Financial Information Integration In the Presence of Equational Ontological Conflicts

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

Abstract

While there are efforts to establish a single international accounting standard, there are strong current and future needs to handle heterogeneous accounting methods and systems. We advocate a context-based approach to dealing with multiple accounting standards and equational ontological conflicts. In this paper we first define what we mean by equational ontological conflicts and then describe a new approach, using Constraint Logic Programming and abductive reasoning, to reconcile such conflicts among disparate information systems. In particular, we focus on the use of Constraint Handling Rules as a simultaneous symbolic equation solver, which is a powerful way to combine, invert and simplify multiple conversion functions that translate between different contexts. Finally, we demonstrate a sample application using our prototype implementation that demonstrates the viability of our approach.

1. Introduction

The recent accounting scandals are underlining the need for more transparent and accurate access to information in financial statements. A recent survey carried out by *McKinsey* found that 90 per cent of institutional investors favored a single international accounting standard, but they differed over what it should be[1]. The likelihood of a single international accounting standard coming to dominate anytime soon is quite slim. This is further complicated by the complexities and localities involved in the accounting practices of different countries (e.g. the UK views the proposed standards as actually reducing the quality of their corporate reporting.) Even within a single country, there are good reasons why many investors need access to data in various forms, such as pro forma numbers that offer insights into the performance of companies' core business by excluding one-time events that can skew the financial results.

There is, however, a lack of information technology products that can conveniently collect and integrate data from disparate financial statements and present them to the users in the way they are accustomed to see or in one of several accounting standards. In this paper, we present a framework that can gracefully handle the representation of different data semantics and integrate information from diverse sources in the presence of equational ontological conflicts (EOC).

In the next sections, we first define what we mean by EOC, and provide specific examples. Then, we explain how we resolve EOC in our extended COntext INterchange (ECOIN) framework by using Constraint Logic Programming techniques, specifically through the use of Constraint Handling Rules (CHR.) Finally, we provide a simple e-business example from our prototype implementation that demonstrates the viability of our approach.

2. Equational Ontological Conflicts

In financial statements many data items are derived from other simpler data items. For example, *Price Earnings Ratio* is calculated by dividing price per share by earnings per share. However, this definition is subject to multiple interpretations, as it does not specify whether the earnings are "*trailing*"[2] or "*forward*"[3], or more importantly what is included in the earnings. In fact, when we collected *Price Earnings Ratios* for a specific company, Daimler-Benz, from several financial sources on the same day the numbers differed significantly, because of the differences in the interpretation of *earnings* (see Figure 1.) A closer examination reveals that these variations are not caused by erroneous reporting, but attributable to definitional differences among data sources.

SOURCE	P/E RATIO
ABC	11.6
Bloomberg	5.57
DBC	19.19
MarketGuide	7.46

Figure 1 Key Financials for Daimler-Benz (from [Madnick 01]).

Financial concepts such as "Revenues", "Expenses" and "Profits" are ontologically[4] distinct

---

[1] "Investors want single accounting standard", Michael Skapinker , Financial Times (London), July 8, 2002.

[2] Trailing earnings are earnings in the last 12 months.

[3] Forward earnings are based on future earning estimates.

[4] An "ontology" is a formally defined vocabulary expressing the semantics of a term. Some vocabulary terms can be defined in terms of others using logical statements, e.g., equations.

but have interdependences that can be expressed as equations, such as "Profit = Revenues – Expenses." We refer to the heterogeneity in the way data items are calculated from other data items in terms of definitional equations, as *equational ontological conflicts*. Such conflicts in accounting methods are quite widespread not only between different countries, but also within the same country [Firat et al. 02]. For example, *The Wall Street Journal* and *S&P* use different methods to calculate the *P/E Ratios* for the *Standard & Poor's 500-stock index*. *The Wall Street Journal* divides the combined market capitalization of the 500 companies currently in the index by their *most recently reported four quarters of earnings*, while S&P updates earnings statistics for the index *just once a quarter* and doesn't revise earnings from previously reported quarters to account for additions or deletions to the index[5].

