Improving Data Quality Through Effective Use of Data Semantics

Stuart Madnick, Hongwei Zhu

Working Paper CISL# 2005-08

October 2005

Composite Information Systems Laboratory (CISL)
Sloan School of Management, Room E53-320
Massachusetts Institute of Technology
Cambridge, MA 02142

Improving Data Quality Through Effective Use of Data Semantics

Stuart Madnick*, Hongwei Zhu

MIT Sloan School of Management, 30 Wadsworth Street, E53-320, Cambridge, MA 02142, USA

Abstract

Data quality issues have taken on increasing importance in recent years. In our research, we have discovered that many "data quality" problems are actually "data misinterpretation" problems – that is, problems caused by heterogeneous data semantics. In this paper, we first identify semantic heterogeneities that, when not resolved, often cause data quality problems. We discuss the especially challenging problem of aggregational ontological heterogeneity, which concerns how complex entities and their relationships are aggregated. Then we illustrate how COntext INterchange (COIN) technology can be used to capture data semantics and reconcile semantic heterogeneities, thereby improving data quality.

Keywords: Data Quality, Data Semantics, Semantic Heterogeneity, Ontology, Context

1. Introduction

Data quality issues have taken on increasing importance in recent years. In our research, we have discovered that many "data quality" problems are actually "data misinterpretation" problems – that is, problems with data semantics. To illustrate how complex this can become, consider Fig. 1. This data summarizes the P/E ratio for DaimlerChrysler obtained from four different financial information sources – all obtained on the same day within minutes of each other. Note that the four sources gave radically different values for P/E ratio.

Source	P/E Ratio
ABC	11.6
Bloomberg	5.57
DBC	19.19
MarketGuide	7.46

Fig. 1. P/E ratios for DaimlerChrysler.

The obvious questions to ask are: "Which source is correct?" and "Why are the other sources wrong – i.e., of bad data quality?" The possibly surprising answer is: they are all correct!

The issue is, what do you really mean by "P/E ratio"[1]. The answer lies in the multiple interpretations and uses of the term "P/E ratio" in financial circles. The earnings are for the entire year in some sources but in one source are only for the last quarter. Even when earnings are for a full year, are they:

- the last 12 months?

* Corresponding author. Tel: +1 617 253 6671; fax: +1 617 253 3321.
Email addresses: smadnick@mit.edu (S. Madnick), mrzhu@mit.edu (H. Zhu).

[1] Some of these sites even provide a glossary which gives a definition of such terms and they are very concise in saying something like "P/E ratio" is "the current stock price divided by the earnings". As it turns out, this does not really help us to explain the differences.

- the last calendar year?
- the last fiscal year? or
- the last three historical quarters and the estimated current quarter – a popular usage?

Such information, which we call *context*, is often not explicitly captured in a form that can be used by the query answering system to reconcile semantic differences in data from different sources. Serious consequences can result from not being aware of the differences in contexts and data semantics. Consider a financial trader that used DBC to get P/E ratio information yesterday and got 19.19. Today he used Bloomberg and got 5.57 (low P/E's usually indicate good bargains) – thinking that something wonderful had happened he might decide to buy many shares of DaimlerChrysler today. In fact, nothing had actually changed, except for changing the source that he used. It would be natural for this trader (after possibly losing a significant amount of money due to this decision) to feel that he had encountered a data quality problem.

We would argue that what appeared to be a data quality problem is actually a data misinterpretation problem. The data source did not have any "error," the data that it provided was exactly the data that it intended to provide – it just did not have the meaning that the receiver expected. In other words, the issue is not what is right or wrong, it is about how data in one context can be used in a different context.

Before going any further, it should be noted that if all sources and all receivers of data always had the exact same meanings, this problem would not occur. This is a desirable goal – one frequently sought through standardization efforts. But these standardizations are often unsuccessful for many reasons [18], e.g., there are legitimate needs for representing and interpreting data in different ways to suit different purposes[2]. This creates the well known problem of semantic heterogeneities that exist pervasively in information systems. It is crucial that we understand the kinds of heterogeneity and develop technologies to provide data that is consistent with receiver preference, thereby improve the data quality at the receiver end. Such a solution can have significant impact as the estimated cost of information mishandling in businesses worldwide is tremendous [19].

In the next section, we exemplify the semantic heterogeneities that, when not reconciled, can cause data quality problems. Then, we present the Context Interchange technology and show how it can be used to capture data semantics and dynamically reconcile semantic differences between the sources and the receivers. This technology supports the uniformity required by any specific receiver, at same time, it supports heterogeneity by preserving the autonomy of all sources and receivers. We conclude in the last section and point out directions for future research.

2. Heterogeneous Data Semantics

There have been a number of studies that identify and catalog various semantic heterogeneities [3,11,16,17]. A subset of the heterogeneities are related to data quality that we address in this paper and can be categorized into two main groups: (1) representational heterogeneity and (2) ontological heterogeneity. Data semantics can sometimes change over time; therefore, representational and ontological semantics of a source or a receiver can evolve, resulting in temporal semantic heterogeneities. These categories are summarized in Fig. 2 and explained next.

[2] A full discussion of all the difficulties with standardization is beyond the scope of this paper. It is worth noting that the "Treaty of the Meter" committing the U.S. government to go metric was initially signed in 1875.

Representational	Ontological
Temporal	

	Snapshot example	Temporal example
Representational	*Currency*: EUR in source *v.* USD in receiver	*Currency*: DEM until 12/31/98 EUR since 1/1/99
Ontological	*Profit*: Net excl. tax in source *v.* Gross incl. tax in receiver	*Profit*: Net until 1999 Gross since 2000

Fig. 2. Data quality related semantic heterogeneities.

Representational heterogeneity – The same concept can have different representations in different sources and receivers. For example, the day of *March 4, 2005* can be represented as *03/04/05*, *05-03-04*, etc; packaging dimensions can be expressed in metric units or in English units; price data can be quoted in different currencies and using different scale factors.

Temporal Representational heterogeneity – The representation in a source or a receiver can also change. For example, a price database in Turkey may list prices in millions of Turkish liras (TRL), but after the Turkish New Lira (TRY) was introduced on January 1, 2005, it may start to list prices in unit of Turkish New Lira[3].

Ontological heterogeneity – The same term is often used to refer to similar but slightly different concepts. Known and quantifiable relationships often exist amongst these concepts. We have already seen an example of this regarding the multiple interpretations of "P/E ratio" in Fig. 1.

Temporal Ontological heterogeneity – In addition, in the same source or receiver, the meaning of a term can shift over time, often due to changes of needs or requirements. For example, *profit* can refer to *gross profit* that includes all taxes collected on behalf of government, or *net profit* that excludes those taxes. Net profit can be calculated from gross profit by deducting the taxes, and vice versa. The "Profit" field in a database may refer to net profit at one time and refer to gross profit at another, because of changes in reporting rules.

