MIT Sloan School of Management

MIT Sloan Working Paper 4543-05
CISL Working Paper No. 2005-05

May 2005

Multi-dimensional Ontology Views via Contexts in the ECOIN Semantic Interoperability Framework

Aykut Firat, Stuart Madnick, Frank Manola

Multi-dimensional Ontology Views via Contexts in the ECOIN Semantic Interoperability Framework

Aykut Firat, Stuart Madnick, Frank Manola

Working Paper CISL# 2005-04

May 2005

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

Multi-dimensional Ontology Views via Contexts in the ECOIN Semantic Interoperability Framework

Aykut Fırat
Northeastern University
Boston, MA 02115, USA
a.firat@neu.edu

Stuart Madnick
Massachusetts Institute of Technology
Cambridge, MA 02142, USA
smadnick@mit.edu

Frank Manola
Independent Consultant
Wilmington, MA 01887, USA
fmanola@acm.org

Abstract

This paper describes the coupling of contexts and ontologies for semantic integration in the ECOIN semantic interoperability framework. Ontological terms in ECOIN correspond to multiple related meanings in different contexts. Each ontology includes a context model that describes how a generic ontological term can be modified according to contextual choices to acquire specialized meanings. Although the basic ECOIN concepts have been presented in the past, this paper is the first to show how ECOIN addresses the case of "single-ontology with multiple contexts" with an example of semantic integration using our new prototype implementation.

Introduction

With the globalization of information over the internet, achieving semantic interoperability among heterogeneous and autonomous systems has become an increasingly important endeavor. A key issue in applying ontologies in practical semantic interoperability problems has proven to be reducing the amount of work needed to agree on a shared model, to describe the different assumptions made by sources and receivers, and to express (or generate) the mappings required to transform the data when moving it between different sources and receivers. In this paper, we discuss the ECOIN[1] approach and how it is able to address this important issue.

In the ECOIN semantic interoperability framework, ontologies describe both the shared domain model and the ways in which contexts can specialize the shared model. This is done by providing a terminology with *generic* meanings, which are *modified* in local contexts to express *specialized* meanings. A *context model* coupled with the shared model explicitly specifies possible modification dimensions of an ontological term. For example, the meaning (and representation) of a generic term like *airfare* can be modified along the *currency*, *coverage*, and *inclusion* dimensions. A context, then, expresses the specific specializations of the shared model that define a given local model (and hence a local model is described by the combination of the shared model and a particular context.) In the airfare example, for instance, the meaning of *airfare* objects is made explicit by local sources when they specify the currency used (e.g. USD); and declare whether the coverage is one-way or round-trip and what is included in the airfare (e.g. tax and shipping).

In the rest of this paper, we first review the ECOIN framework. We then elaborate on this approach in more detail by using a practical example from the air-travel domain and continue with a brief overview of the related work which is contrasted with the ECOIN approach. Finally, we discuss the benefits of our approach in reducing the amount of work needed to (a) construct a shared model, (b) describe local models, and (c) express mappings between contexts.

[1] ECOIN stands for **E**xtended **Co**ntext **In**terchange (see [Firat, 2003]).

The ECOIN Framework

The ECOIN *framework* is a generic logic-based data model that provides a template for the integration of heterogeneous data sources. This template is defined as follows:

Definition: (ECOIN Framework)
An ECOIN framework is a quadruple **(O, S, C, M)** where each component is a set of logical predicates. **O** corresponds to *ontology* that includes both the *domain and context model*; **S** corresponds to *source declarations*; **C** corresponds to *context (instances)*; and **M** corresponds to *mappings* (*conversion function network*) defined between contexts.

In this framework, sources **(S)** and contexts **(C)** are described with respect to the ontology **(O).** Mappings **(M)** are structured according to the context model to enable translation between different contexts. Below each component is described in detail.

Ontology

Ontology in ECOIN includes both the domain and context model. As in other data integration frameworks, an ECOIN domain model is used to define a common type system for the application domain (e.g., financial analysis, travel

information) corresponding to the data sources that are to be integrated. Like many other conceptual models, an ECOIN domain model consists of a collection of (object) *types,* which may be related in a subtype hierarchy. Types have *attributes* to represent both the individual properties of objects and relationships between objects (both things and their properties are uniformly represented as objects).

