

**Corporate Householding Knowledge Engineering and Processing  
Using Extended COIN**

Xiang Xian

Working Paper CISL# 2003-02

June 2003

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

**This page is blank**

# **Corporate Householding Knowledge Engineering and Processing Using Extended COIN**

by  
Xiang Xian

Submitted to the  
Department of Electrical Engineering and Computer Science

May 23, 2003

In Partial Fulfillment of the Requirements for the Degree of  
Master of Engineering in Electrical Engineering and Computer Science

## **ABSTRACT**

This thesis is divided into two parts. In the first half, I present the current research results from corporate householding knowledge exploration. They include the categories of corporate householding problems, the related research in data quality and family household, commercial approaches to corporate householding problems, and the various areas where corporate householding applies.

In the second half of the thesis, I present the design and implementation of a technical solution to an important type of corporate householding problem – entity aggregation – using a motivational example in account consolidation. The proposed technical approach uses extended COntext INterchange (COIN) technology to manage and process corporate householding knowledge.

Thesis Supervisor: Dr. Stuart Madnick

Title: John Norris Maguire Professor of Information Technology / MIT Sloan School of Management & Professor of Engineering Systems / MIT School of Engineering

## **Acknowledgements**

*This research has been supported, in part, by FirstLogic, Dun & Bradstreet, the MIT Total Data Quality Management (TDQM) Program, the Singapore-MIT Alliance (SMA), and Cambridge Research Group.*

## Table of Contents

1	Introduction.....	3
1.1	Motivation.....	3
1.2	Organization of the Thesis.....	3
2	Theory of Corporate Householding .....	4
2.1	Definitions.....	4
2.2	A Typology for Corporate Householding Problems.....	5
2.2.1	Identical Entity Instance Identification.....	6
2.2.2	Entity Aggregation .....	7
2.2.3	Transparency of Inter-entity Relationships.....	8
3	Related Research and Practices.....	9
3.1	Research on Data Quality.....	9
3.2	Research on Family Household.....	9
3.3	Examples of Commercial Approaches .....	9
3.3.1	FirstLogic .....	10
3.3.2	Dun & Bradstreet.....	10
4	Corporate Householding: Areas of Application.....	12
4.1	Account Consolidation.....	12
4.2	Financial Risks .....	13
4.2.1	Credit Risk.....	13
4.2.2	Bankruptcy Risk.....	15
4.2.3	International/Country Risk .....	15
4.3	Laws and Regulations .....	15
4.3.1	Licensing .....	15
4.3.2	Conflict of Interest.....	17
4.3.3	Regulations and Disclosure .....	18
4.4	Business Management and Operations.....	19
4.5	Summary .....	19
5	A Motivational Example of Entity Aggregation.....	21
6	The COntext INterchange Technology.....	26
6.1	Overview .....	26
6.2	The COntext INterchange Architecture.....	26
6.3	The Extended COntext INterchange System.....	28
6.4	Context Interchange and Corporate Householding.....	28
7	Corporate Householding Query Processor.....	29
7.1	Design and Implementation.....	29
7.1.1	Ontology Framework.....	33
7.1.2	Context and Rules.....	35
7.1.3	Modifiers .....	36
7.1.4	Conversion Functions .....	37
7.1.5	Relations.....	39
7.1.6	Elevations .....	42
7.1.7	Query Mediation and Execution.....	43
7.2	Comparison of Mediation-time and Execution-time Approaches .....	49
8	Conclusions and Future Work.....	50
9	References.....	51
	Appendix A: IBM – List of Majority-owned Subsidiaries .....	53
	Appendix B: IBM – Total Revenue.....	58
	Appendix C: Lotus Development – Net Sales.....	59
	Appendix D: General Motors, Hughes Electronics and EDS – Relationships & Revenues.....	60
	Appendix E: Ontology, Contexts, Relations, Elevations and Conversion Functions of Demo Application .....	61
	Appendix F: Conflict Detection Table (complete version) Returned by the Live Demo .....	69

# **1 Introduction**

## **1.1 Motivation**

Today's business environment evolves rapidly. Corporate group structures and the relationships between corporate entities are becoming increasingly complex and difficult to understand. It is easy to imagine a multi-national company that has hundreds of subsidiaries, regional headquarters, branches and local offices all over the world, each one of them carrying out daily business activities within themselves and with hundreds or even thousands of other corporations. Effective use of knowledge about corporate structures and relationships has become an important issue in designing corporations' strategies and performing business functions, such as consolidating financial statements, managing customer records, and determining beneficial ownership. If the information about a corporation's internal structure as well as its relationships with other business entities is fully understood and well managed, it is a priceless asset of the corporation because the knowledge can bring significant comparative business advantages.

Context plays a large role in deciding how this knowledge about corporate structures and relationships should be interpreted. For example, in order to answer the question "how many employees does IBM have?", we have to first 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 aggregated the employee counts from different groups of entities within this enormous corporation. 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 by reducing time, labor, and cost, as well as in many other ways.

## **1.2 Organization of the Thesis**

This thesis is structured as follows. In Chapter 2, I review the theory of corporate householding – definitions and the three categories of corporate householding problems. Chapter 3 briefly summarizes the related research areas and commercial practices. Subsequently, in Chapter 4, I present the major areas of application that cover many important business sectors, as well as examples from these areas. In Chapter 5, I describe a motivational user scenario of corporate householding, and I review the COnText INterchange technology in Chapter 6. Then in Chapter 7, I outline an engineering approach that models the corporate household knowledge domain and provides a technical solution to answer corporate householding questions using the extended context mediation system. I conclude the thesis in Chapter 8 and suggest future research directions.

## 2 Theory of Corporate Householding

### 2.1 Definitions

Many types of corporate entities exist within a corporation. Some of the most common ones are the parent entity, subsidiary, division, and branch. Each of these is based on either location, or ownership percentage, or product/service categorization. Table 2-1 lists a variety of corporate entities that are related to International Business Machines Corporation. Among them, Lotus Development Corporation is a direct wholly-owned subsidiary of IBM; IBM de Colombia, S. A. is also a wholly but indirectly owned subsidiary (some of it is owned by other wholly-owned IBM companies); International Information Products is a partially-owned subsidiary (80%); IBM Global Services and IBM Software are divisions of IBM; IBM Germany is a foreign branch of IBM; Dominion Semiconductor Company is a company in which IBM has a minority joint venture interest; and MiCRUS is a company where IBM has a majority joint venture interest [4].

International Business Machines Corporation
Lotus Development Corporation
IBM de Colombia, S. A.
IBM Far East Holdings B. V.
International Information Products
IBM Global Services
IBM Software
IBM Germany
Dominion Semiconductor Company
MiCRUS

**Figure 2-1: Partial List of the Various Corporate Entities Related to International Business Machines Corporation**

Corporate structures, commonly represented by corporate structure trees or networks, are able to capture the above kinds of corporate entities and the relationships among them. However, there are other kinds of relationships between corporate entities, especially those across corporations, that are more difficult to represent using structure trees. For example, one corporation normally performs numerous business activities with its customers and suppliers. These interweaving relationships add more complexity to the network among corporate entities, and make it more difficult to comprehend.

A “corporate household” not only includes the information about corporate structures, but also encapsulates the complicated relationships among the corporate entities within and outside the corporation. Analogous to a family household, a corporate household is defined as follows.

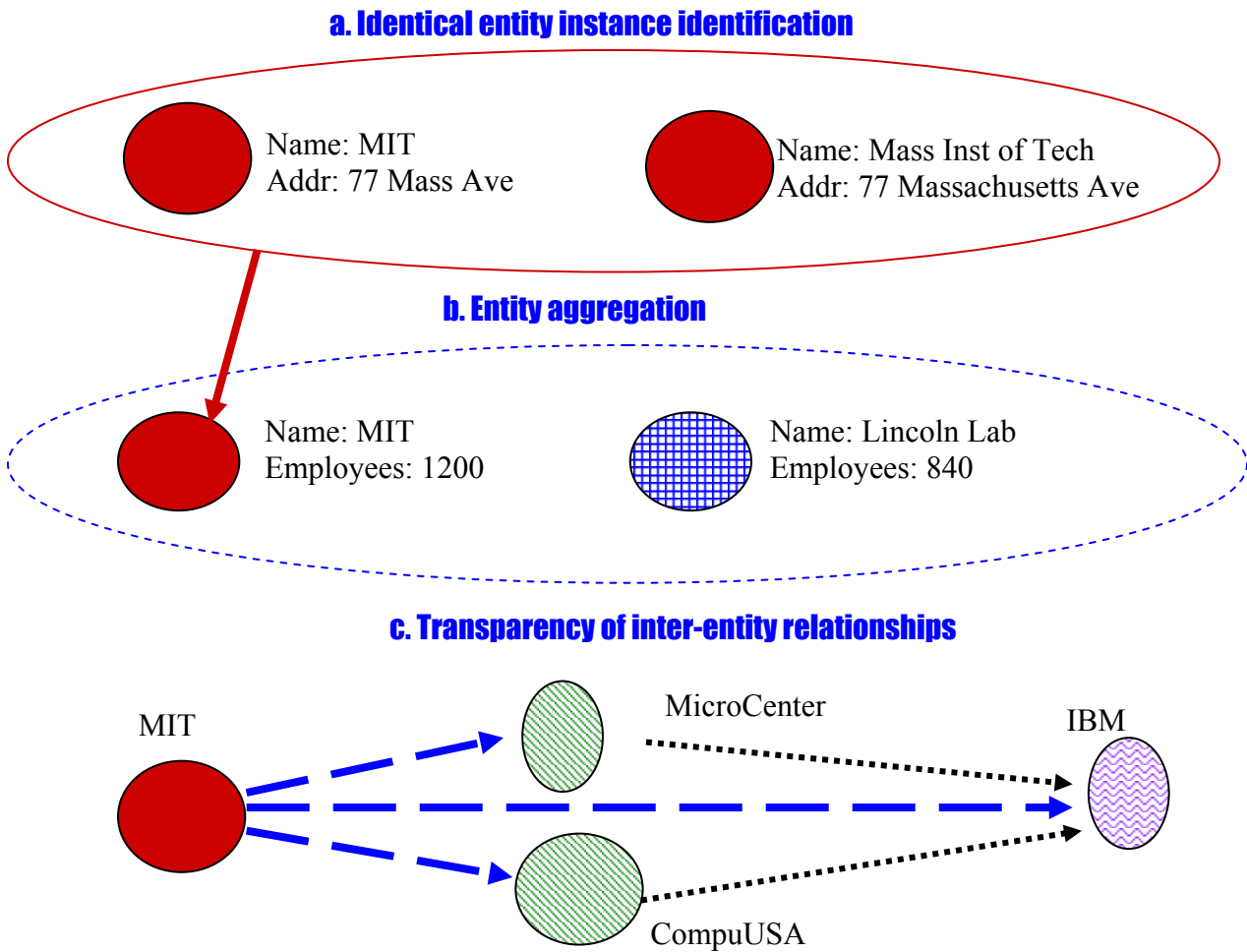
“In the dynamic and rapidly changing e-business environment, any group of persons united or regarded as united with the corporation, such as suppliers and customers whose relationships with the corporation must be captured, managed, and applied for the purpose of activities such as marketing promotion, financial risk analysis, and supply chain management in their entirety forms a corporate household”, [and] “the knowledge developed for such purposes are termed corporate household knowledge.” [5]

Corporate households are dynamic and can overlap each other depending on “the purpose of activities”, as Madnick et al. have pointed out. For example, the corporate household for the purpose of consolidating financial statements of a company may include all of the company’s subsidiaries. But for other purposes, such as identifying the coverage of a software license, the corporate household would usually not include subsidiaries, unless specifically stated in the license agreement. Madnick et al. have also named this kind of process – “the process of identifying the corporate household” – corporate householding, and defined corporate householding knowledge as “the knowledge developed for performing corporate householding.” [8]

Corporate householding knowledge is slightly different from corporate household knowledge in that it includes the reasons underlying the process of identifying a corporate household. In this thesis, I concentrate on corporate householding knowledge, and in the later sections I will present some examples drawn from my research about corporate householding.

## **2.2 A Typology for Corporate Householding Problems**

Problems in corporate householding that a corporation may encounter in its daily business activities come in various forms. Although ambiguities exist in some cases, most of these problems can be categorized into three types, following the typology for corporate householding problems proposed by Madnick et al. in 2002 [8]. The categorization of these commonly encountered problems provides better understanding of corporate householding issues. In addition, because the problems in one category share the same characteristics, we may use similar technical approaches to help solve them. Figure 2-2 [8] illustrates the three categories of corporate householding problems, and the following sections explain each one of them in detail. (In the rest of this thesis, the term “entity” usually refers to a “corporate entity”, unless otherwise stated.)



**Figure 2-2: A Typology for Corporate Householding: Three Categories**

### 2.2.1 Identical Entity Instance Identification

Part of the complexity that is involved in understanding corporate household data results from the multiple representations of the same entity. In general, neither people nor companies have universal identifiers that can work in all circumstances without any ambiguity. As I will explain in detail in Section 3.3.2, the world's leading business information provider, Dun & Bradstreet, has developed the Data Universal Numbering System (D-U-N-S), which assigns a unique I.D. (D-U-N-S number) to every business entity in D & B's databases. This system is helpful, but it does not solve the problem entirely, because: 1) D & B's databases cannot possibly include all the corporate entities in the world, and 2) D-U-N-S numbers are not used universally in all the data storage systems. When heterogeneous data sources are involved, multiple representations of the same corporate entity are unavoidable.



For example, as shown in Figure 2-2a, an entity that has the name “MIT” and the address “77 Mass Ave” is in fact the exactly same entity as the one with the name “Mass Inst of Tech” and the address “77 Massachusetts Ave” – they both refer to the school “Massachusetts Institute of Technology”, whose official street address is “77 Massachusetts Avenue”. The same problem exists for companies such as IBM, which may be represented as “International Business Machines Corporation,” “International Business Machines” or “I.B.M.” in different data sources.

The same identical entity might appear as multiple instances; and the problem of identifying such kind of entities is referred to as identical entity instance identification. Correct and efficient entity identification is difficult, but good progresses have been made in this specific research area and in the commercial world, one of which I will describe in Section 3.3.1.