Aggregational Ontological heterogeneity – Another variation can be that the profit of a firm may include that from majority owned subsidiaries in one case, and excludes them in another case (possibly due to different reporting rules in different countries or for different purposes.) Aggregational ontological heterogeneity has to do with what is included/aggregated in the meaning of an entity or a relationship. A specific example of this situation, sometimes called corporate housekeeping, will be presented later.

Representational and ontological data semantics is often embedded in the explicit data and the implicit assumptions; semantic heterogeneities exist when the implicit assumptions in the sources do not match the implied expectations of the receivers. They must be reconciled to ensure the correct interpretation of the data by the receivers. In the following, we will use several examples to illustrate the semantic heterogeneities and their effects on data quality.

Example 1: Temporal Representational Semantics (Yahoo Historical Stock Prices)

When the same company stock is traded at different stock exchanges around the world, there may be *small* price differences between exchanges, creating arbitraging opportunities (i.e., buying low in one place and selling high at another). Fig. 3 gives an example of how big the

[3] The following fact may help explain why this could be case: 1 USD = 1.39 million TRL; 1 TRY = 1 million TRL; it would be cumbersome to list many 0's if prices were listed in unit of TRL.

differences can be – on the left are IBM stock prices in Frankfurt, Germany, on the right are that in New York, USA. We notice that the values between the two exchanges during the same time period are *huge* (comparing the values in the brackets); in addition, there is an abrupt price drop in Frankfurt while the prices in New York are stable (comparing the values in the circles). This is quite unusual! Again, one may start to question about data quality in the sources, but in fact, the peculiarities in the data are due to semantic mismatches – the implied currencies not only differ between the two exchanges, but also changed in Frankfurt from Deutschmark (DEM) to Euro (EUR); the currency in New York has always been USD. This is an example of representational heterogeneity that also evolves over time. ■

Frankfurt, Germany

PRICES

Date	Open	High	Low	Close	Volume	Adj Close*
8-Jan-99	162.20	165.00	162.20	163.00	5,220	78.96
7-Jan-99	162.00	162.50	160.00	160.00	3,647	77.51
6-Jan-99	160.90	164.00	160.90	164.00	23,616	79.45
5-Jan-99	154.00	155.00	154.00	155.00	5,975	75.09
4-Jan-99	155.00	158.00	155.00	155.50	7,024	75.33
30-Dec-98	311.50	311.50	309.00	309.00	28,324	149.69
29-Dec-98	314.90	314.90	313.30	314.00	22,313	152.12
28-Dec-98	314.80	314.80	314.00	314.00	26,640	152.12
23-Dec-98	303.50	305.00	302.50	305.00	38,363	147.76
22-Dec-98	293.60	297.50	293.00	294.00	29,899	142.43
21-Dec-98	282.50	293.00	282.50	293.00	27,603	141.94

* Close price adjusted for dividends and splits.

New York, USA

PRICES

Date	Open	High	Low	Close	Volume	Adj Close*
8-Jan-99	191.00	192.00	185.63	187.56	4,593,100	89.99
7-Jan-99	187.94	192.38	187.00	190.19	4,155,300	91.25
6-Jan-99	190.31	192.75	188.50	188.75	4,775,300	90.56
5-Jan-99	183.00	189.88	182.81	189.63	4,955,900	90.98
4-Jan-99	185.00	186.50	181.50	183.00	4,077,500	87.80
31-Dec-98	186.75	187.19	183.50	184.38	1,933,800	88.46
30-Dec-98	186.88	188.63	186.31	186.75	2,410,300	89.60
29-Dec-98	188.63	188.94	187.00	187.13	1,881,700	89.78
28-Dec-98	186.50	189.94	186.00	189.25	2,637,200	90.80
24-Dec-98	184.75	187.94	184.06	187.94	1,527,700	90.17
23-Dec-98	182.69	185.38	181.13	185.00	3,537,800	88.76
22-Dec-98	177.50	183.00	175.25	182.25	4,353,600	87.44
21-Dec-98	171.56	178.94	171.56	176.38	3,744,500	84.62

* Close price adjusted for dividends and splits.

Fig. 3. IBM stock prices at different exchanges (from Yahoo).

Example 2: Aggregational Ontological Semantics (Corporate Householding)

The rapidly changing business environment has witnessed widespread and rapid changes in *corporate structure* and *corporate relationships*. Regulations, deregulations, acquisitions, consolidations, mergers, spin-offs, strategic alliances, partnerships, joint ventures, new regional headquarters, new branches, bankruptcies, franchises … all these make understanding corporate relationships an intimidating job. Moreover, the same two corporation entities may relate to each other very differently when marketing is concerned than when auditing is concerned. That is, interpreting corporate structure and corporate relationships depends on the task at hand. To understand the challenges, let us consider some typical, simple, but important questions that an organization, such as IBM or MIT, might have about their relationships:

[MIT]: "How much did we buy from IBM this year?"

[IBM]: "How much did we sell to MIT this year?"

The first question frequently arises in the Procurement and Purchasing departments of many companies, as well as at more strategic levels. The second question frequently arises in the Marketing departments of many companies and is often related to Customer Relationship Management (CRM) efforts, also at more strategic levels. Logically, one might expect that the answers to these two questions would be the same – but frequently they are not, furthermore one often gets multiple different answers even within each company.

These types of questions are not limited to manufacturers of physical goods, a financial services company, such as Merrill Lynch, might ask:

[Merrill Lynch]: "How much have we loaned to IBM?"

[IBM]: "How much do we owe Merrill Lynch?"

On the surface, these questions may sound like both important and simple questions to be able to answer. In reality, there are many reasons why they are difficult and have multiple differing answers.

At least three types of challenges must be overcome to answer questions such as the ones illustrated above: (a) representational semantic heterogeneity, (b) entity aggregational ontological heterogeneity, and (c) relationship aggregational ontological heterogeneity. The first two concern *what IBM or MIT is*, and the third one concerns *how IBM and MIT are related*. These challenges provide a typology for understanding what is sometimes called the Corporate Householding, as illustrated in Fig. 4 and explained below.

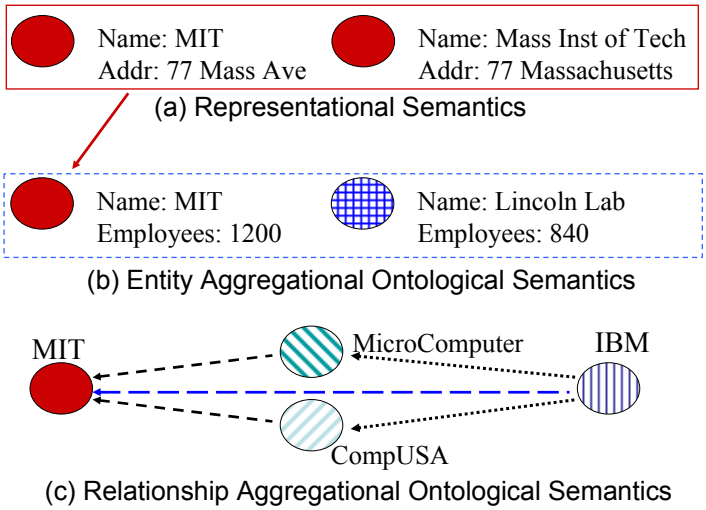


Fig 4. Typology for Corporate Householding Challenges.