The types in an ECOIN domain model are *semantic types*, in that they represent the *generic* semantics of the concepts used in the various data sources. A semantic type is impartial to the exact representation or meaning of its instances in specific contexts and encapsulates all. The various specializations of these concepts used by different sources or receivers are described using a special kind of property called a *modifier*. The modifiers in an ontology are chosen to explicitly describe the contextual

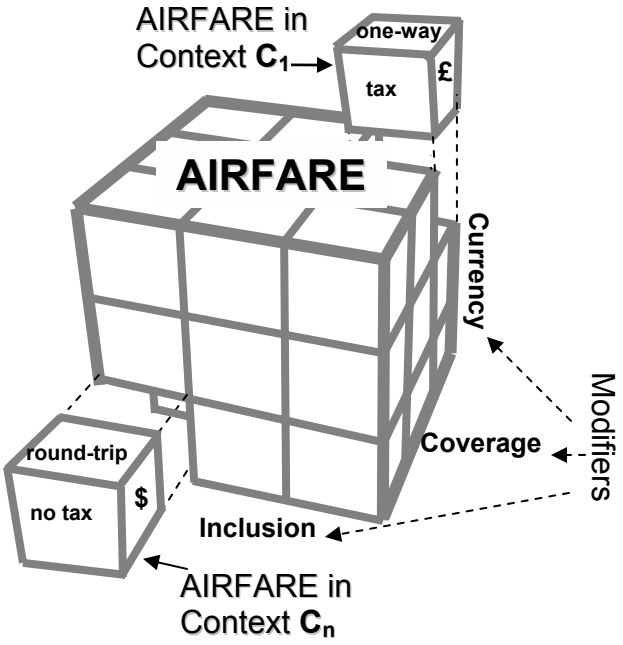


Figure 1 Multi-dimensional modification of the ontological term airfare

specializations of the generic types used by the sources and receivers. For example, in Figure 1 the generic ontological term *airfare* represented by the large cube can be specialized along three modification dimensions of *{Coverage, Currency, Inclusion}*. Different values of these modifiers identify the different component cubes of the overall airfare cube.

The *modifiers* in an ontology collectively define its *context model*; and the collection of *modifier objects* that describe the specializations that can be made by a given source or receiver defines its *context*. Context declarations are source independent, thus multiple sources or receivers may use the same context (use the same specializations for various values), but often different sources use different contexts.

Modifiers themselves are semantic types, thus can be subject to specialization (e.g. how do you represent currency? USD vs. $.) This can be handled via defining modifiers of modifiers. In Figure 2, this situation is illustrated by a *CurrencyFormat* modifier for the *Currency* modifier.

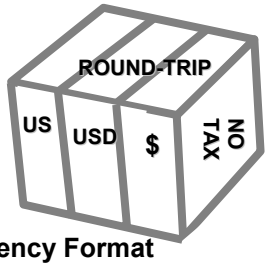


Figure 2 Modifiers themselves can be modified

For objects without modifiers, the context model implies a *current* existence of a common representation and meaning across the sources and receivers. If this assumption changes at a later time, new modifiers can be introduced, further slicing and dicing the generic concepts.

In Figure 3, we illustrate a simplified ontology for the air travel domain. The domain and context model corresponding to the figure are represented in ECOIN with the following logical predicates: (The omitted predicates are indicated with three dots.)

Ontology

Domain Model

Types:
semanticType(country). semanticType(idType)….
semanticType(coverageType).

Type hierarchy:
isa(airfare, moneyAmt). isa(tax, moneyAmt).

Attributes/Relationships:
cxnCountry(ticket,country),….,hasID(ticket, idType).

Context Model

Modifiers:
lformat(airport, Context, airportFormat).
dformat(date, Context, dateFormat).
inclusion(airfare, Context, inclusionType).
coverage(airfare, Context, coverageType).
currency(moneyAmt, Context, currencyType).

The variable 'Context' in the Context Model signifies that a modifier is defined with respect to a given context, thus may acquire different values in different contexts.

