listproduct(VL,PLV,LPV),  
    flatten(LPV,FLPV),  
    sumall(FLPV,VLS),  
    divide(VLS,100,VLSA))),  
  
attr(C, division, DL),  
flatten(DL,TFDL),  
length(TFDL,DLength),  
if(DLength < 1, DivSubVal is 0,  
    (attr(C,percentdivision,PDL),  
    maplist(value_reverse2(Ctxt),PDL,PDLV),  
    maplist(attr_reverse2(subsidiary),DL,DLS),  
    flatten(DLS,TFDLS),  
    length(TFDLS,DLSLength),  
    if(DLSLength < 1, DivSubVal is 0,  
        (maplist(attr_reverse2(percentsubsidiary),DL,PSDL),  
        maplist(maplist(value_reverse2(Ctxt)),PSDL,PSDLV),
```

```

        maplist(maplist(attr2(company)),DLS,DLSR),
        maplist(maplist(value_reverse2(Ctxt)),DLSR,DLSRV),
        listproduct(DLSRV,PDLV,ADLSRV),
        flatten(ADLSRV,FDLSRV),
        flatten(PSDLV,FPSDLV),
        listproduct(FDLSRV,FPSDLV,DivSubValL),
        sumall(DivSubValL,DivSubValI),
        divide(DivSubValI,10000,DivSubValI))),
    attr(C,branch,BL),
    flatten(BL,TFBL),
    length(TFBL,BLength),
    if(BLength < 1, BrSubVal is 0,
      (attr(C,percentbranch,PBL),
       maplist(value_reverse2(Ctxt),PBL,PBLV),
       maplist(attr_reverse2(subsidiary),BL,BLS),
       flatten(BLS,TFBLS),
       length(TFBLS,BLSLength),
       if(BLSLength < 1, BrSubVal is 0, (maplist(attr_reverse2(percentsubsidiary),BL,PSBL),
         maplist(maplist(value_reverse2(Ctxt)),PSBL,PSBLV),
         maplist(maplist(attr2(company)),BLS,BLSR),
         maplist(maplist(value_reverse2(Ctxt)),BLSR,BLSRV),
         listproduct(BLSRV,PBLV,ABLRSRV),
         flatten(ABLRSRV,FBLRSRV),
         flatten(PSBLV,FPSBLV),
         listproduct(FBLRSRV,FPSBLV,BrSubValL),
         sumall(BrSubValL,BrSubValI),
         divide(BrSubValI,10000,BrSubValI))))),
      plus(VLSA,DivSubVal,I),
      plus(I,BrSubVal,II),
      plus(II,Vs,Vt)).

```

Although the conversion function looks quite complex, a user or corporate householding application developer need not be aware of it nor need to change it – that is the power of this approach.

The above conversion function makes calls to many helper functions in the abduction engine. Nevertheless, the reasoning steps can be described in words: first, the ownership percentages (directly or indirectly regardless) of all the subsidiaries of the corporate entity concerned are calculated, through some recursive helper functions that are defined in the abduction engine; then, the function filters out those subsidiaries that are not majority-owned; lastly, it specifies that the revenue in the receiver’s context (i.e., `c_majorityowned_revenue`) should be the sum of the revenue of the corporate entity in the source context (i.e., `c_unconsolidated_revenue`) and the discounted revenue of the majority-owned subsidiaries. There is a subtlety here – we assume that when adding the numbers together, our user Sally first discounts them using IBM’s ownership percentages on these subsidiaries. For example, because International Information Products is only 80% owned by IBM, Sally would multiply IIP’s revenue number by 80% before adding it to the total revenue of IBM. This is slightly different from what has been presented in the motivational example, but nevertheless, it is another interesting and reasonable way of consolidating revenues. Using this slightly modified aggregation rule, Sally will get 80,946,000 as the total revenue of IBM in the context of `c_majorityowned_revenue`.

5.3.7 Elevation

The elevation axioms map the data and data-relationships from the sources to the domain model. There are three steps involved in an elevation process:

- 1) Define a virtual semantic relation corresponding to each relation in the previous section.
- 2) Assign values to each semantic object according to the context of the source.
- 3) Map the semantic objects in the semantic relation to semantic types defined in the domain model.

Recall that the “revenue1” relation has two columns, CorporateEntity and Revenue. The elevated relation corresponding to “revenue1” for the context `c_unconsolidated_revenue` looks as follows (in COIL).