(a) *Representational Semantics*. In general, there are rarely complete unambiguous universal identifiers for either people or companies. Two names may refer to the same physical entity even though they were not intended to create confusions in the beginning. For example, the names "James Jones", "J. Jones", and "Jim Jones" might appear in different databases, but actually be referring to the same person. The same problems exist for companies. As shown in Fig. 4(a), the names "MIT", "Mass Inst of Tech", "Massachusetts Institute of Technology", and many other variations might all be used to refer to the exact same entity. They are different simply because the users of these names choose to do so. Thus, we need to be able to identify the same entity correctly and efficiently when naming confusion happens. This problem has also been called Identical Entity Instance Identification [10]. That is, the same identical entity might appear as multiple instances (i.e., different forms) – but it is still the same entity.

(b) *Entity Aggregational Ontological Semantics*. Even after we have determined that "MIT", "Mass Inst of Tech", "Massachusetts Institute of Technology" all refer to the same entity, we need to determine what exactly is that entity? That is, what other unique entities are to be included or aggregated into the intended definition of "MIT." For example, 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 in Lexington and physically separated from the main campus of MIT (sometimes referred to as the "on-campus MIT"), which is in

Cambridge. Lincoln Lab has a budget of about $500 million, which is about equal to the rest of MIT.

Problem arises when people ask questions such as "How many employees does MIT have?" or "How much was MIT's budget last year?". In the case illustrated in Fig. 4(b), should the Lincoln Lab employees or budget be included in the "MIT" calculation and in which cases they should not be? Under some circumstances, the MIT Lincoln Lab number should be included, whereas under other circumstances they should not be. We refer to these differing circumstances as different contexts. To know which case applies under each category of circumstances, we must know the context. As noted earlier, we refer to this type of problem as Entity Aggregational Ontological heterogeneity.

(c) *Relationship Aggregational Ontological Semantics*. Furthermore, even after we have resolved the aggregation of entities, we still need to determine the relationships between the entities. As illustrated in Fig. 4(c), the buying/selling relationships between MIT and IBM can be direct or indirect through other channels. Consider our original question – for IBM: "How much did we sell to MIT this year?" The answer to question varies depending on the aggregation of sales channels. For example, under some circumstances, only the direct sales from IBM to MIT are included, whereas under other circumstances, sales through other channels (e.g., through partners, retailers, etc.) are also included.

In summary, the answers to the questions can be dramatically different because of the multiple situations that exist. Different answers do not signify that some answers are wrong; all answers can be correct under their corresponding circumstances, i.e., in their own contexts.■

Example 3: Temporal Ontological Semantics (Code v. What Code Denotes)

In everyday communications and in various information systems, it is very common that we refer to things using various codes, e.g., product codes of a company, subject numbers in a university catalog, and ticker symbols commonly used to refer to company stocks. Codes are sometimes reused in certain systems, thus the same code can denote different things at different times. For example, subject number "6.891" at MIT has been used to denote "Multiprocessor Synchronization", "Techniques in Artificial Intelligence", "Computational Evolutionary Biology", and many other subjects in the past decade. As an another example, ticker symbol "C" used to be the symbol for Chrysler; after Chrysler merged with Daimler-Benz in 1997, the merged company chose to use "DCX"; on December 4, 1998, the symbol "C" was assigned to Citigroup, which was listed as "CCI" before this change. ■

Example 4: Temporal Aggregational Ontological Semantics (Yugoslavia)

To study the economic and environmental development of different parts of the world, one often needs longitudinal data from various sources. In the past 30 years, certain regions have gone through significant restructuring, e.g., one country breaking up into several countries. Such dramatic changes can make it difficult to use data from multiple sources or even from a single source. As an example, suppose a Balkans specialist is interested in studying the CO_2 emissions in the region of former *Yugoslavia* during 1980-2000 and prefers to refer to the region (i.e. the geographic area of the territory of former Yugoslavia) as Yugoslavia. Data sources like the Carbon Dioxide Information Analysis Center (CDIAC)[4] at Oak Ridge National Laboratory organize data by country. Fig. 5 lists some sample data from CDIAC. Yugoslavia as a country, whose official name is *Socialist Federal Republic of Yugoslavia* in 1963-1991, was broken into

[4] http://cdiac.esd.ornl.gov/home.html

five independent countries in 1991: *Slovenia*, *Croatia*, *Macedonia*, *Bosnia and Herzegovina*, and *Federal Republic of Yugoslavia* (also called *Yugoslavia* for short in certain other sources). Suppose prior to the break-up the specialist had been using the following SQL query to obtain data from the CDIAC source:

```
Select CO2Emissions from CDIAC where Country = "Yugoslavia";
```

Before the break-up, "Yugoslavia" in the receiver coincidentally referred to the same geographic area as to what "Yugoslavia" in the source referred, therefore, the query worked correctly for the receiver until 1991. After the break-up, the query stopped working because no country is named "Yugoslavia" (or had the source continued to use "Yugoslavia" for the Federal Republic of Yugoslavia, the query would return wrong data because "Yugoslavia" in the source and the receiver refer to two different geographic areas). ■

Country	**Year**	**CO2Emissions**
...	...	...
Yugoslavia	1990	35604
Yugoslavia	1991	24055
Slovenia	1992	3361
Croatia	1992	4587
Macedonia	1992	2902
Bosnia-Herzegovinia[5]	1992	1289
Federal Republic of Yugoslavia	1992	12202
...	...	...

Fig. 5. Sample CO2 emissions data from CDIAC.

These examples demonstrate that poor data quality can result from representational and ontological heterogeneities between the sources and the receivers. They also suggest that we can improve data quality by resolving these heterogeneities. In simple cases, this can be done manually by the receivers. But in most practical cases that involve a large number of sources and data elements, a manual reconciliation will be difficult and error prone. In the next section, we will introduce the Context Interchange technology and show how it is used to improve data quality by automatically reconciling semantic differences between the sources and the receivers.

3. Improving Data Quality with Context Interchange Technology

3.1. Context Interchange Overview

COntext INterchange (COIN) [7,9,10] is a knowledge-based mediation technology that enables meaningful use of heterogeneous databases where there are semantic differences. With the COIN technology, a user (i.e., information receiver) is relieved from keeping track of various source contexts and can use the sources as if they were in the user context. Semantic differences are identified and reconciled by the mediation service of COIN. The overall COIN system includes not only the mediation infrastructure and services, but also a wrapping technology and middleware services for accessing the source information and facilitating the integration of the mediated results into end-user applications (see Fig. 6). The wrappers are physical and logical gateways providing a uniform access to the disparate sources over the network [5].

[5] Correct spelling is "Herzegovina", which is an error; we do not address this kind of data quality problem in this paper.