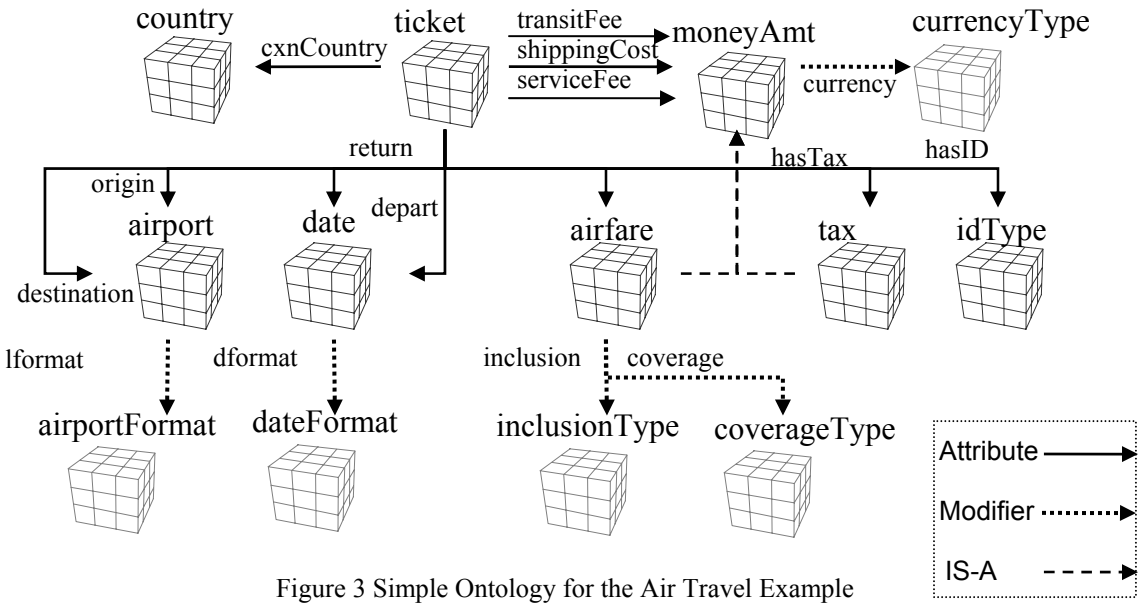


Figure 3 Simple Ontology for the Air Travel Example

Sources

Sources in the ECOIN framework are uniformly treated as relational sources (i.e., as having relational schemas). Many non-relational sources, such as HTML and XML web sites and web services, can be transformed into relational sources via wrappers [Firat at al. 2000]. A wrapped web source, for example, can be represented in logical predicates as (refer to Figure 5):

cheaptickets(Id, Airline, Price, Tax, DepDate, ArrDate, DepCity, CxnCountry, ArrCity)

In the ECOIN framework, these are called *primitive relations*, because these sources are not yet tied to an ontology. These primitive relations are elevated into *semantic relations* by annotating the semantic type and context of each primitive relation.
The semantic relation cheaptickets' can then be expressed as follows[2]:

cheaptickets'(Id', Airline', Price', Tax', Depdate', ArrDate', DepCity', CxnCountry',ArrCity') ←
Id'=object(idType, Id, c_ct, cheaptickets(Id, Airline, Price, Tax,Depdate,ArrDate, DepCity, CxnCountry, ArrCity)), …,ArrCity' = object(…).

With this elevation each column of the cheaptickets relation is indirectly tied to the air travel ontology. For the Id column, for instance, this is accomplished by associating the value Id with the idType semantic type in the cheaptickets context c_ct. Id' in the above declaration is the semantic object corresponding to the primitive object Id from the cheaptickets relation.

In addition, the attribute relationships defined by the ontology are instantiated as part of source declarations. For example, the cxnCountry relationship would be declared for this source as follows:

cxnCountry(T',C')←cheaptickets'(T',_,_,_,_,_,_,C',_).[3]

This declaration says that the cxnCountry of a semantic object T' is another semantic object C', both of which can be obtained from the semantic relation cheaptickets'. This is also known as the Global As View (GAV) approach of relating sources and the global model.

Context (Instances)

For *sources*, contexts define the specializations used for the underlying data values; and for *receivers* contexts describe the specializations assumed in viewing the data values. These specializations may be about the representation of data (e.g. European vs. American style date formats) or nuances in meaning (e.g. nominal vs. bottom-line prices).