As long as the context used by each source of financial data is known, there is nothing wrong with a multiplicity of calculation methods – i.e., of equational ontologies. Yet, problems occur once companies' financial numbers, crunched by analysts, enter a vast information food chain, where they are repeated, often without explanation, in hundreds of news sources, and end up being used out of context. This becomes even more challenging when there is the need to combine or compare data obtained from multiple sources with differing contexts.

We advocate a context-based solution to this problem by making the context of the data items of each source explicit (i.e., how they are derived from other data items) and adjusting their values to different contexts by recalculating them when necessary using the context information – including the definitional equations associated with each context. In the next section we provide the details of our approach that can help decision makers get the data they want the way they want them.

3. Extended Context Interchange Approach to Equational Ontology Conflicts

3.1 Extended COntext INterchange (ECOIN)

The Extended COntext INterchange (ECOIN) approach to resolving equational conflicts is an extension of our core COIN framework [Goh et al. 99]. In the COIN approach, data providers and users furnish their implicit assumptions on the data they provide or receive using context axioms expressed in first order logic. The overall COIN approach, detailed in [Bressan et al. 00], includes the mediation infrastructure and services as well as wrapping technology and middleware services for accessing source information and facilitating the integration of the mediated results into end-users' applications.

In our framework, equational ontological conflicts are not handled by making changes to the ontology, for example by introducing new types and defining equational relationships between their values. Making changes in ontologies is likely to be a time-consuming and difficult process, and is better avoided. Furthermore, in many cases, such an approach would result in an explosion of new ontology types to handle all of the possible variations. In ECOIN we use modifiers, a special type of attributes that collectively define the context of a data source, to specify the implicit aspects of an ontological term.

For example, in Figure 2, although data from all three sources, *eStore*, *KidWorld* & *eToys*, map to the same ontology term *Price,* they exhibit definitional differences that were not foreseen, or that were perhaps deliberately left out during ontology creation. In this case, we specify the definitional differences between different price elements by using the *type* modifier. Conversion libraries are then used to define the relationships, or in this specific case equations, between different modifier values. In conversion libraries, it is enough for new additions simply to establish a connection to the network of conversion functions, and then our system automatically takes care of combining, inverting and simplifying them through the use of Dijkstra's shortest path algorithm and the use of Constraint Handling Rules.

3.2 Constraint Handling Rules

Our extension involves simultaneous symbolic equation solving techniques through the use of *Constraint Handling Rules* (CHR[6]), a high-level language extension of *Constraint logic programming (CLP),* especially designed for writing constraint solvers. This extension, coupled with our context based approach to detecting and reconciling data semantics, provides an elegant and powerful solution to the problem of detecting and resolving equational conflicts.

---

[5] Moving Target: What's the P/E Ratio? Well, Depends on What Is Meant by Earnings --- Terms Like `Operating,' `Core,' `Pro Forma' Catch Fire, Leave Investors Muddled ---`Earnings Before Bad Stuff', Jonathan Weil, Wall Street Journal, Aug 21, 2001.