The set of Context Mediation Services comprises a Context Mediator, a Query Optimizer, and a Query Executioner. The Context Mediator is in charge of the identification and resolution of potential semantic differences induced by a query. This automatic detection and reconciliation of differences present in different information sources is made possible by accessing the knowledge of the underlying application domain, as well as informational content and implicit assumptions associated with the receivers and sources. These bodies of declarative knowledge are represented in the form of a shared ontology, a set of elevation axioms, and a set of context definitions, which we explain below.

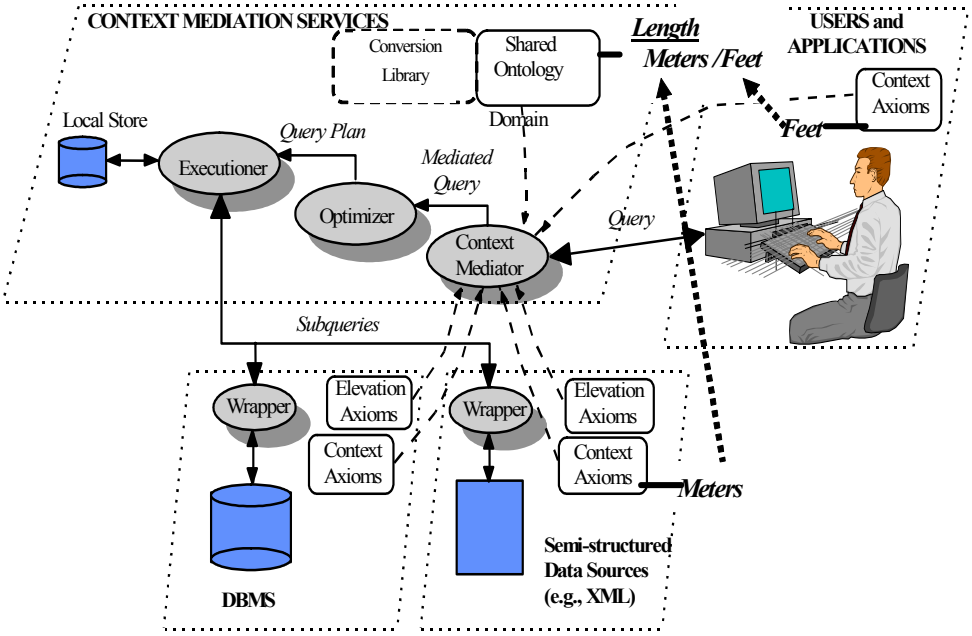


Fig. 6. The architecture of the context interchange system.

The input to the mediator is a user query assuming that all sources were in the user context. The result of the mediation is a mediated query that includes the instructions on how to reconcile the semantic differences in different contexts involved in the user query. To retrieve the data from the disparate information sources, the mediated query is then transformed into a query execution plan, which is optimized, taking into account the topology of the network of sources and their capabilities. The plan is then executed to retrieve the data from the various sources.

For the mediator to identity and reconcile semantic difference, necessary knowledge about data semantics needs to be formally represented. For purposes of knowledge representation, COIN adopts an object-oriented logic data model, based on the formal theory of F-Logic [13], a first order logic with syntactic sugar to support object-orientation (e.g., inheritance, polymorphism, etc.). Loosely speaking, the COIN data model has three elements, for which we give a brief overview below and provide further explanations in the next sub-section:

- The Shared Ontology is a collection of concepts, also called rich types or semantic types, that define the domain of discourse (e.g., "Length");
- Elevation Axioms for each source identify the semantic objects (instances of semantic types) corresponding to source data elements, define integrity constraints, and specify general properties of the sources;

- Context Descriptions annotate the different interpretations of the semantic objects in the different sources or from a receiver's point of view (e.g., "Length" might be expressed in "Feet" or "Meters").

Finally, there is a conversion library which provides conversion functions for resolving potential semantic differences. The conversion functions can be defined declaratively or can use external services or external procedures. The relevant conversion functions are gathered and composed during mediation to resolve the differences. No global or exhaustive pair-wise definition of the conflict resolution procedures is needed. The mediator is implemented using abductive constraint logic programming (ACLP) [12], which not only rewrites queries to reconcile semantic differences, but also performs semantic query optimization.

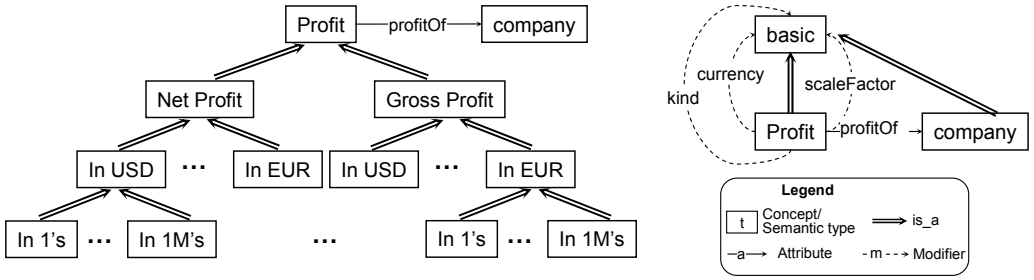
3.2. Representing Heterogeneous Semantics using Ontology and Contexts

To a certain extent, ontology modeling and entity-relationship modeling [2,4] share the same objective of providing a formal way of representing things in the real world. An ontology usually consists of a set of terms corresponding to a set of predefined concepts (similar to entities), relationships between concepts, and certain constraints. There are two types of binary relationships between concepts: *is_a*, and *attribute*. The *is_a* relationship indicates that a concept is more specific (or conversely, more general) than another (e.g., the concept of *net profit* is more specific than the concept of *profit*); the *attribute* relationship simply indicates that a concept is an attribute of another (e.g., the *company* concept is the *profit_of* attribute of the *profit* concept).

A high level concept can have various specializations. As shown in Fig. 7(a) below, *profit* can have specializations such as *gross profit* and *net profit*, each can be further specialized to use various currencies, which can be further specialized to use different scale factors (e.g., in thousands or millions). Since the purpose of ontology is to share knowledge, it is tempting to fully describe these specializations in the ontology so that there will be no ambiguity in the semantics of the concepts. However, the ontology of this approach is difficult to develop because (1) the ontology often consists of a large number of concepts, and (2) it requires various parties engaged in knowledge sharing to agree on the precise definitions of each concept in the ontology.

The COIN ontology departs from the above approach, as shown in Fig.7(b). It only requires the parties to agree on a small set of general concepts. Detailed definitions (i.e., specializations) of the general concepts are captured outside the ontology as localized context descriptions. The context descriptions usually correspond to the implicit (and sometimes evolving) assumptions made by the data sources and receivers. To facilitate context description, the COIN ontology includes a special kind of attribute, called the *modifier*. Contexts are described by assigning values to modifiers.

These two different approaches are illustrated in Fig. 7 using the company profit example.



(a) a fully specified ontology of *profit* (b) a COIN ontology of *profit*

Fig 7. Fully specified ontology v. COIN ontology.