To define a source or receiver context, modifier assignments need to be made. For example, the context labeled as c_ct can be described with the following predicates :

currency(Airfare', c_ct, Currency') ←
transitFee(Ticket',Airfare'),
cxnCountry(Ticket',Country') ,
countryCurrency(Country', Currency').[4]

[2] Notation: We add a single quote ' to semantic objects/relations to distinguish them from primitive ones.

[3] Underscores, as in Prolog, are used to designate any value.

[4] Here, countryCurrency is a semantic relation that relates countries and currencies.

currency(MoneyAmt', c_ct, Currency') ← Currency' = object(currencyType, "USD", c_ct, constant("USD")).

inclusion(Airfare', c_ct, Inclusion') ← Inclusion' = object(inclusionType, "nominal", c_ct, constant("nominal")).

coverage(Airfare', c_ct, Coverage') ← Coverage' = object (coverageType, "oneway", c_ ct, constant("oneway")).

lformat(Airport', c_ct, LFormat')← LFormat' = object(airportFormat, "airport", c_ct, constant("airport")).

dformat(Date', c_ct, DFormat') ← DFormat' = object (dateFormat, "American", c_ct, constant("American")).

These modifier declarations, which use attribute declarations, semantic relations, and some other constructs, explicitly specify which view of the ontology is adopted by the cheaptickets source. Accordingly, the ontology corresponding to the cheaptickets source treats *airfare* as *the one-way nominal price of a ticket in US dollars*. Currency in the cheaptickets context is US dollars except for transitFees which are given in the currency of the transit country. The arrival and departure locations are expressed as airport codes, and date is given using the *American style*.

Mappings

Mappings in ECOIN ensure that a view of the ontology adopted in a context is appropriately mapped to a corresponding ontological view in another context. This is accomplished by defining a conversion function network for each ontological term. Conversion functions are *atomically* defined for each modifier dimension as illustrated in Figure 4.

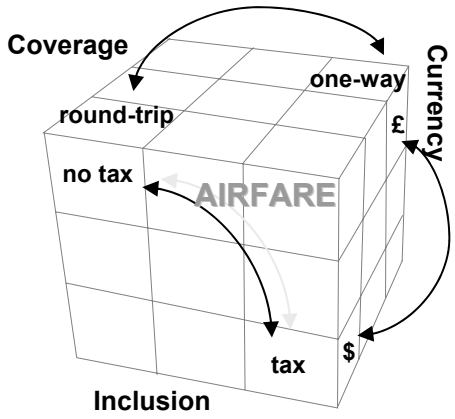


Figure 4 Organization of Conversion Functions for the Ontological term Airfare

As an example, the conversion function for the currency modifier dimension is encoded declaratively in terms of logical predicates as follows:

$f_{currency}$(X, VS, SC, VCurrencyS, VCurrencyT, TC, VT) ←
  value(Today, SC, VToday), system_date(VToday),
  value(CurrencyS,SC,VCurrencyS),
  value(CurrencyT,SC,VCurrencyT),
  currencyrates'(CurrencyS,CurrencyT, Today, Rate),
  value(Rate, SC, VRate), mul(VS, VRate, VT).

For semantic airfare objects, this function uses the modifier value VCurrencyS in source context SC, and modifier value VCurrencyT in target context TC to translate the source value VS of semantic object X to value VT in target context. The value(A,C,B) predicate used above is read as "the value of semantic object A in context C is B". The function is also using another semantic relation currencyrates'; a system function system_date(VToday) and an arithmetic predicate mul to express multiplication.

As in the currency conversion function example above, conversion functions can sometimes be defined parametrically, thus may cover all of the modifier value pairs with a single function. When this can not be done, conversion functions can be defined as a network to minimize the number of declarations, leaving the tasks of combining, inverting, and simplifying tasks to the mediator. Furthermore, most conversion functions are orthogonal, i.e. they can be applied in any order. When they are not orthogonal, priorities are used to determine the order they are to be executed. The details of conversion function network organization can be found in [Firat et al. 2005].