```
revenue_p(
    skolem('CorporateEntity', C1, c_unconsolidated_revenue, 1, revenue1(C1, C2)),
    skolem('Revenue', C2, c_unconsolidated_revenue, 2, revenue1(C1, C2)))
```

The semantic relation “revenue_p” is defined on the semantic objects in the corresponding relation attributes. The columns in relation “revenue” are mapped to semantic objects, which have a unique object-id: the first column is mapped to ‘CorporateEntity’ and the second column is mapped to ‘Revenue’. Similarly, we define other elevation axioms in COIL:

```
structure_p7(
    skolem('CorporateEntity', C1, Ctxt, 1, structure(C1, C2, C3, C4)),
    skolem('CorporateEntity', C2, Ctxt, 2, structure(C1, C2, C3, C4)),
    skolem('Relationship', C3, Ctxt, 3, structure(C1, C2, C3, C4)),
    skolem(basic, C4, Ctxt, 4, structure(C1, C2, C3, C4)));
```

```
country_p(
    skolem('CorporateEntity', C1, Ctxt, 1, country(C1, C2)),
    skolem('Country', C2, Ctxt, 2, country(C1, C2))).
```

We can summarize the elevation as follows.

Column	Semantic Type
revenue1.CorporateEntity	CorporateEntity
revenue1.Revenue	Revenue
structure.childEntity	CorporateEntity
structure.parentEntity	CorporateEntity
country.CorporateEntity	CorporateEntity
country.Country	Country

Table 4: Summary of Elevations from Relations to the Domain Model

5.3.8 Query Mediation

In this section, we will go through the steps in query mediation and execution using a demo application derived from the motivational example. Recall that the source context (the context that the data source “revenue1” uses) is `c_unconsolidated_revenue`, and the receiver context (the context

⁷ As noted before, for demonstration purposes, “structure” is coded as facts, so the database table and its elevation are not used in the current implementation.

that Sally is in) is `c_majorityowned_revenue`. Because Sally would like to find out what IBM's total revenue is according to her consolidation rules, she issues the following query on "revenue1":

```
Select CorporateEntity, Revenue from revenue1
  where CorporateEntity = "International Business Machines".
 [context= c_majorityowned_revenue]
```

the COIN system must convert this into the query:

```
Select "IBM" as CorporateEntity, SUM(Revenue) as Revenue from revenue1
  where CorporateEntity in ("International Business Machines", "Lotus Development", "IBM Far
  East Holdings", "International Information Products", "IBM International Treasury Services").
```

Here, since ownership percentages are used to discount the revenue numbers, the final result should be 80,946,000. As the first step after a SQL query is fed in, the COIN system generates an internal datalog query⁸ as follows:

```
answer("International Business Machines", 'V1'):-
  revenue1("International Business Machines", 'V1').
```

Then, a context-sensitive datalog query is produced, using elevation axioms and contexts defined in above sections. This query ascertains that the result returned to the user has be in the `c_majorityowned_revenue` context:

```
answer ('V4', 'V3'):-
  revenue1_p('V2', 'V1'),
  value('V2', c_majorityowned_revenue, 'V4'),
  'V4' = "International Business Machines",
  value('V1', c_majorityowned_revenue, 'V3').
```

The above unmediated query is then fed to the mediation engine, where conflicts are detected and resolved. The mediation process is based on an Abduction Engine, which takes the datalog query and the domain model axioms (such as the conversion function presented in Section 5.3.6), and computes a set of abducted queries that have considered all the possible cases of conflicts. Modifier values in the source and receiver contexts, as well as the conversion functions between these two contexts are discovered.

The mediated datalog query produced by the Context Mediator is shown below, and also in Figure 7:

```
answer("International Business Machines", 'V25'):-
  revenue1("International Business Machines", 'V24'),
  revenue1("Lotus Development", 'V23'),
  revenue1("IBM Far East Holdings B. V.", 'V22'),
  revenue1("International Information Products", 'V21'),
  'V20' is 'V21' * 80, 'V19' is 'V20' / 100, 'V18' is 'V19' + 'V22', 'V17' is 'V23' * 100, 'V16' is 'V18' * 100, 'V15'
  is 'V17' + 'V16', 'V14' is 'V15' / 100,
  revenue1("IBM International Treasury Services", 'V13'), 'V12' is 100 * 'V13', 'V11' is 'V12' * 33,
```

⁸ Datalog query representation is used internally in COIN

'V10' is 'V12' * 14, 'V9' is 'V12' * 10, 'V8' is 'V12' * 18, 'V7' is 'V12' * 25, 'V6' is 'V11' + 'V10',
 'V5' is 'V6' + 'V9', 'V4' is 'V5' + 'V8', 'V3' is 'V4' + 'V7', 'V2' is 'V3' / 10000, 'V1' is 'V14' + 'V2',
 'V25' is 'V1' + 'V24'.

If the above mediated datalog query is expanded by substituting values (e.g. “V31”, “V32”) with more meaningful notations such as Revenue(“Lotus Development”) and Revenue(“International Business Machines”), we get the following equation:

$$\begin{aligned}
 \text{Revenue of IBM} &= && \text{R(IBM)} \\
 \text{(in c_majorityowned} &+ && \text{R(Lotus)*100\%} \\
 \text{_revenue context)} &+ && \text{R(International Information Products)*100\%*80\%} \\
 &+ && \text{R(IBM Far East holdings)*100\%} \\
 &+ && \text{R(IBM International Treasury Services)*100\%*33\%} \\
 &+ && \text{R(IBM International Treasury Services)*100\%*14\%} \\
 &+ && \text{R(IBM International Treasury Services)*100\%*10\%} \\
 &+ && \text{R(IBM International Treasury Services)*100\%*18\%} \\
 &+ && \text{R(IBM International Treasury Services)*100\%*25\%,}
 \end{aligned}$$

where R(X) denotes the revenue of entity X in the revenue1 table (i.e., in the c_unconsolidated_revenue context). This equation verifies that the mediated query does give the desired sum of revenues, discounted by their ownership percentages.

After the mediated datalog query is generated, it is translated to a SQL statement (shown in Figure 8) through a Query Planner and Optimizer. This SQL statement, unlike the original input query, takes into account the differences between source and receiver contexts and will return a result in the receiver’s context.