## 2.2.2 Entity Aggregation

After one has identified that “IBM,” “International Business Machines Corporation,” and “I.B.M” all refer to the same entity, he needs to determine what exactly this entity is. That is, what corporate entities does this intended definition of IBM include? As discussed in Section 2.1, 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 (see Appendix A for a full list of IBM’s subsidiaries). In one situation, 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 other situations, other parts of the tree may be considered.

For instance, the MIT Lincoln Lab, according to its home page, is “the Federally Funded Research and Development Center of the Massachusetts Institute of Technology.” It is located at a separated site away from the main campus of MIT, and has a budget of about \$500 million, which is approximately equal to the rest of MIT. Also, as illustrated in Figure 2-2b, MIT (not including Lincoln Lab) has about 1200 employees, and Lincoln Lab alone has about 840 employees. Suppose that after the entities “MIT” and “MIT Lincoln Lab” have been correctly identified in the data sources, one wants to find out how many employees MIT has or what MIT’s budget is in year 2003. Whether or not Lincoln Lab is taken into account would make a huge difference to the answers of these questions, because of the Lab’s relatively large size compared to the rest of MIT. It is possible that for the purpose of counting employees, Lincoln Lab’s number is not considered, whereas for the purpose of calculating budget, Lincoln Lab’s number is considered. Therefore, when to aggregate which entities within a corporation depends on the task at hand, and in most cases, it is a necessary step in understanding the intended meaning of a name. The differing circumstances are referred to as contexts, and this type of problem is called entity aggregation. (For an in-depth sample scenario of entity aggregation and the technical approach targeted towards this kind of problems, see Chapter 5 to 7.)

### 2.2.3 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. As shown in Figure 2-2c, MIT buys computers from IBM both directly and through local computer stores, such as MicroCenter and CompuUSA. Knowing when these intermediaries are important and when they are not (in this case, they are considered to be “transparent”) poses another type of problem for corporate householding, which also has to be addressed depending on the context. This type of problem is referred to as transparency of inter-entity relationships.

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 sales records. Then, it needs to make sure to include purchases from entities that may not seem to be directly related to IBM in their names but are in fact part of the corporation, such as IBM’s software subsidiaries Lotus Development and Rational Software. Lastly, purchases of IBM products through brokers like those mentioned in the above paragraph should also be considered. The second type of corporate householding problems, entity aggregation, is the focus of the later chapters in this thesis.

## **3 Related Research and Practices**

### **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 the 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]. This 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. An applied research of the TDQM program is *corporate householding*, which aims at better understanding and utilizing corporate household data.

### **3.2 Research on Family Household**

The concept of corporate household derives from the conventional meaning of a household, which 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) to hundreds of thousands of people and organizations. It 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, and can sometimes be very complex as well. 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, software solutions aimed to identify family households have been developed.

### **3.3 Examples of Commercial Approaches**

Some industry leaders in corporate data management and knowledge engineering have implemented solutions to corporate householding problems, with focuses on certain aspects of the problems. These solutions have weaknesses, but they represent the state-of-art practices in the industry and are very helpful tools in their specific targeted fields. In

the following sections, I will briefly review the commercial approaches provided by two companies – FirstLogic, Inc. and Dun & Bradstreet.

### **3.3.1 FirstLogic**

FirstLogic, Inc. is a software product and services company aimed to help businesses improve the quality of their information and to identify, understand, target, and deliver to customers more effectively. It uses the Subject Matter Experts (SME) approach to identify entities correctly and efficiently [5]. 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 databases. The detailed steps are as follows.

- Establish project goals together with the clients.
- Define applicable terms and gain cross-functional agreement on these terms.
- Define the business rules that attain the project goals, such as the rules for identifying duplicates and the confidence levels of duplicate records.
- Create and verify the rule matrix. Extracting the rules into one matrix makes it easy to view and evaluate them.
- Create application parameters from the rule matrix.
- Check rules for cohesion.
- Verify entire match sets and consolidation criteria.
- Run the householding process after all the consolidation criteria have been tested and confirmed.

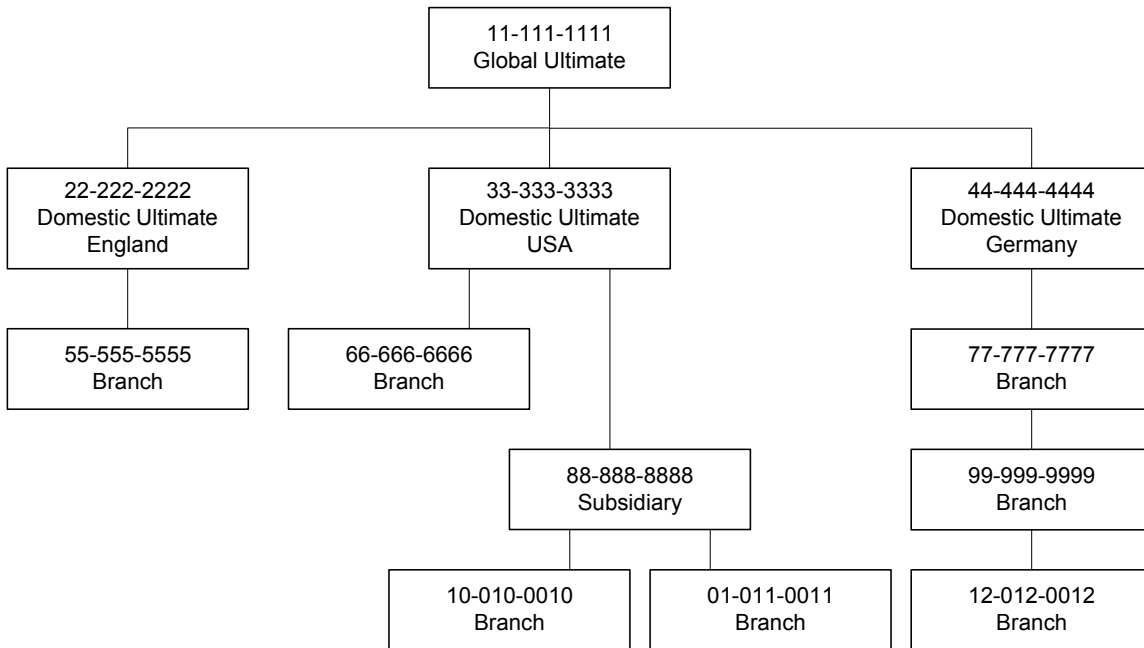
The SME approach employed by FirstLogic consultants in conjunction with the inputs from their clients provides a good corporate data structure capable of supporting various business purposes at different levels. However, this approach also has some weaknesses. For example, rendering the householding process can be very time-consuming and costly. In addition, FirstLogic’s solutions are usually client-specific and thus hard to generalize.

### **3.3.2 Dun & Bradstreet**

Dun & Bradstreet (D&B) has developed a representation of corporate structure to improve the understanding of relationships among corporate entities. 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 corporate group structures and their relationships worldwide. The D&B Family Tree is comprised of linkages and business relationships. Linkage, which is the relationship between different companies or specific sites within a corporation, occurs when one corporate entity has financial and/or legal responsibility for

another corporate entity. Some other types of relationships are not captured in the Family Tree [5].

As shown in Figure 3-1, the D&B Family Tree 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 [5].



**Figure 3-1: A Sample Dun & Bradstreet Family Tree**

D&B's approach captures a significant amount of useful information about a corporation, but there are limitations to it. For example, the application domains of Corporate Householding are much broader than what the D&B family tree covers – financial and legal. Also, this system for identifying corporations may miss some data in companies' internal databases. For example, any subsidiaries that are owned less than 50% by parents are not listed in the parents' family trees. Although the D-U-N-S numbers and the D&B family tree can represent a major part of corporate structure data, they do not embed the corporate householding knowledge.

## 4 Corporate Householding: Areas of Application

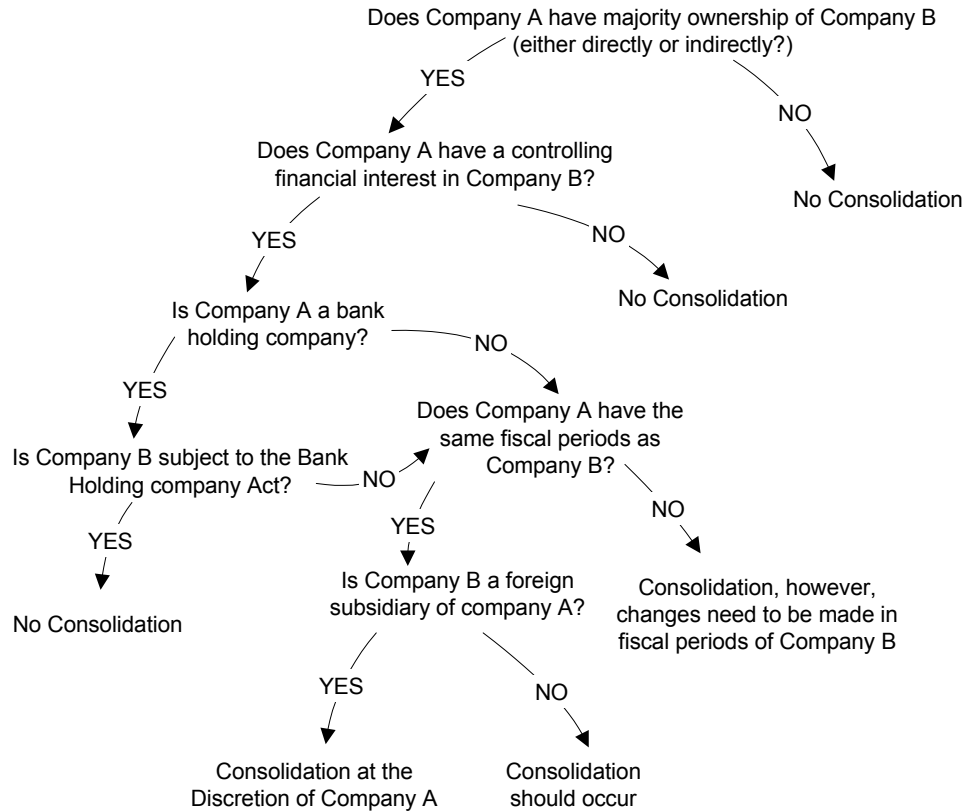
Building on the discussions in Chapter 2 and 3, this chapter presents the common areas of application of corporate householding. The examples included below are drawn from a review of the literature and interviews with subject experts [9]. Many of the examples are not industry specific – any corporations 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 IBM – 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 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 Lotus’ financial statements.

Suppose that IBM directly owns 40% of Lotus and owns another 20% of Lotus through multiple layers of ownership (referred to as “indirect ownership”). Should consolidation of financial statements still occur? Under the SEC regulations, the existence of a parent-subsidiary 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 may be multiple layers of subsidiaries involved, such as 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. Figure 4-1 summarized SEC’s rules of account consolidation.



**Figure 4-1: The SEC Rules on Account Consolidation: Should Company A Consolidate Its Financial Statements with Company B?**

Not only does the parent company IBM need to have and to understand knowledge about its relationships with its subsidiaries – other companies that have business linkages with IBM also need to grasp this information. The challenge for these other companies is how to collect, organize, and retrieve the data on IBM from the available data sources that may have semantic differences. In Chapter 5, I will use a simplified scenario of retrieving IBM’s revenue to demonstrate the concepts of source and receiver contexts and the properties associated with them.

## 4.2 Financial Risks

### 4.2.1 Credit Risk

Credit risk, a crucial consideration in many financial transactions, can be defined as “the possibility that a contractual counter-party does not meet its obligations stated in the contract” (in other words defaults), “thereby causing the creditor a financial loss” [16]. In a more general sense, credit risk is “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]. Because of the complexity of corporate structures and relationships and the importance of credit risk,

corporate householding in this field requires a significant amount of effort and attention. Following the categorization of corporate householding challenges as stated in Section 2.2, two major types of problems may occur in the process of credit risk evaluation as explained below.

- 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.

Looking at credit risk evaluation from another angle, the dynamics of corporations over time also cause corporate householding problems. 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.

As mentioned before, the relationships among corporate entities can normally be represented by a multi-dimensional tree structure. When banks evaluate a corporation's risks, they draw such a tree to represent the corporation and its surrounding entities. The bank considers the credit-worthiness of every entity, and assigns credit limits to those entities. By law, banks must come up with credit concentration limits. Normally each bank comes up with the risk structure for itself. Banks keep their internal risk rating system to themselves because they can use some of the information obtained from their private channels to achieve competitive advantages. However, in some large deals, major credit risk is performed by a group of financial institutions pooling their information as part of the decision-making process. By merging their views of the entity hierarchy, the financial institutions have a more accurate view. Given the complexity of varying relationships and points of view, it is common for risk managers and analysts to have disagreements on questions such as ownership relationships and what the credit limit should be. Without trustworthy and timely corporate householding knowledge, significant mistakes could result from the banks' decisions.



## **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 its 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/Country 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. This risk is called 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?

Another type of international risk involves the constant changes in exchange rates. When a company conducts business with foreign or multinational corporations, it’s highly likely that the currencies are those of the host country, e.g. US dollars for American companies and Japanese Yen for Japanese companies. Exchange rates fluctuate constantly—keeping up with these minute-to-minute changes that affect hundreds and thousands of companies is an incredible challenge.