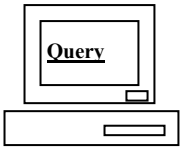
Practical Application

Consider the simplified scenario shown in Figure 5 having a single source *cheaptickets* and a single receiver (user) with conflicting assumptions. (This scenario, including the technical details of query mediation, is discussed more thoroughly in [Firat et al. 2005].) Surprisingly, even the semantic differences between a single source and a receiver provide enough complexity to highlight some of the interesting issues. Under this scenario, the user is an international student looking for a round trip airfare from Boston to Istanbul, with departure on June 1st and return on August 1st 2004. He wants to obtain the price and airline information for his trip and formulates the following SQL query Q1 using column names from the source:

Q1: SELECT Airline, Price
FROM CheapTickets

Context of User

* Fares are expected to be bottom-line price
(round trip, includes taxes, ticket shipment, and transit fees)
* Date is expressed in European style (dd/mm/yy)
* Departure and Destination locations are expressed as city names
* Currency is US $
* Today's date: 01/05/04




```
SELECT Airline, Price
FROM cheaptickets
WHERE DepDate = "01/06/04" and
ArrDate=  "01/08/04" and DepCity= "Boston"
and ArrCity= "Istanbul";
```

Context of cheaptickets

* All fares are for each way of travel and do not include fees and taxes.
* Date is expressed in American style (mm/dd/yy)
* Departure and Destination locations are expressed as three letter airport codes
* Currency is US dollars except for transit fees, which are in the currency of the country that issues the fee.
* Direct air transit fee of £27 is applied if the plane has a connecting flight from United Kingdom
* Ticket shipping cost is $20
* Service fee of $5 is charged

cheaptickets

ID (I)	Airline (A)	Price (P)	Tax (T)	DepDate (DD)	ArrDate (AD)	DepCity (DC)	CxnCountry (CC)	ArrCity (AC)
1	British Airways	495	75	06/01/04	08/01/04	BOS	United Kingdom	IST
2	Lufthansa	525	79	06/01/04	08/01/04	BOS	Germany	IST
…	…	…	…	…	…	…	…	…

Context of Ancillary Sources

Date is expressed in American style

currencyrates

FromCur	ToCur	ExchangeRate	Date
£	$	1.75	05/10/04
…	…	…	…

cityairport

City	Airport
Boston	BOS
Istanbul	IST

Figure 5 Airfare Example Scenario

WHERE DepDate = "01/**06**/04" and ArrDate = "01/**08**/04" and DepCity = "**Boston**" and ArrCity = "**Istanbul**";

As a result of the contextual differences illustrated in Figure 5, without any mediation the user's query would return an empty answer, because cheaptickets has city codes instead of city names; and dates are in American format (refer to sample data). Even if these specific differences were dealt with, for example by writing a new query Q2 with changed city codes and date formats (which itself might be a significant challenge for the user, especially if unfamiliar with the details of each of the multiple sources involved):

```
Q2:  SELECT Airline, Price
FROM CheapTickets
WHERE DepDate = "06/01/04" and ArrDate = "08/01/04"
and  DepCity = "BOS" and ArrCity = "IST";
```

the results returned would be:

Airline	Price
British Airways	495
Lufthansa	525

which is not the correct bottom line price the user expects. Given these results, the user may incorrectly think that British Airways is the cheaper option. If the original query Q1 were submitted to the ECOIN system, however, the semantic conflicts between the sources and the receiver would be automatically determined and reconciled, and Q1 would be rewritten into the following mediated query:

```
MQ1:SELECT Airline, 2*(Price+Tax+27*exchangeRate)+25
FROM cheaptickets, currencyrates,
(select Airport from cityairport where city="Boston") dCode,
(select Airport from cityairport where city="Istanbul") aCode
WHERE DepDate = "06/01/04" and ArrDate="08/01/04" and
DepCity= dCode.Airport and ArrCity=aCode.Airport
and CxnCountry= "United Kingdom" and fromCur= "GBR"
and toCur= "USD"  and Date= "05/10/04";
UNION
SELECT Airline, 2 * (Price+Tax) +25
FROM cheaptickets,
(select Airport from cityAirport where city="Boston") dCode,
(select Airport from cityAirport where city="Istanbul") aCode
```

WHERE DepDate = “06/01/04” and ArrDate=”08/01/04” and DepCity= dCode.Airport and ArrCity=aCode.Airport and CxnCountry <> “United Kingdom”;

In the mediated query MQ1, in addition to representational conflicts such as format differences in date and city codes, semantic conflicts in the interpretation of airfare (price) are also resolved. Mediating such semantic conflicts involves creating a conflict table by comparing the modifiers involved in the query; identifying which mappings to use from the conversion function network to resolve the conflicts; and applying symbolic equation solving techniques to a number of equational relations for inversion, composition and simplification under the Abductive Constraint Logic Programming framework [Firat et. al. 2005].