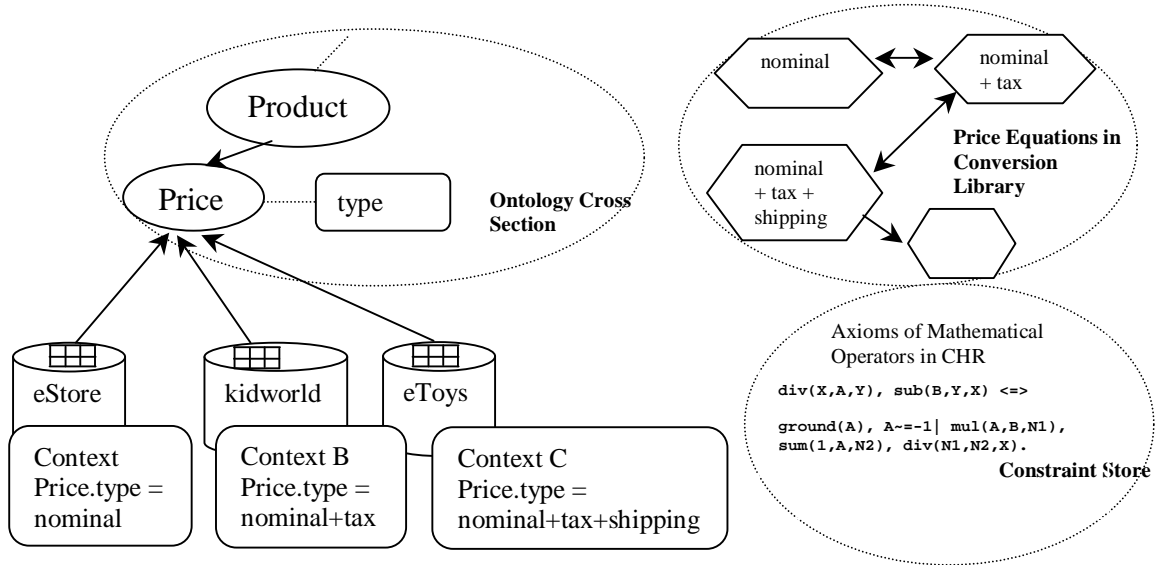[6] For more information on CHR the reader may refer to [Frühwirth 98].

Figure 2. Equational Conflict Handling in ECOIN

Our use of CHR combines the advantages of logic programming and constraint solving by providing a declarative approach to solving problems, while at the same allowing users to employ special purpose algorithms in the sub problems. The constraint solver works by repeatedly applying constraint rules and rewriting constraints into simpler ones until they are solved. CHR has been used to encode a wide range of problems, including ones involving terminological and temporal reasoning.

In our original COIN system, we used CHR in semantic query optimization by pruning our query plans with the use of integrity constraints. In the ECOIN we add new constraints for basic mathematical operations such as addition, subtraction, multiplication and division. These constraints roughly correspond to the basic axioms of group theory for the specified operators. For example we use the following constraint to define the division operation on ground items[7].

```
div_ground ::= div(X,Y,Z) <=>
ground(X), ground(Y), Y ~=0 | Z is
X/Y.
```

In addition we specify the interaction constraints between the operators for further simplification. For example, the following set of constraints

```
div_sub_to_mul_sum_div        ::=
div(X,A,Y), sub(B,Y,X) <=>
ground(A),   A~=-1   |   mul(A,B,N1),
sum(1,A,N2), div(N1,N2,X).
```

```
div_sub_to_mul_sum_div        ::=
div(X,A,Y), sub(B,Y,X) <=>
nonground(A)        |        mul(A,B,N1),
sum(1,A,N2), div(N1,N2,X).
```

correspond to the following general simplification:

$$Y = X/A,\ X = B - Y\ \Rightarrow\ X = (A*B)\ /\ (1+A)\quad [A<>-1\ \text{or A not instantiated}]$$

3.3 Abductive Logic Programming

In ECOIN, we combine abductive logic programming [Kakas et al. 93] techniques with CHR and rewrite naïve user queries into mediated queries that reconcile semantic conflicts among data sources. In this process, we first collect all *abducibles*[8] and post them into our constraint store. Among these abducibles are a number of arithmetic operations that relate ontological terms to each other. For example Pro Forma Earnings that exclude acquisition costs and taxes would be related to Net Income through a series of subtraction operations. The constraint store more generally contains integrity constraints, Clark's free equality[9] (CFQ), and arithmetic axioms. The constraint processing engine then processes these *abducibles* through constraint simplification and propagation. This process optimizes the mediated query through the application of integrity constraints and CFQ axioms, and solves simultaneous symbolic equations that are used in the conversion functions.