## **4.3 Laws and Regulations**

### **4.3.1 Licensing**

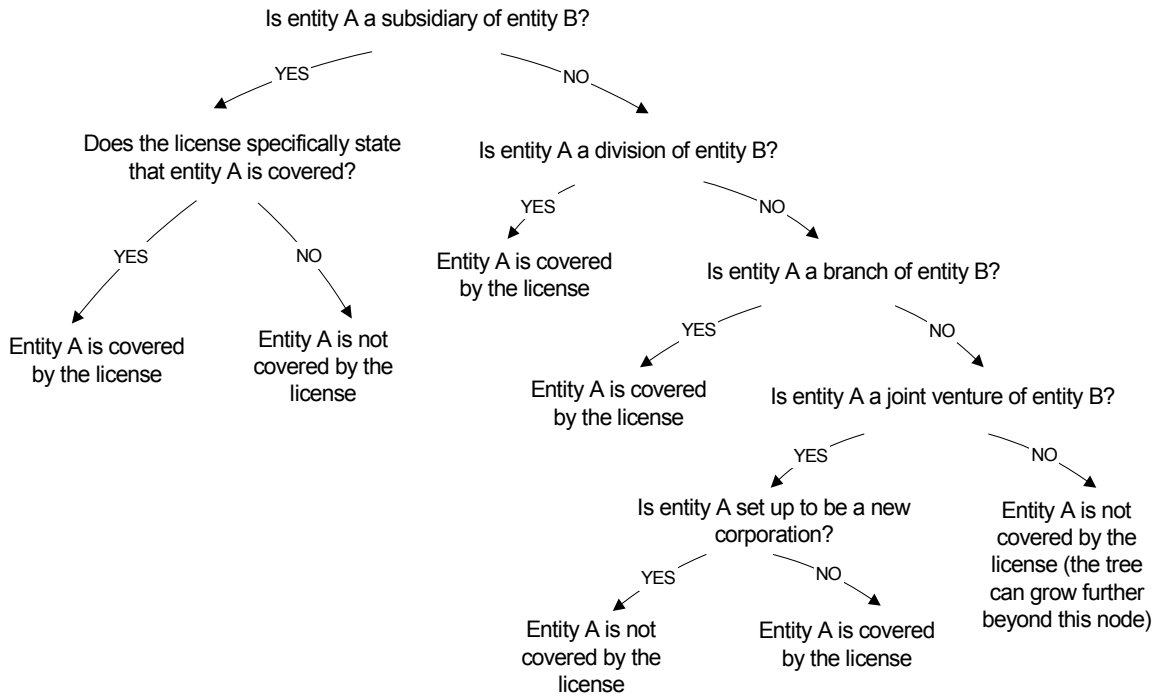
There are many types of corporate householding activities in the legal domain as well, one of which has to do with the coverage of software licenses. For example, if MIT buys

a company-wide license for Windows XP from Microsoft, is Lincoln Laboratory authorized to use this software too? Must the license specifically state the cases that would apply to the Lincoln Lab? If not, what is the common guideline (if any) that could resolve the ambiguity in the license? The same kinds of issues exist in other licensing arrangements, such as licensing of patents.

From an interview with a technology licensing attorney, an answer to part of the problem is found. Whether or not Lincoln Lab is covered in the software license depends on if Lincoln Lab is a legally separated entity with respect to MIT. Parent and subsidiaries are independent legal entities, but corporations and their divisions are not. In this case, because Lincoln Lab is not a separate legal entity by itself, the entity "MIT" should include Lincoln Lab. Since parent and subsidiary are legally separated entities, if one would like to include the subsidiaries, he has to specifically state "...and each of their subsidiaries" or the names of some (not all) of the subsidiaries in the contract. On the other hand, as mentioned before, all the divisions of a company (not legally separated entities) are automatically included if the contract only mentions the name of the company.

Here, a further question is what is the definition of a legally separated entity. A legally separated entity has independent legal standing. Some common types are "corporation", "partnership", and "LLP". Parent and subsidiary are legally separated entities, but divisions are not legally separate from the corporation. Joint ventures are case dependent. In some joint ventures, two sides are only sharing resources, whereas in other cases, they set up a new corporation, which is a legally separated entity by itself. Figure 4-2 illustrates the rules of determining software license coverage.

Understanding corporate structures and relationships is valuable not only to the licensees, but also to the software vendors. For example, when the consolidation of customers through mergers and acquisitions takes place, vendors need to keep their information current, which appears to be a quite challenging task. Also, vendors that sell to multinational corporations that have very extensive corporate family trees may find it beneficial to identify which entities in the family trees they have sold licenses to, and under what restrictions. This not only allows them to roll-up all revenues for their global customers and produce a single customer view, but also allows them to propose the optimal licensing configuration. Obviously, vendors do not want to quote an enterprise-wide license that would mistakenly distribute their software for free to all the corporate entities under the global parents.



**Figure 4-2: The Rules on Software License Coverage: Is Entity A Automatically Covered by the Software License Purchased by Entity B?**

### 4.3.2 Conflict of Interest

In industries such as accounting and consulting, there are legal or professional obligations to avoid conflict of interest. The various forms of conflict of interest include having ownership interests or non-audit business activities in a company that is being audited, and simultaneously consulting for a competitor. An understanding of corporate householding can help firms who would like to sell non-audit services to their clients to compensate for declining revenues avoid violations for the SEC rules on independence of auditors.

In Regulation S-X, Article 2, SEC provides a general definition of auditor independence, that is, an accountant is independent from its audit client only if it is “capable of exercising objective and impartial judgment on all issues encompassed within the accountant’s engagement.”<sup>1</sup> SEC then presents a complex set of rules to reflect the application of this definition in specific situations, such as financial relationships, employment relationship, business relationships, and non-audit services.

The rules for maintaining independence in employment and business relationships are relatively straightforward: an accounting firm is not independent from its audit client if the client currently employs a partner or shareholder; it is also not independent if it

<sup>1</sup> Source: <http://www.sec.gov/divisions/corpfin/forms/regsx.htm>.

engages in any direct or material indirect business relationship with the client. In the area of financial relationships, rules defining auditor independence are more complex. For example, suppose that a firm is auditing the financial reports of IBM. Is it independent if it owns shares of Lotus? Is it independent if it is invested in IBM through an intermediary, such as an investment portfolio? According to the SEC rules, an accountant is not independent if s/he has a direct financial interest or a material indirect financial interest in the audit client. Thus, owning securities in an audit client (or a subsidiary of the client) would render an accountant not independent. So an accountant is not independent of IBM if it owns Lotus' stocks. On the other hand, if IBM were to own shares in the accounting firm, auditor independence would be violated too. However, if an accountant holds 5% or less of the shares of a diversified management investment company that invests in the audit client, then auditor independence is maintained. Therefore, in most cases, indirect investment in IBM through an investment company would not compromise an auditor's independence. Accounting firms also have restrictions in the types of non-audit services that they can provide to their clients, if they wish to maintain independence. The Sarbanes-Oxley Act of 2002 lists possibly prohibited activities including bookkeeping of audit clients' financial statements, actuarial services, broker services, and financial information systems design. Again, following these rules can be difficult if they involve corporate entities that are related. For instance, can the accounting firm provide non-audit services to Lotus while being the auditor of IBM? What if IBM only owned a small percentage of Lotus? Corporate householding can increase the understanding of, and compliance with, the complex rules in the area of conflict of interest.

### **4.3.3 Regulations and Disclosure**

SEC also has numerous rules concerning disclosure of information by publicly-traded companies. For example, it requires an individual or a group of individuals to disclose information if they acquire beneficial ownership of more than 5% of a class of a company's equity securities registered defined in the Securities Exchange Act of 1934. Beneficial ownership is defined as having voting power or investment power (ability to sell the securities) either directly or indirectly. In these cases, individuals must report their ownerships and SEC will make this information available to public for viewing who exercise control in a company.

Corporate householding play a role in disclosure when one tries to determine if an investor has beneficial ownership of 5% or greater of a company's securities. There are a series of rules in this area to objectively define beneficial ownership. For example, any person who has power of attorney over a trust fund that owns 5% of a company's equities should be declared the beneficial owner. The issue becomes more complicated when different definitions of "beneficial ownership" and different requirements of disclosure in various countries need to be taken into account, because the laws defining persons who are required to file a disclosure statement vary from country to country. If corporate householding is performed efficiently, companies can improve their ability to quickly determine beneficial ownership, and thus comply with the SEC regulations.

## **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.

## **4.5 Summary**

Corporate householding application areas span almost all business functions in a corporation. This categorization of application areas, combined with the typology for

corporate householding problems described in Section 2.2, forms a rather complete two-dimensional view of corporate householding problems. For instance, determining IBM's revenue in 2002 is a corporate householding problem in the area of account consolidation, and is also a kind of entity aggregation problem. On the other hand, consolidating customer records belongs to the area of customer relationship management, and is an example of entity identification problems.

In addition, there are other dimensions that can add more complexity to corporate householding problems, two of the obvious ones being time and location:

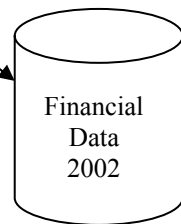
- Corporate structures and relationships evolve over time; business rules in different application domains also change in time. The problem about bond status downgrade mentioned in Section 4.2.1 Credit Risk is an example of the effect of time on corporate householding.
- For corporations in different locations or entities within the same corporation but located around the globe, differences between local regulations or ways of conducting business or even currencies can complicate the corporate householding process. The international risk described in Section 4.2.3 is an example of problems due to differences in locations of corporate entities.

As mentioned in Section 3.3, FirstLogic has developed commercial approaches to tackle entity identification problems, and D&B has provided ways to represent corporate structures and inter-entity relationships such as the parent and subsidiary relationship. In the following chapters, I will propose a different approach for solving the second type of corporate householding problems – entity aggregation, an approach based on the COntext INterchange (COIN) technology developed at MIT, and I will use account consolidation as the application area.

## 5 A Motivational Example of Entity Aggregation

Now 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 31<sup>st</sup>). International Business Machines Corporation is a giant organization with about 100 years of history and numerous branches and offices around the globe. In addition, a century of mergers and acquisitions have created more than 100 subsidiaries<sup>2</sup> 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 5-1 does not include the revenues of Lotus, IBM Far East Holdings, International Information Products and IBM International Treasury Services.

Receiver: Sally



Source Table: revenue1

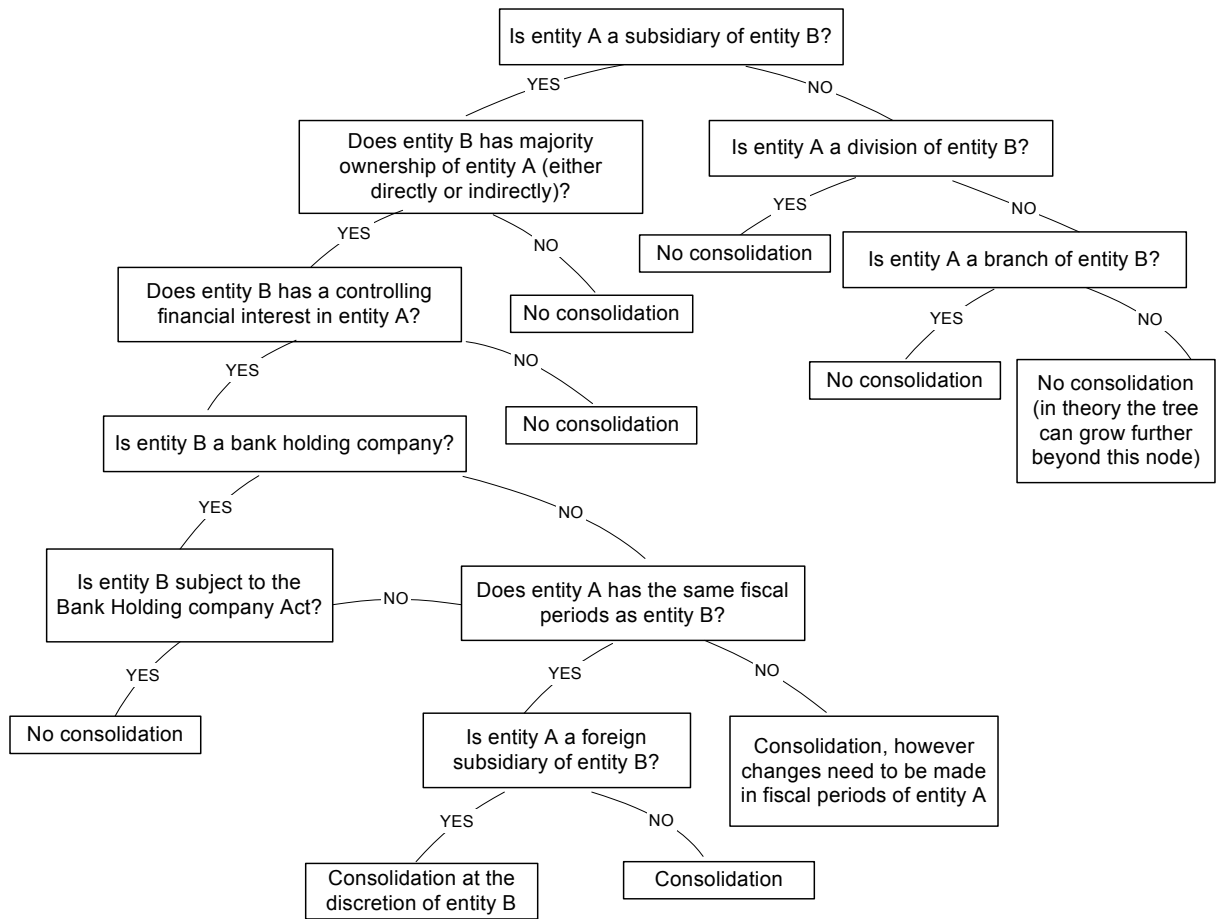
CorporateEntity	Revenue <sup>3</sup>
International Business Machines	77,966,000
IBM Global Services	36,360,000
Lotus Development	970,000
IBM Far East Holdings	550,000
International Information Products	1,200,000
IBM International Treasury Services	500,000
General Motors	177,828,100
Hughes Electronics	8,934,900
Electronic Data Systems	21,502,000

**Figure 5-1: A Motivational Example on Entity Aggregation: Performing a Query for Total Revenue of IBM in Year 2002**

<sup>2</sup> For details, refer to Appendix A: IBM – List of Majority-owned Subsidiaries.

<sup>3</sup> The revenue data is directly extracted or estimated from the revenue/sales data of IBM, Lotus, GM, Hughes, etc. from SEC filings. See Appendix B, C and D.

The source database also includes revenue data on other entities within 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 (the SEC’s rules) better, we summarize them in the decision tree in Figure 5-2. 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.



**Figure 5-2: Decision Tree (simplified) that Represents the SEC’s Rules for Calculating Total Revenue of Corporate Entities such as IBM**

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

*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 are “aggregating” 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 5-2, 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 5-3 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 company<sup>4</sup>, 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 5-3), no consolidation should occur, because IBM Global Services is a division of IBM.

---

<sup>4</sup> 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.

After reasoning through the source data using her accounting rules, Sally knows that she should 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”, which is indeed the total revenue of IBM on the company’s 2002 annual report.

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.