The fully specified ontology in Fig. 7(a) contains all possible variations/specializations of the concept *profit*, organized in a multi-level and multi-branch hierarchy. Each leaf node represents a most specific *profit* concept. For example, the leftmost node at the bottom represents a profit concept that is a "net profit in 1's of USD". In contrast, the COIN ontology contains only concepts in higher levels (e.g, *profit*), further refinements of these concepts do not appear in the ontology; rather, they are specified outside the ontology and are described using modifiers (e.g., if in a context, the profit data is "net profit in 1's USD", the context can be described by assigning appropriate values to the modifiers, i.e., *kind*="net", *scaleFactor*= "1", and *currency*="USD").

Compared with the fully specified approach, the COIN approach has several advantages. First, a COIN ontology is usually much simpler, thus easier to manage. Second, it facilitates consensus development, because it is relatively easier to agree on a small set of high level concepts than to agree on every piece of detail of a large set of fine-grained concepts. And more importantly, a COIN ontology is much more adaptable to changes. For example, when a new concept "*net profit* in billions of South Korean Won" is needed, the fully specified ontology needs to be updated with insertions of new nodes. The update requires the approval of all parties who agreed on the initial ontology. In contrast, the COIN approach can accommodate this new concept by adding new context descriptions without changing the ontology.

Another important distinction is in the provision of conversions for reconciling semantic differences. Other approaches tend to provide pair-wise conversions between the data elements that correspond to the leaf nodes in the fully-specified ontology, e.g., a conversion between the data of "net profit in 1's of USD" and the data of "gross profit in millions of EUR". We call such conversions *composite conversions*. In the COIN approach, conversions are provided for each modifier; such conversions are called *component conversions*. All pair-wise composite conversions are automatically composed by the mediator using the component conversions. In the example illustrated in Fig.7, with three component conversions (i.e., one for each modifier), the COIN mediator can compose all composite conversions as needed between any pair of the leaf nodes in the fully specified ontology.

With the ontological constructs and the component conversions in the COIN approach, all data quality related semantic heterogeneities identified in the previous section can be represented and processed. We will use the simplified ontology for the Yahoo historic stock price example, shown below in Fig. 8, to explain how this is done.

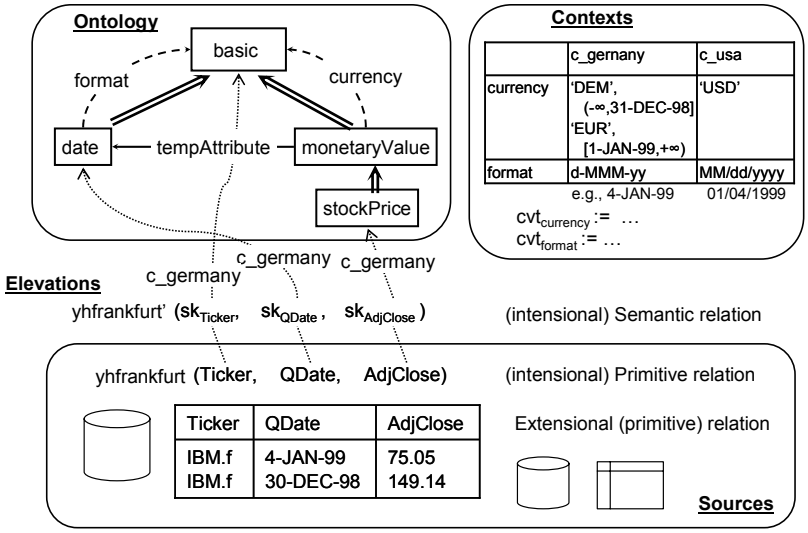


Fig. 8. COIN ontology, contexts, and elevations for historical stock price example.

Ontology and contexts are shown in the upper half of Fig. 8. There are two modifiers in the ontology: *format* for describing different date formats, and *currency* for describing different currencies. We show two sample contexts: (1) *c_germany* for the Yahoo site that provides stock prices at Frankfurt Stock Exchange (we call the source *yhfrankfurt*); and (2) *c_usa* for receivers, say in the U.S. We describe contexts by assigning values to the modifiers in the ontology, as shown in the upper-right corner of the figure.

Formally, we use F-Logic[6] formulas (sometimes called rules) to describe contexts. Temporal semantics can be described using multi-valued modifiers, i.e., a modifier can have different values in different time periods within a context, and we call such context a *temporal context*. For example, to describe that in context *c_germany*, the currency is "DEM" until December 31, 1998, and is "EUR" since January 1, 1999, we use the following two rules to assign different values to modifier *currency* in the two respective time periods:

$$
\begin{aligned}
&O : monetaryValue, currency(c\_germany, O) : basic \vdash \\
&\quad O[currency@c\_germany \rightarrow currency(c\_germany, O)] \leftarrow \\
&\qquad currency(c\_germany, O)[value@c\_germany \rightarrow "DEM"], \\
&\qquad rcvContext(C), O[tempAttribute \rightarrow T], T[value@C \rightarrow T_v], \\
&\qquad skolem(date, "31\text{–}DEC\text{–}98", c\_germany, 1, cste("31\text{–}DEC\text{–}98"))[value@C \rightarrow T_c] \\
&\qquad T_v \le T_c .
\end{aligned}
$$

$$
\begin{aligned}
&O : monetaryValue, currency(c\_germany, O) : basic \vdash \\
&\quad O[currency@c\_germany \rightarrow currency(c\_germany, O)] \leftarrow \\
&\qquad currency(c\_germany, O)[value@c\_germany \rightarrow "EUR"], \\
&\qquad rcvContext(C), O[tempAttribute \rightarrow T], T[value@C \rightarrow T_v], \\
&\qquad skolem(date, "1\text{–}JAN\text{–}99", c\_germany, 1, cste("1\text{–}JAN\text{–}99"))[value@C \rightarrow T_c] \\
&\qquad T_v \ge T_c .
\end{aligned}
$$

Readers are referred to [13] for the details of the F-logic language; here we provide a brief explanation on how it is used for context descriptions. A modifier is represented as a parameterized *method* of an object and expressed within the square brackets following the object.

[6] Although F-Logic is the internal representation used with COIN, a user-friendly interface make it unnecessary for any user (either context administrator or query user) to know F-Logic.

For example, in the head of the first rule above, *currency* modifier is the currency method of the semantic object *O*, whose type is *monetaryValue*. Given the parameter *c_germany*, the method returns an object, of type *basic*, represented by a *Skolem* function (also called a *Skolem* object) with *c_germany* and *O* as the parameters. The body of the rule (after "←") indicates that the *value* of the *Skolem* object in context *c_germany* is "DEM" when the *tempAttribute* attribute of *O* is before "31-DED-98".