Finally, this SQL query is performed on the data source “revenue1”, and IBM’s total revenue is returned. As mentioned before, the revenues of IBM’s subsidiaries are discounted by the ownership percentages before they are added to IBM’s total revenue. Therefore, the result of the consolidation should be:

$$77,966,000 + 970,000*100\% + 550,000*100\% + 1,200,000*80\% + 500,000*100\% = 80,946,000,$$

which is consistent with the result returned after execution, as shown in Figure 9.

6 Conclusions

In this paper, we briefly stated the importance and challenges of corporate householding, described categories of corporate householding problems, and illustrated a few applications areas with examples derived from corporate householding knowledge research. Then we presented a motivational example in account consolidation. Following that, we described the COntext INterchange technology that performs mediated data access among heterogeneous data sources. By extending the COIN model, we developed a technical solution to an important type of corporate householding problem – entity aggregation, and demonstrated the concept by going through the design and implementation of an application derived from the motivational example. The corporate householding query processor needs to be further improved and extended to serve more areas of application – but the feasibility of this approach has been demonstrated.

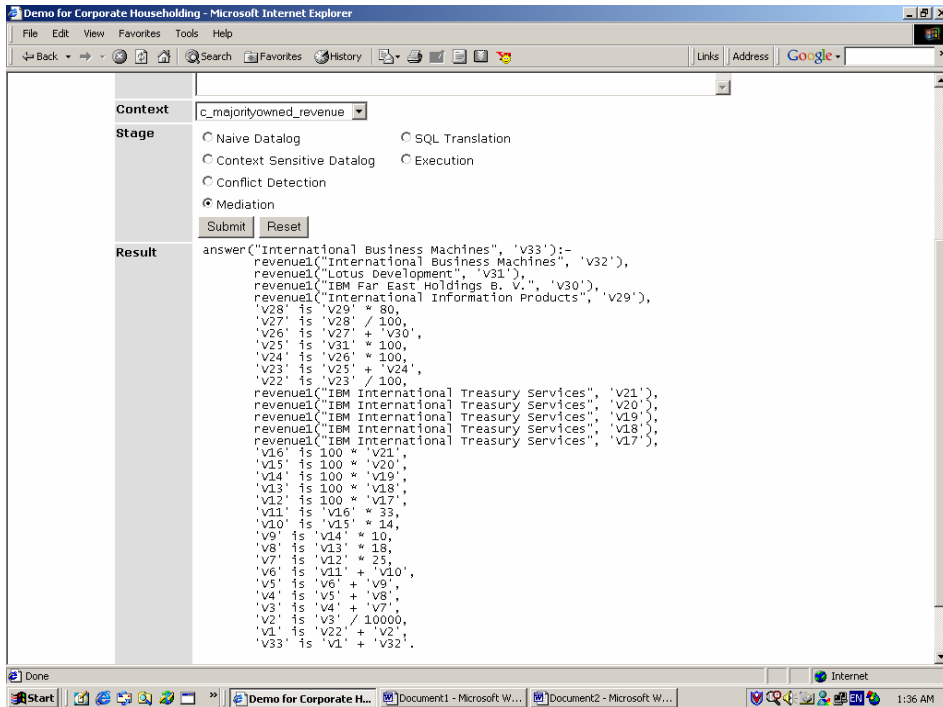


Figure 7: Demo (1) – The Mediated Datalog Query

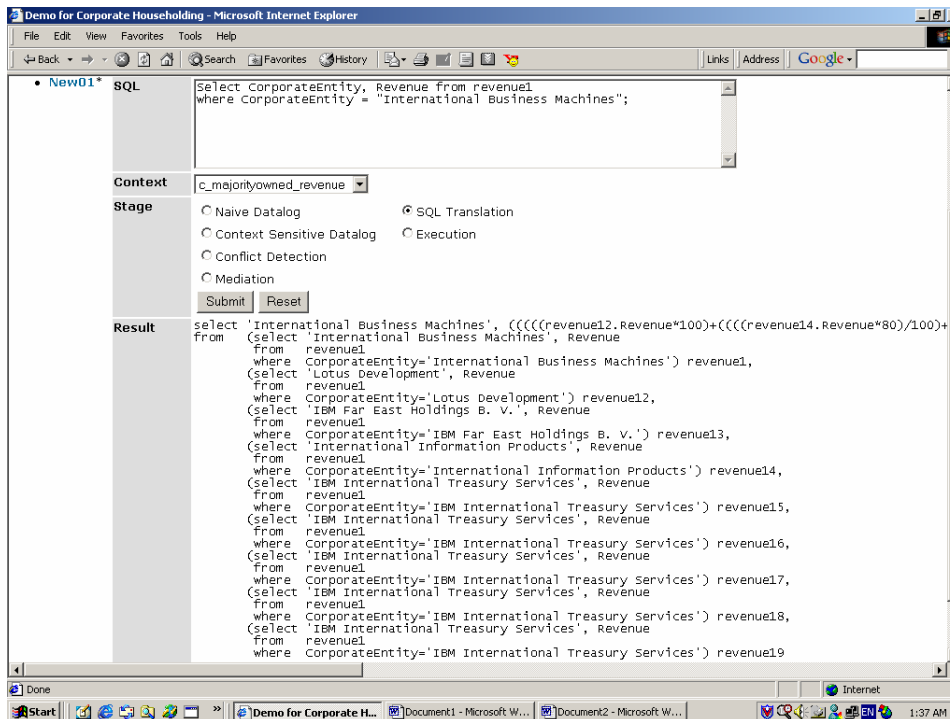


Figure 8: Demo (2) – Result of SQL Translation

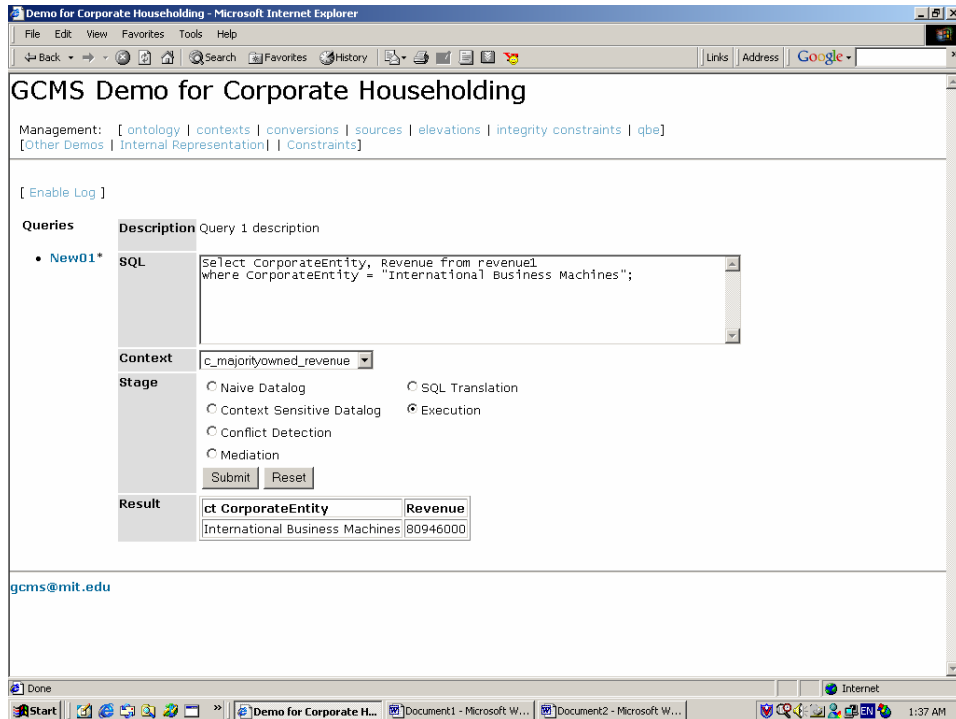


Figure 9: Demo (3) – Result of Execution

7 Acknowledgements

Work reported herein has been supported, in part, by Cambridge Research Group (CRG), D & B, Firstlogic, and Naval Inventory Control Point (NAVICP), Singapore-MIT Alliance (SMA), and Total Data Quality Management (TDQM) Program. Helpful suggestions from Xing Ping Chen, Krishna Chettayar, Frank Dravis, James Funk, Raissa Kats-Haas, Chris Haywood, Cindy Lee, Yang Lee, Pat McCoy, Ahmad Shuja, and Wei Zhang are greatly appreciated.