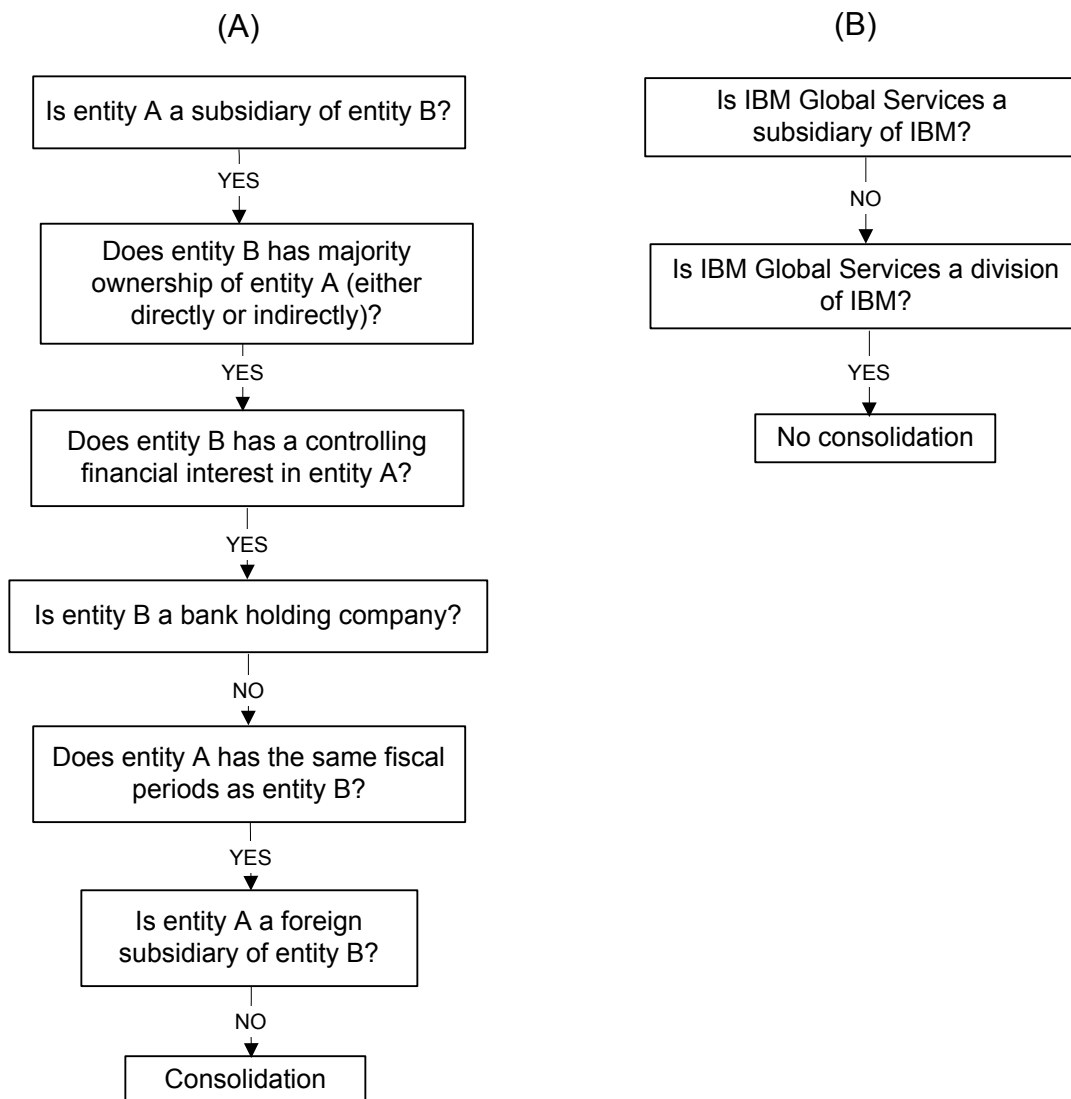
*Select "IBM" as CorporateEntity, SUM(Revenue) as Revenue from revenue1  
where CorporateEntity in ("International Business Machines", "Lotus  
Development Corp", "IBM Far East Holdings", "IBM International  
Treasury Services")*

The result she gets back is

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

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 5-1. 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. We believe that, with some extensions, the COntext INterchange (COIN) system 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 chapter, we will describe the COIN system and the extended COIN in detail.



**Figure 5-3: Steps of Reasoning Using the Decision Tree in Figure 5-2**  
 Entity A = Lotus Development in (A) and IBM Global Services in (B)  
 Entity B = International Business Machines

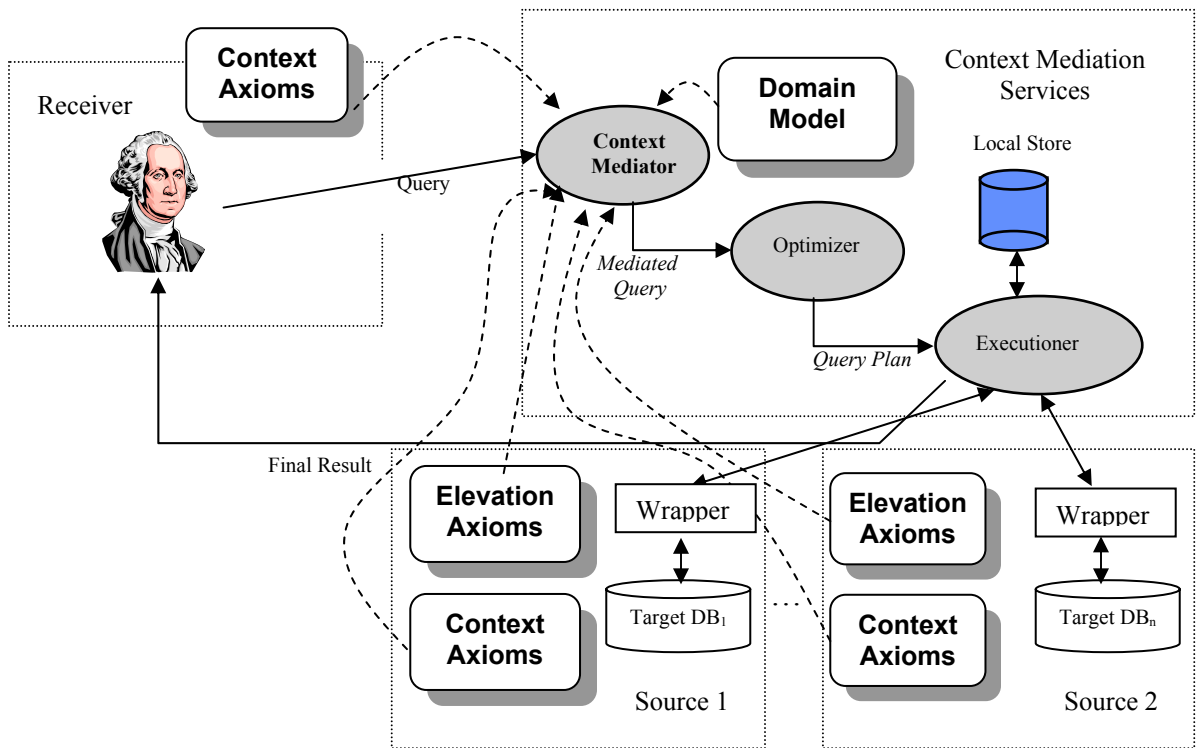
## **6 The COntext INterchange Technology**

### **6.1 Overview**

With the rapid development of information networks, the amount of information that is readily available to people and organizations has been growing at an incredible speed. However, the physical connectivity of information does not necessarily imply that the data can be easily understood and correctly used by the receivers of it, because many of the data sources or the technologies employed by the data source providers do not provide logical connectivity of the available information. Logical connectivity is defined as “the ability to exchange data meaningfully” [3], and it is crucial for achieving semantic interoperability among distributed and heterogeneous data sources. The COntext INterchange (COIN) technology, developed by the Context Research Group at MIT, can capture the knowledge about the source and receivers’ contexts, automatically detect and resolve semantic conflicts, and thus present a meaningful view of the data to the receivers. Context is defined as “the assumptions underlying the way an agent presents or interprets data”[15]. Each context may differ in how data is represented and interpreted; by identifying these differences, data can be exchanged correctly among different contexts.

### **6.2 The COntext INterchange Architecture**

As mentioned above, the COIN technology is a mediation approach for semantic integration of disparate 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 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 6-1: The COIN Architecture**

Figure 6-1 above illustrates the components of COIN. The COIN framework consists of a data model and a logical language, COINL<sup>5</sup>, 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

<sup>5</sup> COINL is a logical programming language based on F-logic, which is based on Prolog.

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 [14]. The essential part of the demo application that I will present in Chapter 7 is mostly written using this interface.

### **6.3 The Extended COntext INterchange System**

The Extended COntext INterchange (ECOIN) system is an extension of the core COIN system, aimed to resolve equational ontological conflicts, which is defined as “the heterogeneity in the way data items are calculated from other data items in terms of definitional equations” [7]. In the original COIN system, new semantic types will need to be introduced in order to handle equational ontological conflicts. This would result an explosion of types and time-consuming changes to the ontology. In ECOIN, modifiers are used to specify the definitional differences, and new constraints for basic mathematical operations such as addition, subtraction and multiplication are added. ECOIN also includes features such as ontology merging and source selection in the presence of equational conflicts. The implementation of corporate householding application presented in Chapter 7 of this thesis utilizes the functionalities in the Extended COIN system, with the underlying concepts based on the core COIN.

### **6.4 Context Interchange and Corporate Householding**

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 ECOIN. In the following chapter, I will define the components of the corporate householding query processor in detail and illustrate how context mediation can be used to solve entity aggregation problems such as the motivational example presented in Chapter 5.

## **7 Corporate Household Query Processor**

### **7.1 Design and Implementation**

In this section, I will present a design of the corporate householding query processor, and explain how the COntext INterchange technology can help mediate (i.e., rewrite) corporate householding queries to better meet users' needs. I will also show the results of query mediation and execution from a live demo. The demo application follows the motivational example in Chapter 5, with some slight modifications to simplify implementation. The purpose of this example is to demonstrate the working of the system, which is scalable and flexible enough to be extended to more realistic cases with a reasonable amount of add-ons and more specification.

Figure 7-1 illustrates how corporate householding knowledge is captured and elevated in ECOIN; Figure 7-2 illustrates the differences between contexts, and what happens when a sample query is asked in difference receiver's contexts. The following sections will describe each of these components (e.g. ontology, elevations, contexts) in detail. A complete version of the actual code that defines these components is included in Appendix E.





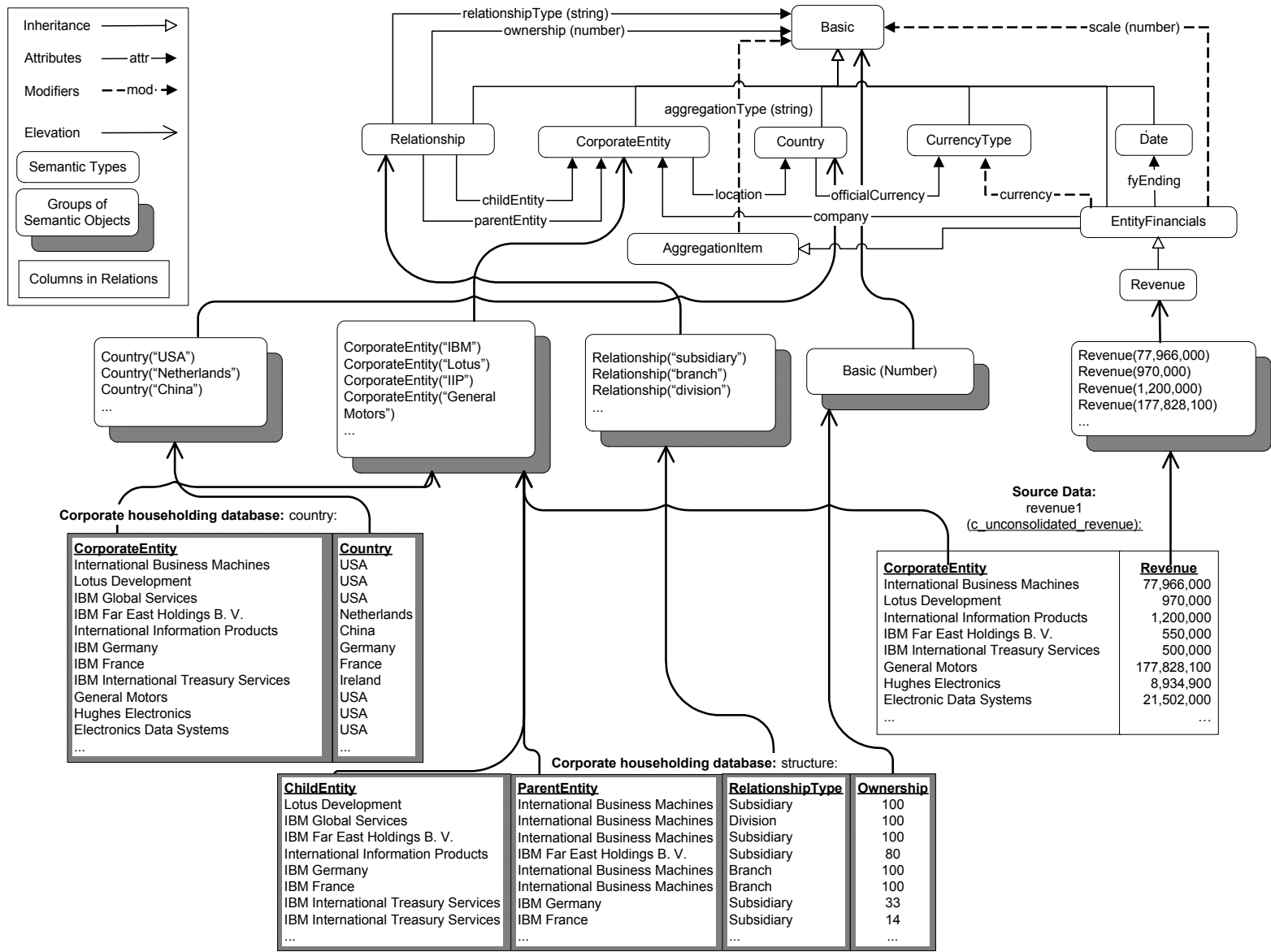
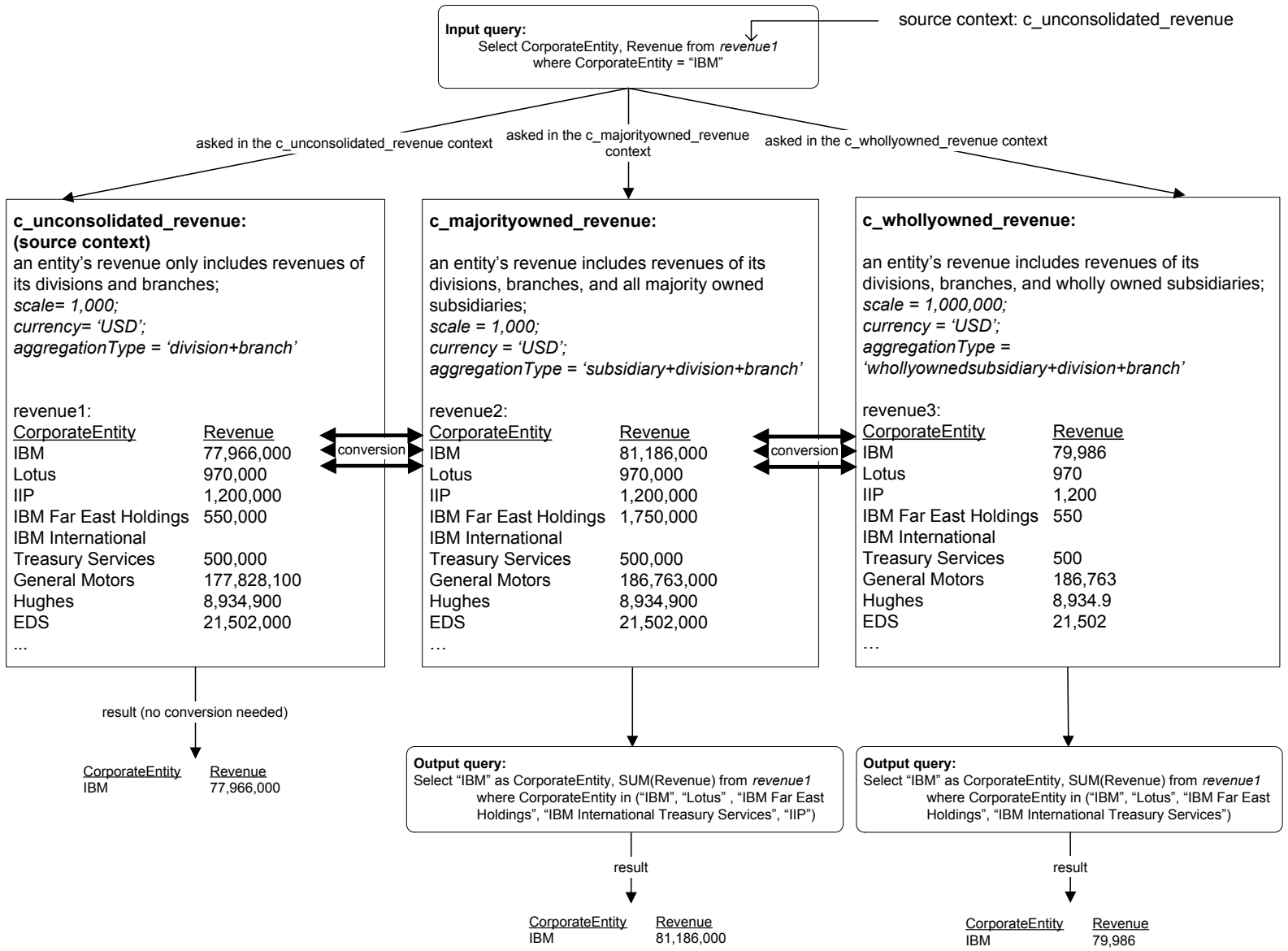