The *currency* modifier in context *c_usa* and the *format* modifier in both contexts can be specified similarly. These specifications are simpler because the modifier value is not time dependent. For example, the rule below states that in context *c_usa* the currency is "USD":

$$O : monetaryValue, currency(c\_usa, O) : basic \vdash$$
$$O[currency@c\_usa \rightarrow currency(c\_usa, O)] \leftarrow$$
$$currency(c\_usa, O)[value@c\_usa \rightarrow "USD"].$$

Component conversions for modifiers *format* and *currency* are specified using F-logic rules, as well:

$$D : date \vdash$$
$$D[cvt@format, C_r, M_s, M_r, U \rightarrow V] \leftarrow$$
$$datecvt(U, M_s, V, M_r).$$

$$M : monetaryValue \vdash$$
$$M[cvt@currency, C_r, M_s, M_r, U \rightarrow V] \leftarrow$$
$$M[tempAttribute \rightarrow T], T[value@C_r \rightarrow T_v], olsen'(F_c', T_c', R', D'),$$
$$F_c'[value@C_r \rightarrow M_s], T_c'[value@C_r \rightarrow M_r], D'[value@C_r \rightarrow T_v],$$
$$R'[value@C_r \rightarrow R_v], V = U * R_v.$$

Both rules use external programs/services. The first rule uses the external program *datecvt* to perform date format conversions; the second rule uses the external service *olsen* to obtain the exchange rate between a pair of currencies on a given day. Wrappers [5] are used for these external programs/services so that they can be accessed like relational databases.

In the lower half of Fig. 8, we show the data source *yhfrankfurt* with its schema and two sample records. The elevation axioms map each column of a relation to a concept in the ontology and associate each column with a context. Thus, for each relation (which we call *primitive relation*) in the source there is a *semantic relation*. Each attribute in a semantic relation is a (meta-) semantic object (i.e., an instance of a semantic type), which has access to the context descriptions and component conversions defined for the modifiers of the corresponding semantic type.

3.3. Reconciling Semantic Heterogeneities Through Mediation

Once contexts are recorded for all sources and receivers, and component conversions are provided, a receiver can query any collection of sources as if all they were in the receiver context. The mediator will intercept the query, compare the contexts involved, introduce appropriate component conversions, and generate a mediated query that reconciles the semantic differences.

The implementation of the mediator is based on the formal theory of abductive constraint logic programming (ACLP) [12], where abductive inference is interleaved with concurrent constraint solving. The constraint store collects all abducible predicates generated by abductive inference. All abducible predicates are treated as constraints, the consistency of which is handled by constraint solvers defined using the declarative language Constraint Handling Rules (CHR) [8]. For example, the descriptions of a temporal context often involve comparisons of time

values; when these comparison predicates are abduced, they are treated as constraints and processed by temporal constraint solvers, which generates a common time period during which all involved modifier are singly valued. We also use CHR to solve symbolic equations [6] and perform semantic query optimization. Detailed descriptions of the implementation can be found in [7,9,10].

Below, we use the Yahoo historical stock price example to illustrate how COIN is used to provide meaningful data to the receiver without the receiver being burdened to keep track of semantic heterogeneities.

Suppose a receiver in context *c_usa* wants to retrieve historical stock prices at Frankfurt Stock Exchange. The receiver prefers to see the adjusted close price in USD and the date in MM/dd/yyyy format (e.g., 01/10/1999). Using the web wrapper technology, we can superimpose the following relational schema to the data source at Yahoo Finance website:

```
YHFrankfurt<Ticker,QDate,AdjClose,
            StartMonth,StartDay,StartYear,EndMonth,EndDay,EndYear>
```

where the last six attributes corresponds to the month, day, and year of "Start Date" and "End Date". These attributes are necessary only because the source is not able to accept date range specified as inequalities on the "QDate" attribute; instead, it is only able to accept equalities on the last six attributes. Like semantic differences, such capability differences should be processed by the system, not the receivers. Therefore, we let the source expose a much simpler schema:

```
<Ticker, QDate, AdjClose>
```

When COIN is used, the receiver can use the data sources as if they were in the receiver context; in this case, the receiver can issue the following query against the simplified schema to obtain adjusted close prices of IBM stock in Frankfurt during December 20, 1998 and January 10, 1999:

Q1:
```
select QDate, AdjClose from YHFrankfurt
where Ticker="IBM.f" and QDate >="12/20/1998" and QDate =<"01/10/1999";
```

This query cannot be executed as is because of the source's inability in evaluating inequalities on "QDate"; even if it could, it does not return meaningful data to the user. Comparing the context definitions for the source and the receiver in Fig. 8, we notice that there are currency and date format differences. The currency assumed in the source also changed within the specified date range. These capability restrictions, semantic differences and the change of semantics are recognized by the COIN mediator, which subsequently generates the following mediated Datalog[7] query:

[7] Datalog is a set-oriented, non-procedural, and function-free logic programming language designed for use as a database language. A Datalog query is the logical equivalent of a SQL query. We use the predicate *answer* to simulate projection; other predicates correspond to relations or selection conditions. For example, a Datalog query for SQL query Q1 is: *answer(QDate,Price):-yhfranfurt("IBM.f",QDate,Price,_,_,_,_,_), QDate>="12/20/1998", QDate=<"01/01/1999"*. A "-" in a predicate represents an unnamed argument of the predicate. Further detail of Datalog can be found in [1].

MDQ1:
```
answer(V6, V5):-
    olsen("DEM", "USD", V4, V3),
    datecvt(V3, "MM/dd/yy", V6, "MM/dd/yyyy"),
    datecvt(V2, "d-MMM-yy", V6, "MM/dd/yyyy"),
    V5 is V1 * V4,
    yhfrankfurt("IBM.f", V2, V1, "Dec", "20", "1998", "Dec", "31", "1998").

answer(V6, V5):-
    olsen("EUR", "USD", V4, V3),
    datecvt(V3, "MM/dd/yy", V6, "MM/dd/yyyy"),
    datecvt(V2, "d-MMM-yy", V6, "MM/dd/yyyy"),
    V5 is V1 * V4,
    yhfrankfurt("IBM.f", V2, V1, "Jan", "1", "1999", "Jan", "10", "1999").
```

The corresponding SQL query generated by the COIN mediator is:

MQ1:
```
select datecvt.date2, (yhfrankfurt.adjClose*olsen.rate)
from   (select 'DEM', 'USD', rate, ratedate from   olsen
        where  exchanged='DEM' and expressed='USD') olsen,
       (select date1, 'MM/dd/yy', date2, 'MM/dd/yyyy' from   datecvt
        where  format1='MM/dd/yy' and format2='MM/dd/yyyy') datecvt,
       (select date1, 'd-MMM-yy', date2, 'MM/dd/yyyy'
        from   datecvt
        where  format1='d-MMM-yy' and format2='MM/dd/yyyy') datecvt2,
       (select 'IBM.f', qDate, adjClose, 'Dec', '20', '1998', 'Dec', '31', '1998'
        from   yhfrankfurt where  Ticker='IBM.f'
        and    StartMonth='Dec' and StartDay='20' and StartYear='1998'
        and    EndMonth='Dec' and EndDay='31' and EndYar='1998') yhfrankfurt
where  datecvt2.date1 = yhfrankfurt.qDate
and    datecvt.date2 = datecvt2.date2 and  olsen.ratedate = datecvt.date1
union
select datecvt3.date2, (yhfrankfurt2.adjClose*olsen2.rate)
from   (select 'EUR', 'USD', rate, ratedate from   olsen
        where  exchanged='EUR' and  expressed='USD') olsen2,
       (select date1, 'MM/dd/yy', date2, 'MM/dd/yyyy' from   datecvt
        where  format1='MM/dd/yy' and  format2='MM/dd/yyyy') datecvt3,
       (select date1, 'd-MMM-yy', date2, 'MM/dd/yyyy' from   datecvt
        where  format1='d-MMM-yy' and  format2='MM/dd/yyyy') datecvt4,
       (select 'IBM.f', qDate, adjClose, 'Jan', '1', '1999', 'Jan', '10', '1999'
        from   yhfrankfurt where  Ticker='IBM.f'
        and    StartMonth='Jan' and StartDay='1' and StartYear='1999'
        and    EndMonth='Jan' and EndDay='10' and EndYear='1999') yhfrankfurt2
where  datecvt4.date1 = yhfrankfurt2.qDate and datecvt3.date2 = datecvt4.date2
and    olsen2.ratedate = datecvt3.date1
```