The ECOIN system further processes this mediated query by an optimizer to produce an efficient plan, and executes it by a query processor, which submits subqueries to individual sources that can optimally execute the subqueries and perform the data transformations. The final results reported by the system below now allow the user to make the right choice and choose Lufthansa over British Airways:

Airline	Price
British Airways	1260
Lufthansa	1233

As this practical application shows, despite sharing the same ontology, the users and sources are not locked into a single integrated view. Multiple integrated views can co-exist with a well defined context model coupled with the ontology.

Related Work and Discussion

One of the fundamental issues of information integration is achieving interoperability between multiple local models. There have been various approaches proposed in the past. Although there are some similarities, ECOIN has a number of important distinct differences and advantages.

In database integration, for instance, local models are in the form of database schemas and achieving interoperability among multiple schemas constitutes the fundamental problem. The traditional centralized solution maps local models (schemas) to a shared standard ontology (global schema) to eliminate representational and semantic disparities. This approach has been criticized for lack of scalability and difficulty of maintenance over time. Furthermore, it is seen as overly restrictive and inflexible in trying to reconcile local models that suit different needs in a single shared model [Bouquet and Serafini 04].

Modularized versions of the traditional centralized approach with the explicit use of “contexts” appears in [McCarthy and Buvac 97], and in CYC [Lenat et al., 1990; Guha, 1991]. In these approaches, axioms and statements are true only in a context. This is expressed by a *modality*[5] called ist(c,p)[6]. For example,

c_0: ist(context-of(“Sherlock Holmes stories”), “Holmes is a detective”).

means that the statement “Holmes is a detective” is true in the context of Sherlock Holmes stories. The preceding c_0 denotes that this statement is asserted in an outer context, thus points out to the nested composition of context dependent statements. Formulas between contexts can be related together with the use of lifting axioms. In [McCarthy and Buvac 97], an example of integrating Navy and General Electric (GE) databases, which differ on the definition of engine prices, is given. In the Navy database price includes assortment of spare parts and warranty, whereas in GE price is the plain engine price. Contexts defined in this example are c_{GE}, c_{navy} corresponding to the GE and navy databases and c_{ps}, the problem solving context. The details of this example are shown in Table 1 (i.e. the query posed in the problem solving context, the existing facts expressed in their own context, and lifting axioms that define translations between different contexts).

Query
c_{ps}: ist(c_{navy}, price(FX-22-engine, $3611K))
Facts
ist(c_{GE}, price(FX-22-engine, $3600K)).
ist(c_{GE}, price(FX-22-engine-fan-blades, $5K)).
ist(c_{GE}, price(FX-22-engine-two-year-warranty, $6K)).
ist(c_{navy},spares(FX-22-engine,FX-22-engine-fan-blades)).
ist(c_{navy}, warranty(FX-22-engine,FX-22-engine-two-year-warranty)).
Lifting axioms
value[7](c_{GE}, price(x)) = GE-price(x)
value(c_{navy}, price(x)) = GE-price(x) + GE-price(spares($C_{Navy,}$ x)) + GE-price(warranty(C_{Navy}, x)).