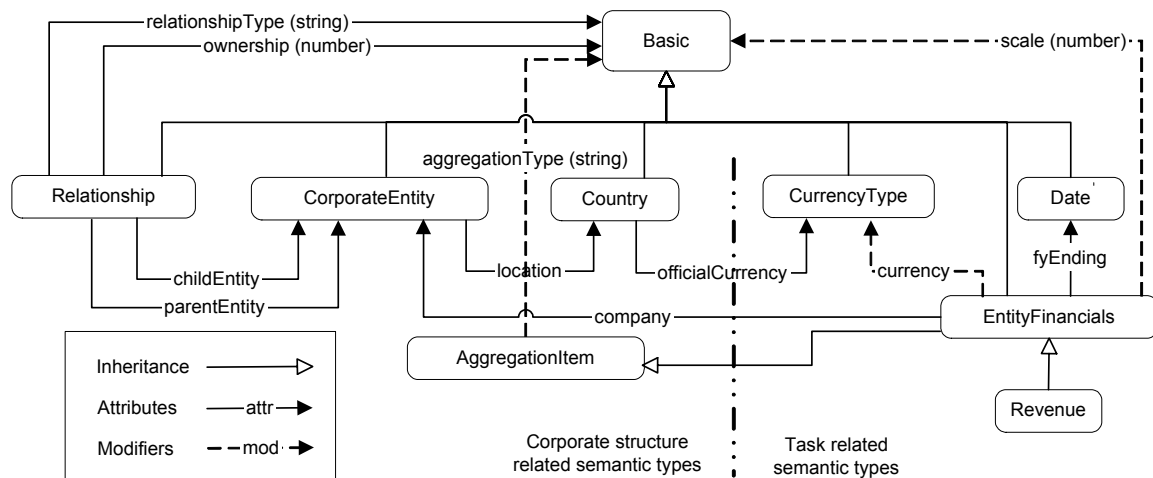
Figure 7-1: Summary of Ontology, Relations and Elevations



**Figure 7-2: Summary of Contexts and Queries**

## 7.1.1 Ontology Framework

First, let us specify a domain model (ontology framework) for the sample corporate householding problem, as shown in the top part of Figure 7-1 and in Figure 7-3 below. 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 7-3: Ontology for the Motivational Example on Account Consolidation**

Three kinds of arrows in Figure 7-3 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 7-3, “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 legally independent) 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 relationships 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”, which is a property of the semantic type “AggregationItem”, 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.

#### 7.1.1.1 Corporate Structure Related 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 Hughes Electronics is a subsidiary 100% owned by General Motors. This relationship can be represented as “childEntity = Hughes Electronics, parentEntity = General Motors, relationshipType = Subsidiary, ownership = 100.” IBM Global Services is a division of IBM, and this relationship can be represented as “childEntity = IBM Global Services, parentEntity = International Business Machines, relationshipType = division, ownership = 100.”

### 7.1.1.2 Task Related 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. Theoretically, different kinds of task related components could be added onto the current ontology model when different problem domains are considered. Here, I 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* (of type Date), as well as modifiers *company* (of type CorporateEntity), *currency* (of type CurrencyType) and *scale* (of type Basic). For example, the fact that Entity A’s fiscal year ends on December 31 and its financial data is in thousands USD is represented by “company = Entity A, fyEnding = 12/31, currency = USD, scale = 1,000.” 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.

### 7.1.2 Context and Rules

Recall the motivational example described in Chapter 5. Sally wants to find out IBM’s total revenue in fiscal year 2002. Here, it does not matter 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. I have assumed that Sally and all the data sources have a common understanding about the meaning of ‘Revenue’, that is, there is no ontological heterogeneity of the semantic type “Revenue”. In addition, other context differences, such as differences in currencies, are handled by other parts of the ECOIN engine.

Let us name the context of the data source that 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 is “c\_whollyowned\_revenue”, which requires account consolidation of only wholly owned subsidiaries. The three boxes in the center section of Figure 7-2 summarize the three contexts. Figure 7-4 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

**Figure 7-4: Context Comparison in the Motivational Example**

Similar to the modifier values 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 “total revenue” of a company using SEC’s consolidation rules, and the tree that represents the rules in this context is shown in Figure 5-2. Here, I have simplified this example so that the value of the modifier “aggregationType” captures the rules in different contexts.

### 7.1.3 Modifiers

Figure 7-4 summarizes the values that the modifiers take in the three contexts. For example, the modifier *scale* is 1,000 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 COINL 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 last clause from above 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.

### 7.1.4 Conversion Functions

Conversion functions define how modifier values change between different contexts. In most cases, they can be 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 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.

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) looks 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,DivSubVal))))),

```

```

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,ABLRSV),
      flatten(ABLRSV,FBLRSV),
      flatten(PSBLV,FPSBLV),
      listproduct(FBLRSV,FPSBLV,BrSubValL),
      sumall(BrSubValL,BrSubValI),
      divide(BrSubValI,10000,BrSubVal))))),
plus(VLSA,DivSubVal,I),
plus(I,BrSubVal,II),
plus(II,Vs,Vt)).

```



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`.

### 7.1.5 Relations

Now let us continue exploring the sample problem. “Relations” are the data sources necessary for answering the queries. I have already presented the three tables that are used in this example in Figure 7-1, one of which is also shown in Figure 5-1 – the “revenue1” table:

CorporateEntity	Revenue
International Business Machines	77,966,000
Lotus Development	970,000
International Information Products	1,200,000
IBM Far East Holdings B. V.	550,000
IBM International Treasury Services	500,000
IBM Global Services	36,360,00
General Motors	177,828,100
Hughes Electronics	8,934,900
Electronic Data Systems	21,502,000

**Figure 7-5: Relation “revenue1” – the Atomic Revenues of Some Selected Corporate Entities** (in the `c_unconsolidated_revenue` context)

In reality, the above table could be the result of a join on multiple data sources. There are some assumptions made to simplify the problem: 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.

Besides the revenue data, we also need information about IBM’s corporate group structure in order to apply the rules. A desired relation (“structure”<sup>6</sup>) is shown in the following table:

ChildEntity	ParentEntity	RelationshipType	Ownership
IBM Credit Corp.	International Business Machines	Subsidiary	100
Lotus Development	International Business Machines	Subsidiary	100
IBM World Trade Corp.	International Business Machines	Subsidiary	100
IBM de Colombia, S.A.	IBM World Trade Corp.	Subsidiary	90 <sup>(a)</sup>
IBM Plans Management Corp.	IBM World Trade Corp.	Subsidiary	88 <sup>(a)</sup>
Compagnie IBM France, S.A.	International Business Machines	Subsidiary	100 <sup>(b)</sup>
Tunisian Business Machines	Compagnie IBM France, S.A.	Subsidiary	83 <sup>(b)</sup>
IBM East Africa Limited	International Business Machines	Subsidiary	67 <sup>(a)</sup>
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 Enterprise Investment	International Business Machines	Division	100
IBM Software	International Business Machines	Division	100
IBM Hardware	International Business Machines	Division	100
IBM Global Financing	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
IBM International Treasury Services	IBM Switzerland	Subsidiary	25
Hughes Electronics	General Motors	Subsidiary	100
...	...	...	...

<sup>(a)</sup> Remaining percentage owned by other wholly-owned IBM company(s).

<sup>(b)</sup> Minor percentage held by other IBM shareholders, subject to repurchase option.

**Figure 7-6: Relation “structure” – Pairs of Related Corporate Entities and Details about Their Relationships**

The columns in the above table are self-explanatory. For example, IBM World Trade Corporation is a wholly owned subsidiary of International Business Machines Corporation; IBM de Colombia, S. A. is owned 90% by IBM World Trade Corporation, and the rest 10% is owned by other wholly-owned IBM subsidiaries. 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).

Note, for demonstration purposes, the structure table is presented as an inline database table, that is, a set of facts (“rule” statements) in the abduction code. Due to the limited

<sup>6</sup> The data in this table is extracted from Exhibit 20.01 in IBM’s annual report for the year ending Dec.31<sup>st</sup>, 2002. The complete list of IBM’s majority-owned subsidiaries is in Appendix A.

time frame of this thesis, database access routines were not developed. Nevertheless, only a minor change in the code is needed if database calls are implemented and added in the future. Some sample statements that define the data in the structure table follow:

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

```
rule(structure("IBM Far East Holdings B. V.", "International Business Machines",
"Subsidiary", 100), (true)).
```

```
rule(structure("International Information Products", "IBM Far East Holdings B. V.",
"Subsidiary", 80), (true)).
```

Another necessary piece of information is the country of incorporation or location table “country”:

CorporateEntity	Country
International Business Machines	USA
IBM Credit Corp.	USA
Lotus Development	USA
IBM World Trade Corp.	USA
IBM Plans Management Corp.	USA
IBM Americas Holding Limited	Bermuda
IBM Brasil Industria, Maquinas e Servicos Ltda	Brazil
IBM de Colombia, S.A	Columbia
Compagnie IBM France, S.A.	France
Tunisian Business Machines	Tunisia
IBM East Africa Limited	Kenya
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
Electronics Data Systems	USA
...	...

**Figure 7-7: Relation “country” – Country of Incorporated or Locations**

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 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 Figure 5-2, 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, I have made some reasonable assumptions – IBM has controlling financial interest on its subsidiaries, it is not a bank holding company, and all entities share the same fiscal period end date. So the types of relationships and ownership percentages are what determine the aggregation between entities.

### 7.1.6 Elevations

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.

The upward arrows in Figure 7-1 indicate how each column in the relations are elevated through semantic objects to semantic types in the ontology. Recall that the “revenue1” relation has the following columns:

Name	Type
CorporateEntity	VARCHAR(60)
Revenue	INTEGER

**Figure 7-8: Schema of the “revenue1” table**

The elevated relation corresponding to “revenue1” for the context `c_unconsolidated_revenue` looks as follows (in COINL).

```
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, other elevation axioms are defined in COIL as follows.

```
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)),
```

---

<sup>7</sup> 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.

```

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))).

```

A summary of the elevations follow:

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

**Figure 7-9: Summary of Elevations from Relations to the Domain Model**

### 7.1.7 Query Mediation and Execution

In this section, I will go through each step of the query mediation process 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 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".

```

Referring back to Figure 7-2, the current case corresponds to the path from the input query through the middle context box (`c_majorityowned_revenue`) to the desired output query, which looks as follows:

```

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, the only difference is: since ownership percentages are used to discount the revenue numbers (as explained in Section 7.1.4), the final result should be 80,946,000, instead of 81,186,000 as shown in Figure 7-2. (As proven later, this is indeed the result returned by the demo application.)

As the first step after a SQL query is fed in, the COIN system generates a native datalog query 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 Section 7.1. This query ascertains that the result returned to the user has been 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 I will not go into details in this thesis. The engine takes the datalog query and the domain model axioms (such as the conversion function presented in Section 7.1.4), and computes a set of abducted queries that have considered all the possible cases of conflicts. The result of context detection is shown in Figure 7-10. (The complete table as a result of context detection is included in Appendix F.) Modifier values in the source and receiver contexts, as well as the conversion functions between these two contexts are discovered.