We extend our abduction algorithm in the way it handles conversion functions that are used to relate

---

[7] Ground terms in logic programming correspond to terms that assume values and are not variables.

[8] Informally, abducibles are atoms appearing only in the body of some logical clause in a logic program. For more information see [Kakas et al. 93]

[9] The axioms defining the consistency of a set of equations between variables and constants.

objects between different contexts. Specifically we allow conversion functions to be declared as *commutative* between different contexts, if

$$[ObjectValue(target) \leftarrow ConversionFunction(context(source), context(target)) \circ ObjectValue(source)] \rightarrow [ObjectValue(source) \leftarrow ConversionFunction^{-1}(context(target), context(source)) \circ ObjectValue(target)]$$

These are processed in part using graph-based algorithms, which operate over a graph defined to have nodes corresponding to contexts in which the commutative conversion functions are bi-directional paths between context nodes, both directions derivable from a single conversion function. In many domains, such as financial information, specification of commutative conversion functions cuts the number of conversion function declarations significantly. We use Dijkstra's shortest path algorithm to find the best combination of conversion functions that can translate between two context nodes according to specified optimization criterion, e.g., to minimize the total cost of the conversion functions employed.

4. Prototype Implementation

We have implemented ECOIN using the Eclipse Prolog Engine, its extended CHR library, and the Java programming language. ECOIN also has other features not mentioned in this paper, such as ontology merging and source selection in the presence of equational conflicts. The system implements the abduction engine and constraint based symbolic reasoning in Eclipse, and the distributed query executioner in Java.

We have developed several sample applications to demonstrate the features of our system. One such example, an online toy store application, consists of several toy store data sources, which report prices in different ways. Just like the existence of multiple interpretations of earnings in financial information systems, the concept "price" has a different meaning in each toy store database. As shown earlier in Figure 2, in the *eToys* database the price includes tax and shipping, whereas in the *Kid's World* database the price includes tax but not shipping and in the *eStore* context, the price is just the nominal price and includes neither tax nor shipping.

In Figure 3, we show a trace of the execution of an SQL query, that requests names and prices of products that are cheaper at *eToys* compared to *Kid's World*. The user asks the query in the *eStore* context, which means that the data returned will be adjusted to fit the assumptions stated in the *eStore* context[10]. Figure 3a shows the conflicts detected between the

data sources and the target user context. It also shows conversion functions that are needed to translate between different contexts. Note that the system chains through two conversion functions to go from *c_et(eToys context)* to *c_kw(Kid's World context)* to *c_es(eStore context)*. In Figure 3b, the mediated datalog query is shown (the datalog representation is used internally.) This query is a rewriting of the original query with conflicts detected, conversion functions, and symbolic equation solving applied. In Figure 3c, we show the mediated Query as expressed in SQL. Finally, in Figure 3d, we show the result of the query when executed against the data sources.

5. Conclusions

In this paper, we have described equational ontology conflicts, an important kind of semantic heterogeneity in financial information systems and developed a new context-based declarative approach to deal with these conflicts. While our work does not fully automate the process of financial information integration, it significantly reduces the work needed in current integration projects. We require data sources to declare their context axioms using a common context declaration scheme, but allow the flexibility of subscribing to different ontologies. Our approach derives its power from the combination of logic programming and constraint solving techniques. This approach is important for resolving semantic interoperability problems among heterogeneous information sources, which may range from financial analysis to other domains (such as shopping for toys). We found that CHR is a versatile tool that can be used for simultaneous symbolic equation solving in addition to semantic query optimization and terminological reasoning. We believe that multiple accounting standards will co-exist and intelligent financial information systems that can deal with equational conflicts will be very valuable for the financial community. Our work is a contribution in that direction.

6. Acknowledgements

---

[10] In this case, the user has "adopted" the eStore context. Alternatively, the user could choose a unique context.