Table 1 Navy and General Electric Integration Example

An opposite approach, called “compose and conquer” [Bouquet et. al. 01], is based on the premise that the existence of a global ontology is not viable in open settings such as the envisioned Semantic Web. In the proposed solution, relations between local models are established on a peer-to-peer basis, as a collection of constraints on what can (or cannot) be true in a local model given that there is some relation with what holds in another local model. This approach has been used in [Ghidini and Serafini 98, 00], in integrating information systems. They provide an example

[5] The classification of propositions on the basis of whether they assert or deny the possibility, impossibility, contingency, or necessity of their content.
[6] Read as “p is true in context c”
[7] value(c,t) is a function which returns the value of term t in context c

that integrates the databases of four fruit sellers with different contexts. Conflict resolution between contexts is done pair wise for each database, since they do not subscribe to a common global theory. In the example, one of the sellers (1) provides fruit prices without including taxes, the other denoted as the mediator (m) considers prices with taxes (7% percent). This conflict is resolved by defining a view constraint as following:

1: has-price(x,y) $\rightarrow$ m:$\exists$y′ has-price(x,y′) $\wedge$ y′ =y +(0.07*y)

This view constraint establishes the link between differing price definitions of source 1 and mediator m.

While this approach offers important benefits, especially in providing a viable architecture for open settings, the lack of a shared model creates a number of serious problems. Even finding a way to query disparate data sources, connected on a peer-to-peer basis, becomes a non-trivial task. Furthermore, the coordination of establishing relationships between local models on a peer-to-peer basis is problematic. ECOIN provides a much simpler solution.

Contextual Coupling of Ontology and Local Models in ECOIN

The ECOIN strategy of relating local models favors the use of ontologies to relate local models, albeit in a much more flexible way than the traditional centralized approaches. It may be too early to predict how the Semantic Web will ultimately evolve, but it is perceivable that similar local models will be linked via ontologies, which in turn may be treated as local models and linked via higher level ontologies thus achieving gradual semantic interoperability. Given such a possibility, the ECOIN approach introduces a contextual coupling of ontology and local models.

Our approach may seem similar to the efforts discussed in [McCarthy and Buvac 97], [Lenat et al., 1990; Guha, 1991], [Kashyap and Sheth 96], [Bouquet et al. 2004] at the surface level, but there are important differences. While we like the explicit treatment of contexts in these efforts; and share their concern for sustaining an infrastructure for data integration, our realization of these differ significantly. First, the ontology in ECOIN only defines *generic* terms without specifying their exact semantics, which has no equivalent in the aforementioned approaches. Second, *lifting axioms* [Guha 1991] in our case operate at a finer level of granularity: rather than writing axioms which map “statements” present in a data source to a common knowledge base, they are used for translating “properties" of individual “data objects" and organized as a conversion function network between contexts. These differences account largely for the scalability and extensibility of our approach.

Compared with the Context-OWL (C-OWL) approach discussed in [Bouquet et al. 2004], our effort is more focused on query mediation than trying to come up with a general theory of contextual reasoning. Furthermore, the contextual mappings in our case go beyond the rather limited set of mappings that exist in C-OWL (i.e. equivalent, onto (superset), into (subset), compatible, and incompatible). The limited expressiveness of the C-OWL language fails to address the contextual differences such as those possible with ECOIN.

The *description logic* based context representation as contextual coordinates in [Kashyap and Sheth 96] has compelling similarities with our approach. While the desire to dynamically express the context of data is paramount in both approaches, there are also important conceptualization differences. While the *contextual coordinates* denote aspects of the context in [Kashyap and Sheth 96], *modifiers* denote aspects of the ontological terms in ECOIN. Our conceptualization results in a simpler context model, which works very well for query mediation by allowing us to organize, compose, invert and simplify conversion functions that maps between different contexts.

Compared with the “compose and conquer” approach, the ECOIN approach is similar in its desire to perform peer to peer mappings (although mappings need not be defined between every peer). Unlike “compose and conquer”, however, ECOIN assumes the existence of an ontology to tie the sources, but this ontology does not act like a “global schema”. The ontology acknowledges the *minimal* agreements between the local models, and defines a well-defined (yet extensible) context model to facilitate the reconciliation of possible conflicts between local models. With these differences in mind, the benefits of the ECOIN approach can be summarized as follows.

First, ontology developers do not have to standardize the exact meaning and representation of ontological terms; but only need to agree on generic identities without exposing the specific details. A major advantage of this approach is that ontology developers frequently find it straightforward (if not necessarily "easy") to agree on the generic concepts; it is getting all the precise details worked out that creates a lot of the work. Moreover, it's often the case that differences in these precise details are only discovered later (sometimes even after the system is in operation). The ECOIN approach enables these details to be factored out, reducing the amount of work needed to introduce these details all at once.