Demo for Corporate Householding - Microsoft Internet Explorer

Address: http://avocado.mit.edu:8080/gcms/Demo.jsp

### GCMS Demo for Corporate Householding

Management: [ [ontology](#) | [contexts](#) | [conversions](#) | [sources](#) | [elevations](#) | [integrity constraints](#) | [qbe](#) ]  
 [ [Other Demos](#) | [Internal Representation](#) | [Constraints](#) ]

[ [Enable Log](#) ]

**Queries**

Query	Description
New01*	Query 1 description

**SQL**

```
Select CorporateEntity, Revenue from revenue1
where CorporateEntity = "International Business Machines";
```

**Context**: c\_majorityowned\_revenue

**Stage**

Naive Datalog       SQL Translation  
 Context Sensitive Datalog       Execution  
 Conflict Detection  
 Mediation

Submit    Reset

**Result**

SemanticType	Column	Source	Modifier	Modifier value in source context	Modifier value in target context	Conversion Function
Revenue	C2	revenue1 (International Business Machines, C2)	aggregationType	c_unconsolidated_revenue : division+branch	c_majorityowned_revenue : subsidiary+division+branch	attr(V54, company, V53), attr(V53, subsidiary, V52), flatten(V52, V51), length(V51, V50), if (V50 < 1, V49 is 0, (attr(V53, percentsubsidiary, V48), maplist(attr2 (company), V52, V47), maplist (value_reverse2 (V46), V47, V45), maplist (value_reverse2 (V46), V48, V44), listproduct(V45, V44, V43), flatten (V43, V42), sumall (V42, V41), divide (V41, 100, V49))), attr(V53, division, V40), flatten(V40, V39), length(V39, V38), if(V38 < 1, V37 is 0, (attr(V53, percentdivision, V36), maplist (value_reverse2 (V46), V36, V35), maplist (attr_reverse2 (subsidiary), V40, V34), flatten(V34, V33), length(V33,

Figure 7-10: Demo (1) – Results of Context Detection

The mediated query produced by the Context Mediator is shown below, and also in Figure 7-11:

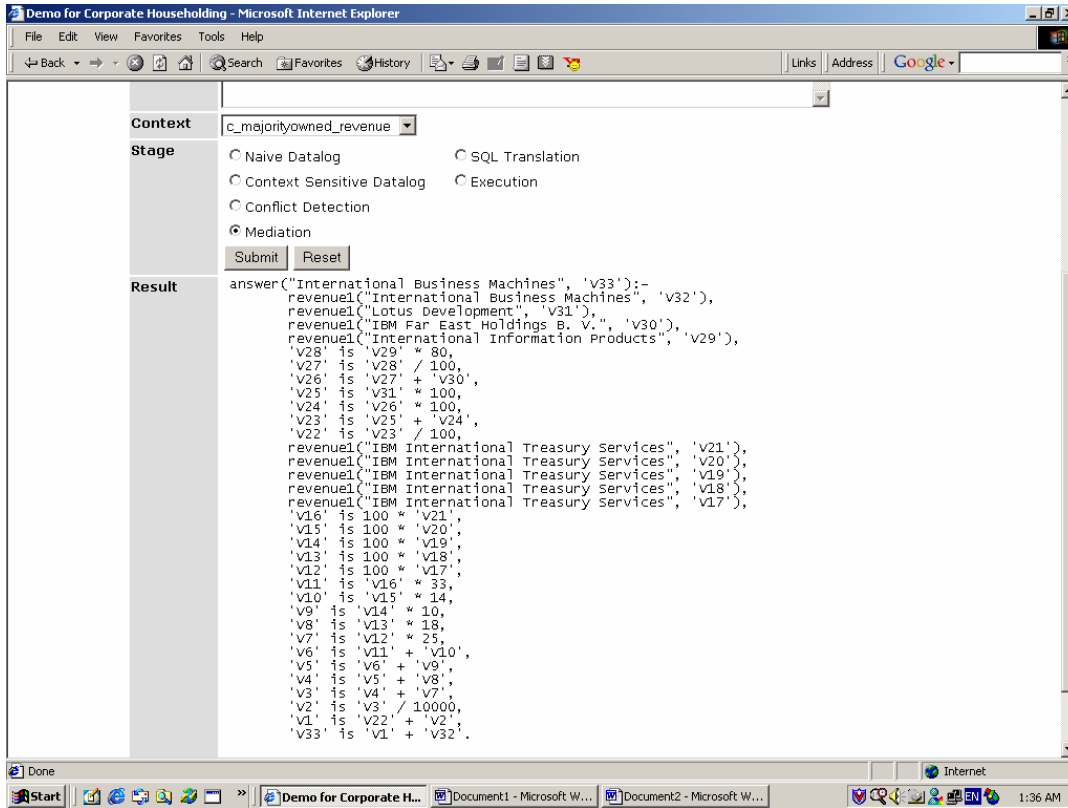
```
answer("International Business Machines", 'V25'):-
  revenue1("International Business Machines", 'V24'),
  revenue1("Lotus Development", 'V23'),
  revenue1("IBM Far East Holdings B. V.", 'V22'),
  revenue1("International Information Products", 'V21'),
  'V20' is 'V21' * 80, 'V19' is 'V20' / 100, '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, '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.





**Figure 7-11: Demo (2) – The Mediated Datalog Query**

After the mediated query is generated, it is being translated back to a SQL statement (shown in Figure 7-12) through a Query Planner and Optimizer [17]. Now 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 in Section 7.1.4, 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 7-13.

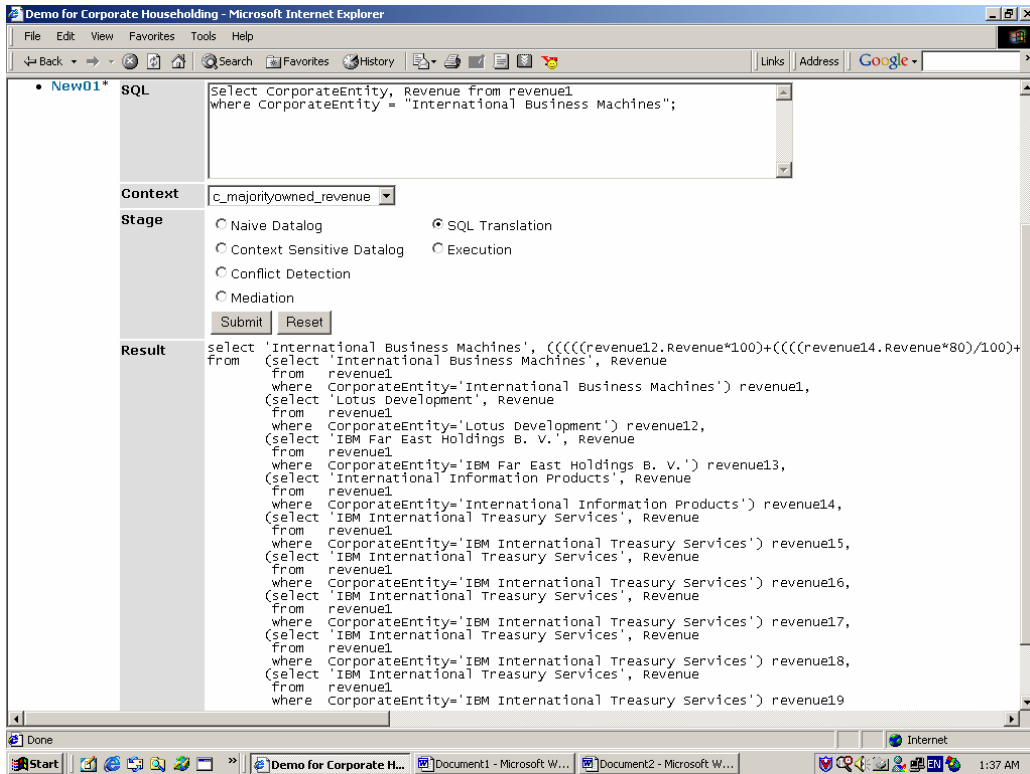


Figure 7-12: Demo (3) – Result of SQL Translation

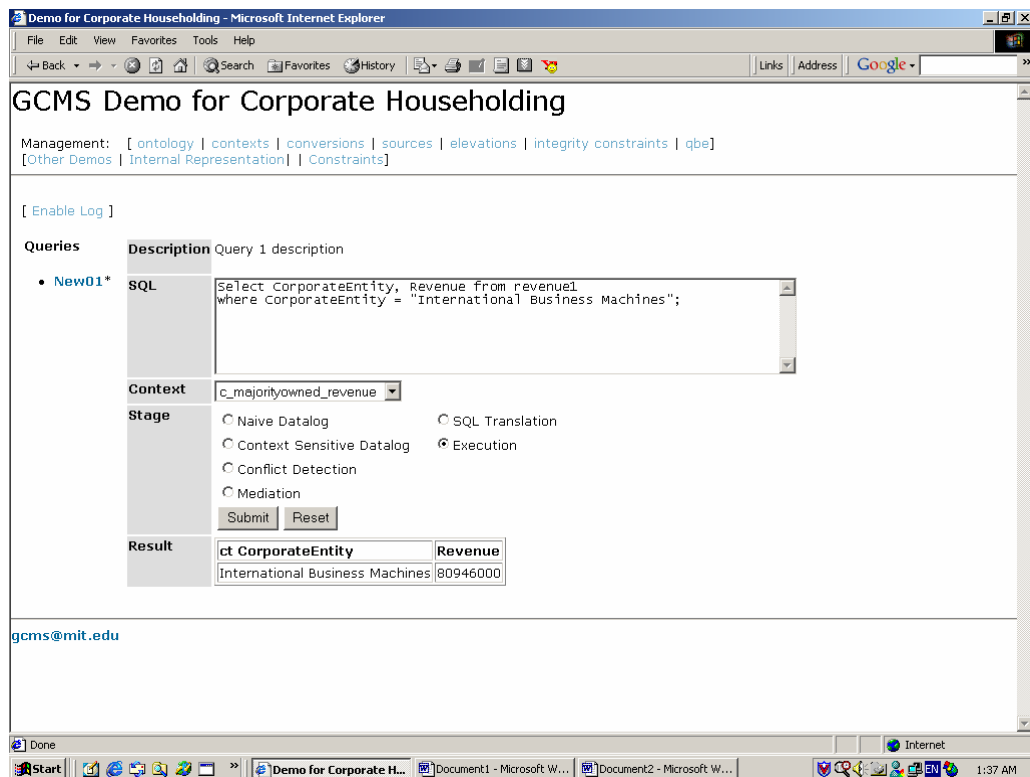


Figure 7-13: Demo (4) – Result of Execution

## **7.2 Comparison of Mediation-time and Execution-time Approaches**

The demo application presented in the above section performs the reasoning based on corporate structures during query mediation, and produces a mediated query that can be directly applied to the source data. Let us refer to this approach as the mediation-time approach. On the contrary, while the reasoning process using the business rules extracted from application domains is done during mediation, the actual corporate householding process can be done at execution time – this approach can be referred to as the execution-time approach.

Technically, both the mediation-time and the execution-time approaches should be able to achieve the same set of functionalities. Nevertheless, they both have advantages and weaknesses. The mediation-time approach encapsulates everything inside the corporate householding query engine with very complex conversion functions and abduction logic. Compared to the execution-time approach, it might be faster due to potential code optimization, or specialization in solving certain types of problems, such as account consolidation. But this design is low in modularity and reusability. Therefore, in terms of flexibilities and cleanness, the execution-time approach is better. If the actual corporate householding process specific to an application domain is performed during execution and outside the query mediation engine, extensions to other application domains are easier to implement, whereas in the case of the mediation-time approach, generalizations require a considerable amount of time and effort.

## 8 Conclusions and Future Work

In this thesis, I briefly stated the importance and challenges of corporate householding, described categories of corporate householding problems, and illustrated a few application areas with examples derived from corporate householding knowledge research. Then I presented a motivational example in account consolidation. Although simplified, this sample scenario can be extended to be more realistic by adding real data sources and more specific rules. Following that, I described the COntext INterchange technology that performs mediated data access among heterogeneous data sources, and briefly covered the extended version of it. Based on the ECOIN model, I proposed 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.

There are two major directions for future research:

- Continue exploring corporate householding knowledge – extract more detailed, specific, and useful business rules that can be used in the knowledge processor; and
- Continue improving the corporate householding knowledge processor – improve the current domain model (e.g. replace the modifier “aggregationType” with more specific modifiers, each representing if one type of corporate entities should be aggregated), extend the query mediation engine to deal with more areas of application, or try to implement the execution-time approach.

It would also be interesting to see how entity aggregation in corporate householding relates to other areas where entity aggregation may apply, such as deciding which part/component(s) should be grouped in a particular system in the field of manufacturing. In addition, future refinements on the Extended COntext INterchange system can potentially provide more functionalities to corporate householding query processor.

## 9 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 6<sup>th</sup> 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 Household*, in the proceedings of the 7<sup>th</sup> 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*. 6<sup>th</sup> edition. West group, 2002: p.126-127.
14. Lee, P., *COIN Textual Interface Manual*, 2003.
15. Goh, C. H., Madnick, S., Siegel, M., *Context Interchange: Overcoming the Challenges of Large-scale Interoperable Database Systems in a Dynamic Environment*, in proceedings of the 3<sup>rd</sup> International Conference on Information and Knowledge Management, November 1994.
16. Manuel, A., *Credit Risk Valuation: Methods, Models and Application*. 2<sup>nd</sup> edition. Springer-Verlag, 2001: p.1.

17. Alatovic, T., *Capabilities Aware Planner/Optimizer/Executioner for Context INterchange Project*, MIT Masters Thesis, Electrical Engineering and Computer Science, 2002.

## Appendix A: IBM – List of Majority-owned Subsidiaries

Source: <http://www.sec.gov/Archives/edgar/data/51143/000104746903008194/a2102367z-ex-21.htm>

	State or country of incorporation or organization	Percentage of voting securities owned by its immediate parent	
Registrant:			
International Business Machines Corporation	New York		
Subsidiaries:			
IBM Credit Corporation	Delaware	100	
Lotus Development Corporation	Delaware	100	
Tivoli Systems Inc.	Delaware	100	
IBM World Trade Corporation	Delaware	100	
IBM Central and Eastern Europe/ Middle East/Africa, Inc	Delaware	100	
IBM Plans Management Corporation	New York	88	(A)
IBM World Trade Asia Corporation	Delaware	100	
IBM World Trade Holding LLC	Delaware	100	
WTC Insurance Corporation, Ltd.	Bermuda	100	
IBM Argentina, S.A.	Argentina	100	(A)
IBM Australia Ltd.	Australia	100	
IBM Bahamas Ltd.	Bahamas	100	
IBM de Bolivia, S.A.	Bolivia	100	
IBM Americas Holding Limited	Bermuda	100	
IBM Brasil Industria, Maquinas e Servicos Ltda	Brazil	100	(A)
Donelly Financial Inc.	British V.I.	100	
IBM Foreign Sales Corporation	Barbados	100	

Fitzal Capital Inc.	British V.I.	100	
IBM Canada Credit Services Company	Canada	100	
IBM Canada Limited—IBM Canada Limitee	Canada	100	
IBM China Company Limited	China	100	
IBM China/Hong Kong Limited	Hong Kong	100	
IBM de Chile, S.A.C.	Chile	100	(A)
IBM de Colombia, S.A.	Colombia	90	(A)
IBM Middle East FZE	United Arab Emirates	100	
IBM Middle East FZ-LLC	United Arab Emirates	100	

IBM del Ecuador, S.A.	Ecuador	100	
IBM Global Services India Pvt. Ltd.	India	90	(A/B)
IBM India Ltd.	India	100	
YK IBM AP Holdings	Japan	100	
IBM Japan, Ltd.	Japan	100	
IBM Asia Pacific Service Corporation	Japan	100	
IBM Business Consulting Services KK	Japan	100	
Display Technologies, Inc.	Japan	100	
IBM Korea, Inc.	Korea (South)	100	
PT IBM Indonesia	Indonesia	99	
IBM Malaysia Sdn. Bhd	Malaysia	100	
Financiera de Tecnologia e Informatica S.A. de C.V., Sociedad Financiera del Objeto Limitado Filial	Mexico	99	(A)
Grupo IBM Mexico, S.A. de C.V.	Mexico	100	(B)
IBM de Mexico, S.A.	Mexico	100	(B)
IBM New Zealand Ltd.	New Zealand	100	
IBM del Peru, S.A.	Peru	100	



IBM Philippines, Incorporated	Philippines	100	(B)
IBM Romania Srl	Romania	100	
IBM Taiwan Corporation	Taiwan	100	(A)
IBM Thailand Company Ltd.	Thailand	100	(B)
IBM del Uruguay, S.A.	Uruguay	100	
IBM de Venezuela, S.A.	Venezuela	100	
IBM Vietnam Company	Vietnam	100	
International Business Machines of Belgium S.A.	Belgium	100	(A)
IBM Bulgaria Ltd.	Bulgaria	100	
IBM Croatia Ltd./ IBM Hrvatska d.o.o.	Croatia	100	
IBM Global Services Delivery Center Czech Republic s.r.o.	Czech Republic	100	
IBM Business Consulting Services s.r.o.	Czech Republic	100	
IBM Eesti Osauhing (IBM Estonia Ou)	Estonia	100	
Compagnie IBM France, S.A.	France	100	(B)
IBM Eurocoordination	France	96	(A)
Tunisian Business Machines (TBM)	Tunisia	83	(B)
IBM Europe Middle East Africa	France	100	(A)
IBM International Treasury Services Company	Ireland	—	(C)
IBM Italia S.p.A.	Italy	100	
IBM Hellas Information Handling Systems S.A.	Greece	100	(A)
IBM Israel Ltd.	Israel	100	(A)
Companhia IBM Portuguesa, S.A.	Portugal	100	
IBM (International Business Machines) Turk Ltd. Sirketi	Turkey	98	(A)
IBM South Africa (Pty) Ltd.	South Africa	100	
IBM East Africa Limited	Kenya	67	(A)

	Sabiedriba ar irobezotu atbildibu IBM Latvija	Latvia	100	
	IBM Lietuva	Lithuania	100	
	IBM Business Consulting Services Limited	Lebanon	98	(A)
	IBM BTO Global Holdings B.V.	Netherlands	100	
	IBM Business Consulting Services Eastern Europe B.V.	Netherlands	100	
	IBM Far East Holdings B.V.	Netherlands	100	
	International Information Products (Shenzhen) Co., Ltd.	China	80	
	IBM Holdings B.V.	Netherlands	100	
	IBM Global Holdings B.V.	Netherlands	65	(A)
	IBM Central Holdings GmbH	Germany	100	
	IBM Ceskra Republika spol.s.r.o.	Czech Republic	100	
	IBM Deutschland GmbH	Germany	100	
	IBM Oesterreich Internationale Bueromaschinen Gesellschaft m.b.H.	Austria	100	
	IBM (Schweiz)—IBM (Suisse)—IBM (Swizzera)—IBM (Switzerland)	Suitzerland	100	
	IBM East Europe/Asia Ltd.	Russia	100	
	IBM Polska Sp. z.o.o.	Poland	100	
	International Business Machines Corporation Magysrorssyagi Kft.	Hungary	100	(A)
	IBM Eastern Europe Holdings B.V.	Netherlands	100	(A)
	IBM Global Services Kft.	Hungary	100	
	IBM International Holdings B.V.	Netherlands	100	
	IBM Ireland Limited	Ireland	100	
	IBM Singapore Pte. Ltd.	Singapore	100	
	IBM Storage Products Kft.	Hungary	100	
	IBM North Region Holdings	United Kingdom	100	(B)

	IBM Nederland N.V.	Netherlands	100	
	IBM United Kingdom Holdings Limited	United Kingdom	100	
	IBM United Kingdom Limited	United Kingdom	100	
	IBM International Centre for Asset Management N.V.	Netherlands	100	
	International Business Machines A/S	Norway	60	(A)
	IBM West Africa Limited	Nigeria	100	
	IBM Slovensko spol.s.r.o.	Slovak Republic	100	
	IBM Slovenija d.o.o.	Slovenia	100	
	International Business Machines, S.A.	Spain	100	(A)
	IBM Nordic Aktiebolag	Sweden	100	
	IBM Danmark A/S	Denmark	100	
	Oy International Business Machines AB	Finland	100	
	IBM Svenska Aktiebolag	Sweden	100	
	IBM Business Consulting Services Limited	United Kingdom	98	(B)
	IBM (International Business Machines) Is Danismanlik Hizmetleri Limited	Turkey	100	

(A) Remaining percentage owned by other wholly-owned IBM company(s).

(B) Minor percentage held by other IBM shareholders, subject to repurchase option.

(C) IBM Germany owns 33.7%, IBM France owns 13.5%, IBM Finland owns 10.1%, IBM Denmark owns 18.0% and IBM Switzerland owns 24.7% of IBM International Treasury Services Company.

## Appendix B: IBM – Total Revenue

Source: <http://www.sec.gov/Archives/edgar/data/51143/000104746903008194/a2102367z-ex-13.htm>

Units: millions USD

Fiscal year ending: December 31

	2002	2001	2000
Global Services	\$ 36,360	\$ 34,956	\$ 33,152
Hardware	27,456	30,593	34,470
Software	13,074	12,939	12,598
Global Financing	3,232	3,426	3,465
Enterprise Investments/Other	1,064	1,153	1,404
Total Revenue	81,186	83,067	85,089

## Appendix C: Lotus Development – Net Sales

Source: <http://www.sec.gov/Archives/edgar/data/711761/0000711761-95-000007.txt>

Unit: thousands USD

1994	1993	1992	1991	1990
\$970,723	\$981,168	\$900,149	\$828,895	\$692,242

## Appendix D: General Motors, Hughes Electronics and EDS – Relationships & Revenues

### (I) Revenue of General Motors

Source:

<http://www.sec.gov/Archives/edgar/data/40730/000004073003000054/complete10k.txt>

Units: millions USD

Fiscal year ending: December 31

2002	2001	2000	1999
\$186,763	\$177,260	\$184,632	\$176,558

### (II) A list of majority-owned subsidiaries of GM:

<http://www.sec.gov/Archives/edgar/data/40730/000004073003000054/exhibit21.txt>

(III) Hughes Electronics Corporation is a wholly-owned subsidiary of General Motors Corporation. Hughes is a Delaware corporation incorporated in 1977 and was acquired by GM in 1985.

### Revenue of Hughes Electronics

Source:

<http://www.sec.gov/Archives/edgar/data/944868/000095016803000639/d10k.htm>

Units: millions USD

Fiscal year ending: December 31

2002	2001	2000
\$8,934.9	\$8,264.0	\$7,287.6

(IV) Electronic Data Systems became an independent company on June 7, 1996, after 11 years of GM ownership.

### Revenue of Electronic Data Systems Corp

Source:

<http://www.sec.gov/Archives/edgar/data/1007456/000093066103001011/d10k.htm>

Units: millions USD

Fiscal year ending: December 31

(Base: revenue from non-GM clients; GM: revenue from contracts with GM)

	2002	2001	2000
Base	\$18,907	\$18,068	\$15,486
GM	\$2,595	\$3,073	\$3,370
Total	\$21,502	\$21,141	\$18,856

## Appendix E: Ontology, Contexts, Relations, Elevations and Conversion Functions of Demo Application

%% generation timestamp: 4/22/2003 2:40:48 PM