Receiver is in context **c_es**:
Price is nominal price(no tax, no shipping, etc. included) Product names are expressed numerically

```
SQL    select eToys.Product, eToys.FinalPrice
       from kidworld, eToys
       where eToys.Product=kidworld.Product
       and eToys.FinalPrice<kidworld.TotalPrice;
```

Context c_es

Stage
- Naive Datalog
- SQL Translation
- Context Sensitive Datalog
- Execution
- Conflict Detection
- Mediation

Submit Reset

Kid's World is in context **c_kw**:
Price includes tax but not shipping Product names are expressed using characters

TotalPrice and FinalPrice from kidworld and eToys elevate to the same ontological term price with different contexts

Result

SemanticType	Column	Source	Modifier	Modifier value in source context	Modifier value in target context	Conversion Function
price	TotalPrice	kidworld(Product, TotalPrice, Tax)	type	c_kw : nominal+tax	c_es : nominal	attr(O, product, P), attr(P,taxRate, T), value(T, Ctxt, TV),multiply (Vs,TV,Tax), plus (Vs,Tax,Vt)
price	FinalPrice	eToys(Company, State, Product, FinalPrice, ShippingCost, IntSurcharge)	type	c_kw : nominal+tax	c_es : nominal	attr(O, product, P), attr(P,taxRate, T), value(T, Ctxt, TV),multiply (Vs,TV,Tax), plus (Vs,Tax,Vt)
price	FinalPrice	eToys(Company, State, Product, FinalPrice, ShippingCost, IntSurcharge)	type	c_et : nominal+tax+shipping	c_kw : nominal+tax	attr(O, product, P), attr(P,shippingCost, T), value(T, Ctxt, TV), plus(Vs,TV,Vt)
product	Product	kidworld(Product, TotalPrice, Tax)	format	c_kw : alpha	c_es : numeric	name_id_map(V2, V1)

This conversion function expresses the mathematical relationship between Price objects in c_kw and c_es contexts

My Computer

eToys is in context **c_et**:
Price includes tax and shipping Product names are expressed using characters

Figure 3a Conflict Detection

Conversion functions defined between c_et & c_kw and c_kw & c_es are combined --with possible inversions– to convert between c_et & c_es