The SQL syntax is a bit more verbose, so we will examine the more concise Datalog query MDQ1. It has two sub-queries: one for the time period from December 20, 1998 to December 31, 1998, the other for the time period from January 1, 1999 to January 10, 1999. This is because the currency assumed in the source is Deutschmark in the first period and is Euro in the second period, each needing to be processed separately.

Let us focus on the first sub-query for the moment, which is reproduced in Fig. 9 with line numbers and annotations added. Line 6 queries the *yhfrankfurt* source. Notice that the date range has been translated to equalities of the six attributes of *month*, *day*, and *year* of start date and end date of the actual schema; the values for month are now in the format required by the source, i.e., "Dec" for December. Variable V2 corresponds to "QDate", V1 corresponds to "AdjClose". None of them are in line 1 to be reported back to the user; the code in lines 2-5 has the instructions on how to transform them to V6 and V5 as values to be returned to the user.

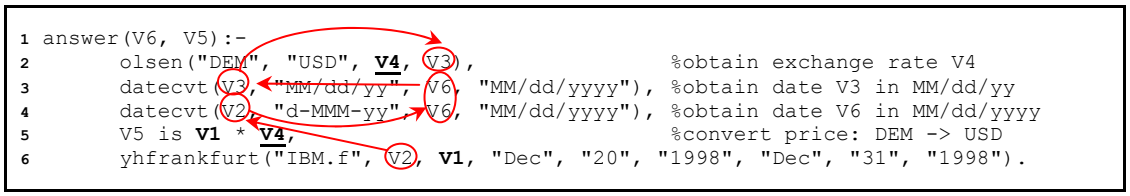


Fig. 9. Reconciliation of semantic differences in MDQ1.

The procedural reading of the code is:

- *line 4* converts "QDate" (V2) from the source format to the format expected by user (V6), i.e., from "d-MMM-yy" format (e.g., 20-Dec-98) to "MM/dd/yyyy" format (e.g, 12/20/1998);
- *line 3* converts V6 (from *line 4*) to V3 so that V3 has the format expected by source *olsen*, i.e., it converts date format from "MM/dd/yyyy" (e.g, 12/20/1998) to "MM/dd/yy" (e.g, 12/20/98);
- *line 2* queries the *olsen* source to obtain exchange rate (V4) between Deutschmark (DEM) and U.S. dollar (USD) for the date given by V3; and
- *line 5* converts "AdjClose" (V1) to USD using the exchange rate (V4) from *line 2*.

The second sub-query is almost the same except that it deals with a different date range within which the currency difference is EUR v. USD instead of DEM v. USD.

When the mediated query is executed, the user receives data instances[8] as shown in the left pane of Fig. 10. For comparison, we also show the "raw" data from the source; notice that the unusual abrupt price drop in the raw data (which is actually due to the change in currencies) no longer appears in the mediated data.

Mediated results

QDate	AdjClose
01/08/1999	91.65
01/07/1999	90.10
01/06/1999	92.94
01/05/1999	88.28
01/04/1999	88.61
12/30/1998	89.27
12/29/1998	90.54
12/28/1998	90.14
12/23/1998	88.06
12/22/1998	84.84
12/21/1998	84.96

(user format) (in USD)

Non-mediated results (original data)

QDate	AdjClose	
8-Jan-99	78.67	EUR
7-Jan-99	77.22	
6-Jan-99	79.15	
5-Jan-99	74.81	
4-Jan-99	75.05	
30-Dec-98	149.13	DEM
29-Dec-98	151.54	
28-Dec-98	151.54	
23-Dec-98	147.2	
22-Dec-98	141.89	
21-Dec-98	141.41	

(original format)

Fig. 10. Mediated and non-mediated data instances.

We have also applied the COIN technology to the other examples. For detailed descriptions, interested readers are referred to [15] for the corporate householding scenario (where we illustrate how entity aggregational ontological heterogeneity is resolved), and [21] for the Yugoslavia example (where we show how temporal entity aggregational heterogeneity is resolved). We are currently extending COIN to address the problem of relationship aggregational ontological heterogeneity.

[8] Mediated results are rounded for easy reading.

4. Concluding Remarks

We are in the midst of exciting times – the opportunities to access and integrate diverse information sources, most especially the enormous number of sources provided over the web, are incredible but the challenges are considerable. It is sometimes said that we now have "more and more information that we know less and less about." This can lead to serious "data quality" problems caused due to improperly understood or used data semantics, as illustrated by the situation described in Fig. 11.

Unit-of-Measure mixup tied to loss of $125 Million Mars Orbiter

"NASA's Mars Climate Orbiter was lost because engineers did not make a simple conversion from English units to metric, an embarrassing lapse that sent the $125 million craft off course … The navigators [JPL] **assumed metric units** of force per second, or newtons. In fact, the numbers **were in pounds** of force per second as supplied by Lockheed Martin [the contractor]."

Source: Kathy Sawyer, *Boston Globe*, October 1, 1999, pg. 1.

Fig 11. Examples of consequences of misunderstood data semantics

The effective use of data semantics and context knowledge is needed to enable us to overcome the challenges described in this paper and more fully realize the opportunities. A particularly interesting aspect of the context mediation approach described is the use of context to describe the expectations of the receiver as well as the semantics assumed by the sources.

In this paper, we identify the kinds of semantic heterogeneities that can cause data quality problems. Then we show how COIN technology can be used to capture context knowledge and improve data quality by automatically reconciling semantic differences between the sources and the receivers. An important aspect of this approach is that COIN is a flexible and scalable technology. As shown in [20], the number of component conversions that need to be specified depends on the number of modifiers in the ontology and the number of unique values of each modifier; it does not depend on the number of sources and receivers involved, *N*. When *N* is large, COIN approach requires one to several orders of magnitude less conversions to be specified than other approaches that hard-code the conversions. This is not surprising because the mediator can be thought of as an automatic code generator – it can generate composite conversions using a small set of component conversions and supply appropriate parameters depending on contexts. Through demonstrations, we have shown that COIN can be used to solve many data quality problems caused by semantic heterogeneities.