```
:- module_interface(application511).  
:- export rule/2.  
:- begin_module(application511).  
:- dynamic rule/2.
```

%%

%% Semantic Types

%%

```
rule(is_a('CorporateEntity', basic), (true)).  
rule(is_a('Country', basic), (true)).  
rule(is_a('CurrencyType', basic), (true)).  
rule(is_a('Date', basic), (true)).  
rule(is_a('EntityFinancials', basic), (true)).  
rule(is_a('Relationship', basic), (true)).  
rule(is_a('Revenue', 'EntityFinancials'), (true)).
```

%%

%% Modifiers

%%

```
rule(modifiers(basic, []), (true)).  
rule(modifiers('CorporateEntity', []), (true)).  
rule(modifiers('Country', []), (true)).  
rule(modifiers('CurrencyType', []), (true)).  
rule(modifiers('Date', []), (true)).  
rule(modifiers('EntityFinancials', [aggregationType]), (true)).  
rule(modifiers('Relationship', []), (true)).  
rule(modifiers('Revenue', []), (true)).
```

%%

%% Attributes

%%

```
rule(attributes(basic, []), (true)).  
rule(attributes('CorporateEntity', [subsidiary,division,branch,percentsubsidiary,  
percentbranch, percentdivision]), (true)).  
rule(attributes('Country', [location, officialCurrency]), (true)).  
rule(attributes('CurrencyType', []), (true)).  
rule(attributes('Date', []), (true)).  
rule(attributes('EntityFinancials', [company, fyEnding]), (true)).
```

```
rule(attributes('Relationship', [childEntity, ownership, parentEntity, relationshipType]),
(true)).
rule(attributes('Revenue', []), (true)).
```

```
%%
%% Contexts
%%
```

```
rule(contexts([c_whollyowned_revenue,c_majorityowned_revenue,c_unconsolidated_rev
enue]),(true)).
rule(context(c_whollyowned_revenue), (true)).
rule(context(c_majorityowned_revenue), (true)).
rule(context(c_unconsolidated_revenue), (true)).
```

```
%%
%% “c_whollyowned_revenue” Context
%%
```

```
rule(modifier('EntityFinancials', Object, aggregationType, c_whollyowned_revenue,
Modifier),
(cste(basic, Modifier, c_whollyowned_revenue,
"whollyownedsubsidiary+division+branch"))).
```

```
rule(modifier('EntityFinancials', Object, currency, c_whollyowned_revenue, Modifier),
(cste(CurrencyType, Modifier, c_whollyowned_revenue, "USD"))).
```

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

```
%%
%% “c_majorityowned_revenue” Context
%%
```

```
rule(modifier('EntityFinancials', Object, aggregationType, c_majorityowned_revenue,
Modifier),
(cste(basic, Modifier, c_majorityowned_revenue, "subsidiary+division+branch"))).
```

```
rule(modifier('EntityFinancials', Object, currency, c_majorityowned_revenue, Modifier),
(cste(CurrencyType, Modifier, c_majorityowned_revenue, "USD"))).
```

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

```
%%
%% “c_unconsolidated_revenue” Context
%%
```



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

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

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

```
%%
```

```
%% Conversion Functions for CorporateEntity with respect to aggregationType
```

```
%%
```

```
rule(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), %[a,b,c]
flatten(DL,TFDL),
length(TFDL,DLength),
if(DLength < 1, DivSubVal is 0,
(attr(C,percentdivision,PDL), %[10,20,30]
maplist(value_reverse2(Ctxt),PDL,PDLV), %[10,20,30]
maplist(attr_reverse2(subsidiary),DL,DLS),%[[a1,a2],[b1], []]
flatten(DLS,TFDLS),
length(TFDLS,DLSLength),
if(DLSLength < 1, DivSubVal is 0,
(maplist(attr_reverse2(percentsubsidiary),DL,PSDL),%[[100,90],[80],[[]]
maplist(maplist(value_reverse2(Ctxt)),PSDL,PSDLV), %[100,90],[80],[[]]
maplist(maplist(attr2(company)),DLS,DLSR), %[a1,a2],[b1], []]
maplist(maplist(value_reverse2(Ctxt)),DLSR,DLSRV), %[a1,a2],[b1], []]
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), %[a,b,c]
flatten(BL,TFBL),
length(TFBL,BLength),
if(BLength < 1, BrSubVal is 0,
(attr(C,percentbranch,PBL), %[10,20,30]
maplist(value_reverse2(Ctxt),PBL,PBLV), %[10,20,30]
maplist(attr_reverse2(subsidiary),BL,BLS),%[[a1,a2],[b1], []]
flatten(BLS,TFBLS),
length(TFBLS,BLSLength),
if(BLSLength < 1, BrSubVal is 0,
(maplist(attr_reverse2(percentsubsidiary),BL,PSBL),%[[100,90],[80],[]]
maplist(maplist(value_reverse2(Ctxt)),PSBL,PSBLV), %[100,90],[80],[]]
maplist(maplist(attr2(company)),BLS,BLSR), %[a1,a2],[b1], []]
maplist(maplist(value_reverse2(Ctxt)),BLSR,BLSRV), %[a1,a2],[b1], []]
listproduct(BLSRV,PBLV,ABLRSRV),
flatten(ABLRSRV,FBLRSRV),
flatten(PSBLV,FPSBLV),
listproduct(FBLRSRV,FPSBLV,BrSubValL),
sumall(BrSubValL,BrSubValI),
divide(BrSubValI,10000,BrSubValI))))),
plus(VLSA,DivSubValI),
plus(I,BrSubVal,II),
plus(II,Vs,Vt)).