8 References

1. 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, 1999.
2. Madnick, S., *Metadata Jones and the Tower of Babel: The Challenge of Large-scale Semantic Heterogeneity*, in the proceedings of the 1999 IEEE Meta-Data Conference, 1999.
3. Bressan, S., Goh, C.H., Levina, N., Shah, A., Madnick, S., and Siegel, M., *Context Knowledge Representation and Reasoning in the Context Interchange System*, Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-solving Technologies, Vol. 12, Num.2, p.165-179, 2000.
4. Madnick, S., and Wang, R. Y., *Corporate Household Knowledge Processing: Challenges, Concepts, and Solution Approaches*, 2001. MIT Sloan School of Management.
5. Madnick, S., Wang, R., Dravis, F., and Chen, X., *Improving the Quality of Corporate Household Data: Current Practices and Research Directions*. In 6th International Conference on Information Quality (ICIQ), 2001.
6. Firat, A., Madnick, S., and Grosz, B., *Knowledge Integration to Overcome Ontological Heterogeneity: Challenges from Financial Information Systems*, 2002

7. Firat, A., Madnick, S., and Grosz, B., *Financial Information Integration in the Presence of Equational Ontological Conflicts*, 2002
8. Madnick, S., Wang, R. Y., and Zhang, W., *A Framework for Corporate Householding*, in the proceedings of the 7th International Conference on Information Quality (ICIQ), 2002.
9. Madnick, S., Wang, R., Chettayar, K., Dravis, F., Funk, J., Katz-Haas, R., Lee, C., Lee, Y., and Xian, X., *Exemplifying Business Opportunities From Corporate Household Research*, 2003
10. Wang, R. Y., Allen, T., Harris, W., and Madnick, S., *An Information Product Approach for Total Information Awareness*, 2003
11. Shankaranarayan, G., Ziad, M., and Wang, R. Y., *Managing Data Quality in Dynamic Decision Environments: An Information Product Approach*, 2002.
12. Bielecki, T. R. & Rutkowski, M., *Credit Risk: modeling, valuation and hedging*. Springer-Verlag, 2002: p.3.
13. Epstein, David G., *Bankruptcy and Related Law in a Nutshell*. 6th edition. West group, 2002: p.126-127.
14. Lee, P., COIN Textual Interface Manual, 2003.