Format conversion both for eToys & kidworld

Price conversion for eToys

```
Result    answer('V16', 'V15') :-
          kidworld('V14', 'V13', 'V12'),
          name_id_map('V16', 'V14'),
          tax_table('V11', 'V10'),
          eToys('V9', 'V11', 'V14', 'V8', 'V7', 'V6'),
          'V5' is 1 + 'V10',
          'V4' is 'V8' - 'V7',
          'V15' is 'V4' / 'V5',
          tax_table('MA', 'V3'),
          'V2' is 1 + 'V3',
          'V1' is 'V13' / 'V2',
          'V15' < 'V1'.
```
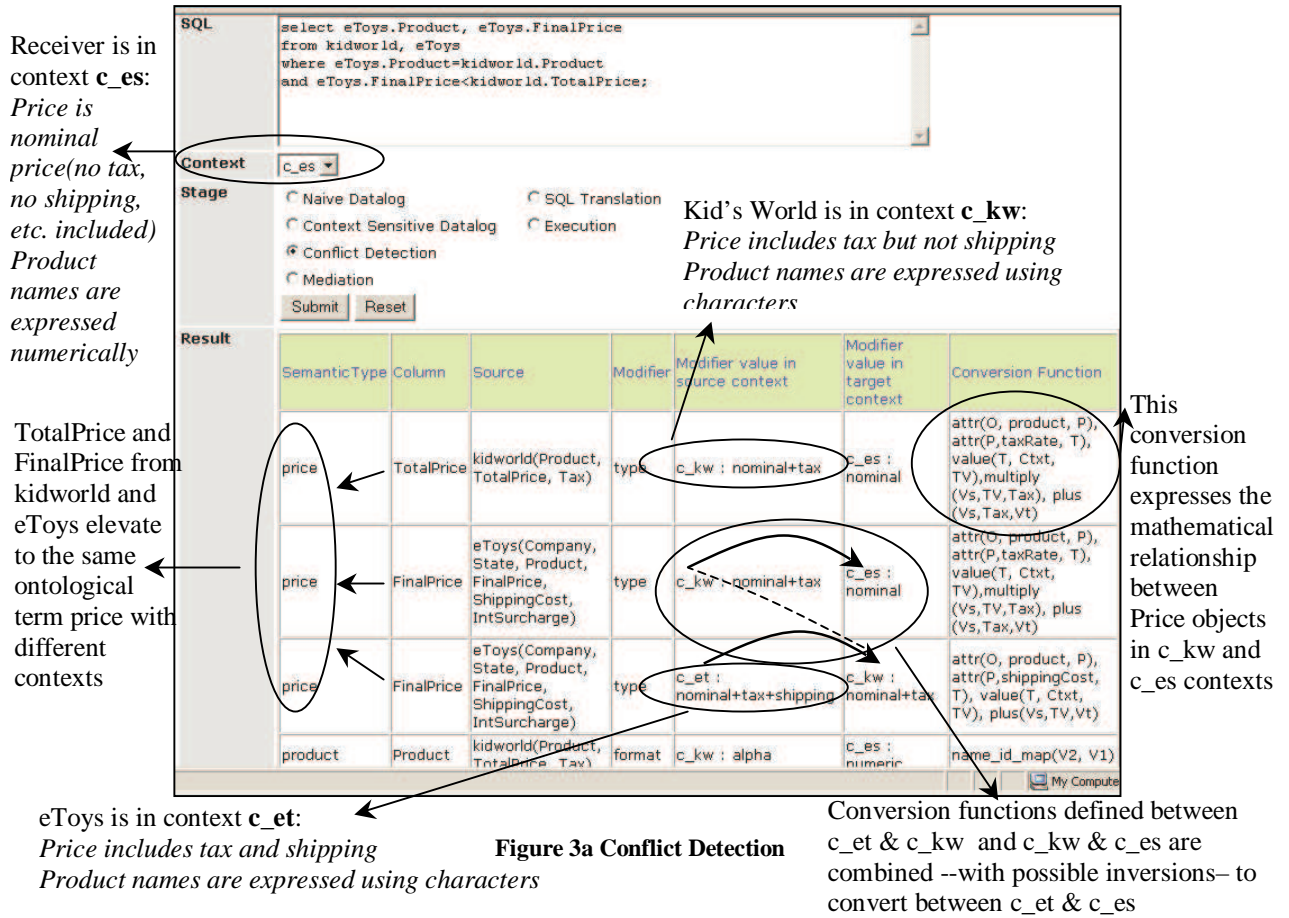
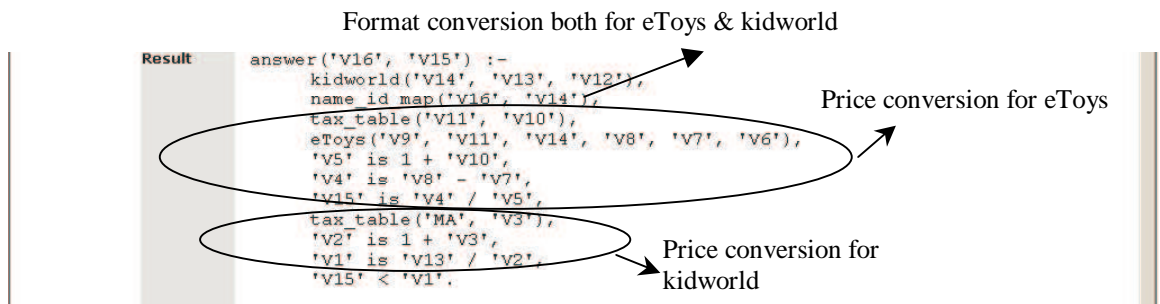Price conversion for kidworld