%%
%% Elevations
%%

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

rule(
  revenue1_p(
    skolem('CorporateEntity', C1, c_unconsolidated_revenue, 1,
      revenue1(C1, C2)),

```

```
    skolem('Revenue', C2, c_unconsolidated_revenue, 2,
          revenue1(C1, C2)),
    (revenue1(C1, C2))).
```

```
rule(
  revenue2_p(
    skolem('CorporateEntity', C1, c_majorityowned_revenue, 1,
          revenue2(C1, C2)),
    skolem('Revenue', C2, c_majorityowned_revenue, 2,
          revenue2(C1, C2))),
  (revenue2(C1, C2))).
```

```
rule(
  revenue3_p(
    skolem('CorporateEntity', C1, c_whollyowned_revenue, 1,
          revenue3(C1, C2)),
    skolem('Revenue', C2, c_whollyowned_revenue, 2,
          revenue3(C1, C2))),
  (revenue3(C1, C2))).
```

```
rule(
  structure_p(
    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))),
  (structure(C1, C2, C3, C4))).
```

```
%%  
%% Relations  
%%
```

```
rule(relation(oracle,  
  country,  
  ie,  
  [['CorporateEntity', string],  
  ['Country', string]],  
  cap([[0, 0]],  
  [])), (true)).
```

```
rule(relation(oracle,  
  revenue1,
```

```

ie,
[['CorporateEntity', string],
 ['Revenue', integer]],
cap([[0, 0]],
 [])), (true)).

rule(relation(oracle,
revenue2,
ie,
[['CorporateEntity', string],
 ['Revenue', integer]],
cap([[0, 0]],
 [])), (true)).

rule(relation(oracle,
revenue3,
ie,
[['CorporateEntity', string],
 ['Revenue', integer]],
cap([[0, 0]],
 [])), (true)).

rule(relation(oracle,
structure,
e,
[['ChildEntity', string],
 ['ParentEntity', string],
 ['Relationship', string],
 ['Ownership', integer]],
cap([[0, 0, 0, 0]],
 [])), (true)).

%%
%% Helper Functions
%%

rule(attr(X, subsidiary, Y),
(value(X,c_unconsolidated_revenue,XV),
 findall(YV,db(structure,[YV,XV,"Subsidiary",PV]),YVL),
 maplist(value2(c_unconsolidated_revenue),YVL,Y)
)).

rule(attr(X, percentsubsidiary, Y),
(value(X,c_unconsolidated_revenue,XV),
 findall(PV,db(structure,[YV,XV,"Subsidiary",PV]),PVL),
 maplist(value2(c_unconsolidated_revenue),PVL,Y)
)).

```

```

)).
rule(attr(X, division, Y),
(value(X,c_unconsolidated_revenue,XV),
 findall(YV,db(structure,[YV,XV,"Division",PV]),YVL),
 maplist(value2(c_unconsolidated_revenue),YVL,Y)
)).
rule(attr(X, percentdivision, Y),
(value(X,c_unconsolidated_revenue,XV),
 findall(PV,db(structure,[YV,XV,"Division",PV]),PVL),
 maplist(value2(c_unconsolidated_revenue),PVL,Y)
)).
rule(attr(X, branch, Y),
(value(X,c_unconsolidated_revenue,XV),
 findall(YV,db(structure,[YV,XV,"Branch",PV]),YVL),
 maplist(value2(c_unconsolidated_revenue),YVL,Y)
)).
rule(attr(X, percentbranch, Y),
(value(X,c_unconsolidated_revenue,XV),
 findall(PV,db(structure,[YV,XV,"Branch",PV]),PVL),
 maplist(value2(c_unconsolidated_revenue),PVL,Y)
)).

rule(attr(X, company, Y), (revenue1_p(Y,X))).
rule(attr(X, company, Y), (revenue1_p(Y1,X),value(Y,c_unconsolidated_revenue,YV),
value(Y1,c_unconsolidated_revenue,YV))).
rule(attr(X, company, Y), (revenue2_p(Y,X))).
rule(attr(X, company, Y), (revenue2_p(Y1,X),value(Y,c_majorityowned_revenue,YV),
value(Y1,c_majorityowned_revenue,YV))).
rule(attr(X, company, Y), (revenue3_p(Y,X))).
rule(attr(X, company, Y), (revenue3_p(Y1,X),value(Y,c_whollyowned_revenue,YV),
value(Y1,c_whollyowned_revenue,YV))).

%%
%% Data from the "structure" Table
%%

rule(structure("Lotus Development", "International Business Machines", "Subsidiary",
100), (true)).
rule(structure("IBM Far East Holdings B. V.", "International Business Machines",
"Subsidiary", 100), (true)).
rule(structure("International Information Products", "IBM Far East Holdings B. V.",
"Subsidiary", 80), (true)).
rule(structure("IBM Global Services", "International Business Machines", "Division",
100), (true)).
rule(structure("IBM Enterprise Investment", "International Business Machines",
"Division", 100), (true)).

```

rule(structure("IBM Software", "International Business Machines", "Division", 100), (true)).  
rule(structure("IBM Hardware", "International Business Machines", "Division", 100), (true)).  
rule(structure("IBM Global Financing", "International Business Machines", "Division", 100), (true)).  
rule(structure("IBM Germany", "International Business Machines", "Branch", 100), (true)).  
rule(structure("IBM France", "International Business Machines", "Branch", 100), (true)).  
rule(structure("IBM Finland", "International Business Machines", "Branch", 100), (true)).  
rule(structure("IBM Denmark", "International Business Machines", "Branch", 100), (true)).  
rule(structure("IBM Switzerland", "International Business Machines", "Branch", 100), (true)).  
rule(structure("IBM International Treasury Services", "IBM Germany", "Subsidiary", 33), (true)).  
rule(structure("IBM International Treasury Services", "IBM France", "Subsidiary", 14), (true)).  
rule(structure("IBM International Treasury Services", "IBM Finland", "Subsidiary", 10), (true)).  
rule(structure("IBM International Treasury Services", "IBM Denmark", "Subsidiary", 18), (true)).  
rule(structure("IBM International Treasury Services", "IBM Switzerland", "Subsidiary", 25), (true)).

## Appendix F: Conflict Detection Table (complete version) Returned by the Live Demo

Semantic Type	Column	Source	Modifier	Modifier value in source context	Modifier value in target context	Conversion Function
Revenue	C2	revenue1(International Business Machines, C2)	aggregationType	c_unconsolidated_revenue : division+branch	c_majorityowned_revenue : subsidiary+division+branch	attr(V54, company, V53), attr(V53, subsidiary, V52), flatten(V52, V51), length(V51, V50), if(V50 < 1, V49 is 0, (attr(V53, percentsubsidiary, V48), maplist(attr2(company), V52, V47), maplist(value_reverse2(V46), V47, V45), maplist(value_reverse2(V46), V48, V44), listproduct(V45, V44, V43), flatten(V43, V42), sumall(V42, V41), divide(V41, 100, V49))), attr(V53, division, V40), flatten(V40, V39), length(V39, V38), if(V38 < 1, V37 is 0, (attr(V53, percentdivision, V36), maplist(value_reverse2(V46), V36, V35), maplist(attr_reverse2(subsidiary), V40, V34), flatten(V34, V33), length(V33, V32), if(V32 < 1, V37 is 0, (maplist(attr_reverse2(percentsubsidiary), V40, V31), maplist(maplist(...), V31, V30), maplist(...), ..., ...))), attr(V53, branch, V22), flatten(V22, V21), length(V21, V20), if(V20 < 1, V19 is 0, (attr(V53, percentbranch, V18), maplist(value_reverse2(V46), V18, V17), maplist(attr_reverse2(subsidiary), V22, V16), flatten(V16, V15), length(...), if(...))), plus(V49, V37, V4), plus(V4, V19, V3), plus(V3, V2, V1)
Revenue	C2	revenue1(Lotus Development, C2)	aggregationType	c_unconsolidated_revenue : division+branch	c_majorityowned_revenue : subsidiary+division+branch	attr(V54, company, V53), attr(V53, subsidiary, V52), flatten(V52, V51), length(V51, V50), if(V50 < 1, V49 is 0, (attr(V53, percentsubsidiary, V48), maplist(attr2(company), V52, V47), maplist(value_reverse2(V46), V47, V45), maplist(value_reverse2(V46), V48, V44), listproduct(V45, V44, V43), flatten(V43,

						<p>V42), sumall(V42, V41), divide(V41, 100, V49))), attr(V53, division, V40), flatten(V40, V39), length(V39, V38), if(V38 &lt; 1, V37 is 0, (attr(V53, percentdivision, V36), maplist(value_reverse2(V46), V36, V35), maplist(attr_reverse2(subsidiary), V40, V34), flatten(V34, V33), length(V33, V32), if(V32 &lt; 1, V37 is 0, (maplist(attr_reverse2(percentsubsidiary), V40, V31), maplist(maplist(...), V31, V30), maplist(...), ..., ...))))), attr(V53, branch, V22), flatten(V22, V21), length(V21, V20), if(V20 &lt; 1, V19 is 0, (attr(V53, percentbranch, V18), maplist(value_reverse2(V46), V18, V17), maplist(attr_reverse2(subsidiary), V22, V16), flatten(V16, V15), length(...), if(...))), plus(V49, V37, V4), plus(V4, V19, V3), plus(V3, V2, V1)</p>
Revenue	C2	revenue1(IBM Far East Holdings B. V., C2)	aggregationType	c_unconsolidated_revenue : division+branch	c_majorityowned_revenue : subsidiary+division+branch	<p>attr(V54, company, V53), attr(V53, subsidiary, V52), flatten(V52, V51), length(V51, V50), if(V50 &lt; 1, V49 is 0, (attr(V53, percentsubsidiary, V48), maplist(attr2(company), V52, V47), maplist(value_reverse2(V46), V47, V45), maplist(value_reverse2(V46), V48, V44), listproduct(V45, V44, V43), flatten(V43, V42), sumall(V42, V41), divide(V41, 100, V49))), attr(V53, division, V40), flatten(V40, V39), length(V39, V38), if(V38 &lt; 1, V37 is 0, (attr(V53, percentdivision, V36), maplist(value_reverse2(V46), V36, V35), maplist(attr_reverse2(subsidiary), V40, V34), flatten(V34, V33), length(V33, V32), if(V32 &lt; 1, V37 is 0, (maplist(attr_reverse2(percentsubsidiary), V40, V31), maplist(maplist(...), V31, V30), maplist(...), ..., ...))))), attr(V53, branch, V22), flatten(V22, V21), length(V21, V20), if(V20 &lt; 1, V19 is 0, (attr(V53, percentbranch, V18),</p>



						maplist(value_reverse2(V46), V18, V17), maplist(attr_reverse2(subsidiary), V22, V16), flatten(V16, V15), length(...), if(...)), plus(V49, V37, V4), plus(V4, V19, V3), plus(V3, V2, V1)
Revenue	C2	revenue1(International Information Products, C2)	aggregationType	c_unconsolidated_revenue : division+branch	c_majorityowned_revenue : subsidiary+division+branch	attr(V54, company, V53), attr(V53, subsidiary, V52), flatten(V52, V51), length(V51, V50), if(V50 < 1, V49 is 0, (attr(V53, percentsubsidiary, V48), maplist(attr2(company), V52, V47), maplist(value_reverse2(V46), V47, V45), maplist(value_reverse2(V46), V48, V44), listproduct(V45, V44, V43), flatten(V43, V42), sumall(V42, V41), divide(V41, 100, V49))), attr(V53, division, V40), flatten(V40, V39), length(V39, V38), if(V38 < 1, V37 is 0, (attr(V53, percentdivision, V36), maplist(value_reverse2(V46), V36, V35), maplist(attr_reverse2(subsidiary), V40, V34), flatten(V34, V33), length(V33, V32), if(V32 < 1, V37 is 0, (maplist(attr_reverse2(percentsubsidiary), V40, V31), maplist(maplist(...), V31, V30), maplist(...), ..., ...))), attr(V53, branch, V22), flatten(V22, V21), length(V21, V20), if(V20 < 1, V19 is 0, (attr(V53, percentbranch, V18), maplist(value_reverse2(V46), V18, V17), maplist(attr_reverse2(subsidiary), V22, V16), flatten(V16, V15), length(...), if(...))), plus(V49, V37, V4), plus(V4, V19, V3), plus(V3, V2, V1)
Revenue	C2	revenue1(IBM International Treasury Services, C2)	aggregationType	c_unconsolidated_revenue : division+branch	c_majorityowned_revenue : subsidiary+division+branch	attr(V54, company, V53), attr(V53, subsidiary, V52), flatten(V52, V51), length(V51, V50), if(V50 < 1, V49 is 0, (attr(V53, percentsubsidiary, V48), maplist(attr2(company), V52, V47), maplist(value_reverse2(V46), V47, V45), maplist(value_reverse2(V46), V48, V44), listproduct(V45, V44, V43), flatten(V43, V42), sumall(V42, V41), divide(V41, 100, V49))), attr(V53, division, V40),

						<pre> flatten(V40, V39), length(V39, V38), if(V38 &lt; 1, V37 is 0, (attr(V53, percentdivision, V36), maplist(value_reverse2(V46), V36, V35), maplist(attr_reverse2(subsidiary), V40, V34), flatten(V34, V33), length(V33, V32), if(V32 &lt; 1, V37 is 0, (maplist(attr_reverse2(percentsubsidiary), V40, V31), maplist(maplist(...), V31, V30), maplist(...), ..., ...))), attr(V53, branch, V22), flatten(V22, V21), length(V21, V20), if(V20 &lt; 1, V19 is 0, (attr(V53, percentbranch, V18), maplist(value_reverse2(V46), V18, V17), maplist(attr_reverse2(subsidiary), V22, V16), flatten(V16, V15), length(...), if(...))), plus(V49, V37, V4), plus(V4, V19, V3), plus(V3, V2, V1) </pre>
--	--	--	--	--	--	---