We find that the interplay of data quality and data semantics is interesting and has practical significance. This paper presents only some initial work in this area. For future research, we plan to identify other semantic heterogeneities that affect data quality either in the source or from the receiver's perspective. Then we extend COIN-based system to facilitate automatic reconciliation of such heterogeneities. Ultimately, we expect to develop a unifying framework for analyzing data quality from data semantics perspective and applying semantic interoperability technologies to improving data quality.

Acknowledgements

Work reported herein has been supported, in part, by Banco Santander Central Hispano, Citibank, Defense Advanced Research Projects Agency (DARPA), D & B, Fleet Bank, FirstLogic, Merrill Lynch, MITRE Corp., MIT Total Data Quality Management (TDQM)

Program, PricewaterhouseCoopers, Singapore-MIT Alliance (SMA), Suruga Bank, and USAF/Rome Laboratory.

References

[1] S. Ceri, G. Gottlob, and L. Tanca, "What You Always Wanted to Know About Datalog (And Never Dared to Ask)", *IEEE Transactions on Knowledge and Data Engineering,* **1**(1)**,** 146-166, 1989.
[2] P.P. Chen, "The Entity-Relationship Model: Toward a Unified View of Data," ACM Transactions on Database Systems, 1(1) (1976) 1-36.
[3] H. T. El-Khatib, M. H. Williams, L. M. MacKinnon, and D. H. Marwick, "A framework and test-suite for assessing approaches to resolving heterogeneity in distributed databases," Information & Software Technology, vol. 42, pp. 505-515, 2000.
[4] R. Elmasri, J. Weeldreyer, A. Hevner, "The Category Concept: an Extension to the Entity-Relationship Model", Data & Knowledge Engineering, 1(1) (1985) 75-116.
[5] A. Firat, S. E. Madnick, and M. D. Siegel, "The Cameleon Web Wrapper Engine," in: Workshop on Technologies for E-Services (TES'00), Cairo, Egypt, 2000.
[6] A. Firat, S. E. Madnick, and B. Grosof, "Financial Information Integration in the Presence of Equational Ontological Conflicts," in: 12th Workshop on Information Technology and Systems (WITS), Barcelona, Spain, 2002.
[7] A. Firat, "Information Integration using Contextual Knowledge and Ontology Merging," PhD Thesis, MIT, 2003.
[8] T. Frühwirth, "Theory and Practice of Constraint Handling Rules," Journal of Logic Programming, vol. 37, pp. 95-138, 1998.
[9] C. H. Goh, "Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Systems," Ph.D. Thesis, MIT, 1997.
[10] C. H. Goh, S. Bressan, S. Madnick, and M. Siegel, "Context Interchange: New Features and Formalisms for the Intelligent Integration of Information," ACM TOIS, vol. 17, pp. 270-293, 1999.
[11] V. Kashyap and A. P. Sheth, "Semantic and Schematic Similarities Between Database Objects: A Context-Based Approach," VLDB Journal, vol. 5, pp. 276-304, 1996.
[12] A. C. Kakas, A. Michael, and C. Mourlas, "ACLP: Abductive Constraint Logic Programming," Journal of Logic Programming, vol. 44, pp. 129-177, 2000.
[13] M. Kiffer, G. Laussen, and J. Wu, "Logic Foundations of Object-Oriented and Frame-based Languages," J. ACM, vol. 42, pp. 741-843, 1995.
[14] S. E. Madnick and R. Wang, "The Inter-Database Instance Identification Problem in Integrating Autonomous Systems," in: 5th International Conference on Data Engineering (ICDE'89), Los Angeles, CA, 1989.
[15] S. Madnick, R. Wang, and X. Xian, "The Design and Implementation of a Corporate Householding Knowledge Processor to Improve Data Quality," Journal of Management Information Systems, vol. 20, pp. 41-69, 2004.
[16] C. F. Naiman and A. M. Ouskel, "A classification of semantic conflicts in heterogeneous database systems," Journal of Organizational Computing, vol. 5, pp. 167-193, 1995.
[17] S. Ram and J. Park, "Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data and Schema-Level Semantic Conflict," IEEE Transactions on Knowledge and Data Engineering, vol. 16, pp. 189-202, 2004.
[18] A. Rosenthal, L. Seligman, and S. Renner, "From Semantic Integration to Semantics Management: Case Studies and a Way Forward," ACM SIGMOD Record, vol. 33, pp. 44-50, 2004.
[19] Z. Schreiber, "Semantic Information Management (SIM)," Unicorn, White Paper, 2003.
[20] H. Zhu and S. E. Madnick, "Context Interchange as a Scalable Solution to Interoperating Amongst Heterogeneous Dynamic Services," in: 3rd Workshop on eBusiness (WEB), Washington, D.C., 2004.
[21] H. Zhu, S. E. Madnick, and M. D. Siegel, "Representation and Reasoning About Changing Semantics in Heterogeneous Data Sources," in Semantic Web and Databases: Second International Workshop (SWDB 2004), vol. LNCS 3372, C. Bussler, V. Tannen, and I. Fundulaki, Eds., 2005, pp. 127-139.

Stuart Madnick is the John Norris Maguire Professor of Information Technology, Sloan School of Management and Professor of Engineering Systems, School of Engineering at the Massachusetts Institute of Technology. He has been a faculty member at MIT since 1972. He has served as the head of MIT's Information Technologies Group for more than twenty years. He has also been a member of MIT's Laboratory for Computer Science, International Financial Services Research Center, and Center for Information Systems Research. Dr. Madnick is the author or co-author of over 250 books, articles, or reports including the classic textbook, Operating Systems, and the book, The Dynamics of Software Development. His current research interests include connectivity among disparate distributed information systems, database technology, software project management, and the strategic use of information technology. He is presently co-Director of the PROductivity From Information Technology Initiative and co-Heads the Total Data

Quality Management research program. He has been active in industry, as a key designer and developer of projects such as IBM's VM/370 operating system and Lockheed's DIALOG information retrieval system. He has served as a consultant to corporations, such as IBM, AT&T, and Citicorp. He has also been the founder or co-founder of high-tech firms, including Intercomp, Mitrol, and Cambridge Institute for Information Systems, iAggregate.com and currently operates a hotel in the 14th century Langley Castle in England. Dr. Madnick has degrees in Electrical Engineering (B.S. and M.S.), Management (M.S.), and Computer Science (Ph.D.) from MIT. He has been a Visiting Professor at Harvard University, Nanyang Technological University (Singapore), University of Newcastle (England), Technion (Israel), and Victoria University (New Zealand).

Hongwei Zhu is a Research Scientist at the Information Quality Program at MIT. His research interests include the development of technologies to enable meaningful information sharing, and theories to address policy issues related to information sharing/data reuse. He holds a Ph.D. in Technology, Management, and Policy form MIT, where he worked on the Context Interchange Project at the Sloan School of Management. Prior to coming to MIT, he was a software engineer and IT consultant developing web based solutions for both private sector and government agencies.