Figure 3b Mediated Datalog Query

```
Result      select name_id_map.ProductID, ((etoys.FinalPrice-etoys.ShippingCost)/(1+tax_table.TaxRate))
            from    (select Product, TotalPrice, Tax
                     from   kidworld) kidworld,
                    (select ProductID, ProductString
                     from   name_id_map) name_id_map,
                    (select State, TaxRate
                     from   tax_table) tax_table,
                    (select Company, State, Product, FinalPrice, ShippingCost, IntSurCharge
                     from   etoys) etoys,
                    (select 'MA', TaxRate
                     from   tax_table
                     where  State='MA') tax_table2
            where   kidworld.Product = name_id_map.ProductString
            and     name_id_map.ProductString = etoys.Product
            and     tax_table.State = etoys.State
            and     ((etoys.FinalPrice-etoys.ShippingCost)/(1+tax_table.TaxRate)) < (kidworld.TotalPrice/(1+tax_table2.TaxRate))
```

External sources used in conversions

Equations obtained from the constraint store after combining & constraint processing conversion functions

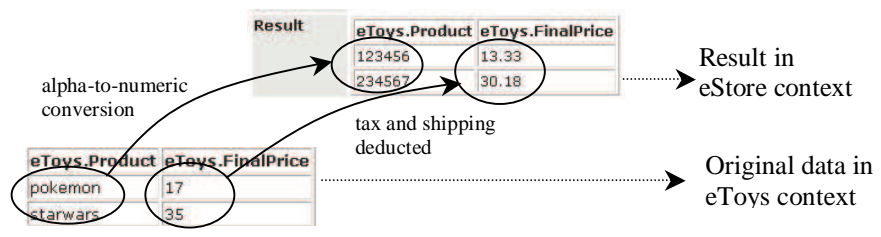**Figure 3c Mediated Query in SQL**

alpha-to-numeric conversion

tax and shipping deducted

Result in eStore context

Original data in eToys context

eToys.Product	eToys.FinalPrice
123456	13.33
234567	30.18

eToys.Product	eToys.FinalPrice
pokemon	17
starwars	35

Figure 3d Execution of the query

References

[Bressan et al. 00] S. Bressan, C. Goh, N. Levina, S. Madnick, A. Shah, M. Siegel, "Context Knowledge Representation and Reasoning in the Context Interchange System", *Applied Intelligence: The International Journal of Artificial Intelligence, Neutral Networks, and Complex Problem-Solving Technologies,* Vol 12 (2): 165-179, September 2000.

[Firat et al. 02] Firat, A., Madnick, S., Grosof, B. "Knowledge Integration to Overcome Ontological Heterogeneity: Challenges from Financial Information Systems", Proceedings of ICIS 2002, Barcelona, Spain.

[Frühwirth 98] Frühwirth, Th., "Theory and Practice of Constraint Handling Rules", Special Issue on Constraint Logic Programming (P. Stuckey and K. Marriot, Eds.), *Journal of Logic Programming*, Vol 37(1-3): 95-138, October 1998,

[Goh et al. 99] C. H. Goh, S. Bressan, S. Madnick, and M. Siegel, "Context Interchange: New Features and Formalisms for the Intelligent Integration of Information", *ACM Transactions on Office Information Systems*, July 1999

[Kakas et al. 93] Kakas, A.C., Kowalski, R.A., and Toni, F. (1993). "Abductive logic programming," *Journal of Logic and Computation*, 2(6): 719-770.

[Madnick 01] Madnick, S. (2001) "The Misguided Silver Bullet: What XML will and will NOT do to help Information Integration," *Proceedings of the Third International Conference on Information Integration and Web-based Applications and Services*, IIWAS2001, Linz, Austria, September: 61-72.