Second, allowing the same ontological term to assume nuances of meaning and varying representations in local contexts saves the ontology from being cluttered with non-essential terms such as *airfareIn$*, *airfareWithTax*, *airfareRoundTrip*, etc. In ECOIN, only the essential term *airfare* with its modification dimensions belongs to the ontology, and takes its specialized meanings in local contexts with corresponding modifier values.

Third, because the ontology is impartial to the precise semantics defined in the various contexts, mappings are

not defined between the sources and the ontology as it is done in most current approaches to information integration. Instead, mappings are structured with respect to a context model and defined for each modification dimension as a *conversion function network*. This modularization of mappings allows a mediator to create custom point to point translations between contexts by selecting or composing appropriate mappings from the conversion function network. These capabilities of ECOIN have been demonstrated in an example application requiring the integration of counter-terrorism intelligence information, where ECOIN was able to generate over 22,000 conversion programs to enable semantic integration amongst 150 data sources and receivers using just six parameterized conversion rules [Zhu and Madnick 2004].

Conclusion

The ECOIN semantic information integration framework couples ontologies and contexts in a unique way for the semantic integration of disparate data sources. The approach presupposes the existence of an ontology, but unlike traditional approaches this ontology does not provide a rigid specification of the meanings and representations of its terms. In this novel conceptualization, ontological terms are modified according to a context model, thus correspond to multiple integrated views according to contextual choices.

We have implemented these ideas in a prototype [Firat 2003] using the Eclipse Prolog engine [Cheadle et al. 2003] and procedural programming languages. This prototype provides mediated access to traditional databases, as well as semi-structured web sites, and web services, creates and maintains metadata that are used in ECOIN through graphical interfaces, and supports merging multiple applications.

We believe that semantic information integration should have the dual purpose of: (1) *reconciling semantic heterogeneity across information sources*; and (2) *supporting semantic heterogeneity across information receivers*. The ECOIN approach achieves this objective by providing an integration framework that requires minimal agreement on a generic ontology, and allowing the local models to modify the ontology to fit their context.

References

Bouquet P., and Serafini, L. (2004). *Meaning Coordination and Negotiation*, Working Notes of the ISWC Workshop on Meaning Coordination and Negotiation, 3rd International Semantic Web Conference, Hiroshima, Japan.

Bouquet P., Ghidini C., Giunchiglia F., and Blanzieri E. (2001). *Theories And Uses Of Context In Knowledge Representation And Reasoning*, Technical Report # 0110-28, Istituto Trentino di Cultura.

Bouquet P., and Serafini L. (2003). *On the Difference between Bridge Rules and Lifting Axioms*. Modeling and Using Context, 4th International and Interdisciplinary Conference, CONTEXT: 80-93

Bouquet P., Giunchiglia F., Harmelen, F., Serafini, L., and Stuckenschmidt, H.(2004). *Contextualizing Ontologies*, Journal of Web Semantics , vol. 26, 2004: 1-19.

Cheadle, A. M., Harvey, W., Sadler, A.J., Schimpf, J., Shen, K., and Wallace M. G. (2003). *ECLiPSe: An Introduction* by. IC-Parc, Imperial College London, Technical Report IC-Parc-03-1.

Firat, A. Madnick S., Siegel M., Grosof, B., and Manola, F. (2005) *Reconciling Semantic Heterogeneity with Symbolic Equation Solving Techniques*, manuscript submitted for publication.

Firat, A. (2003). *Information Integration using Contextual Knowledge and Ontology Merging*, Ph.D. Thesis, Massachusetts Institute of Technology.

Firat, A., Madnick, S., and Siegel, M. (2000). *The Caméléon Web Wrapper Engine*, In Proceedings of the VLDB2000 Workshop on Technologies for E-Services, 1-9.

Ghidini, C., and Giunchiglia, F. (2001). *Local Models Semantics, or Contextual Reasoning = Locality + Compatibility*. Artificial Intelligence. 127(2):221-259.

Ghidini, C., and Serafini, L. (1998). *Information Integration for Electronic Commerce*. In Agent Mediated Electronic Commerce. First International Workshop on Agent Mediated Electronic Trading, AMET-98, Volume 1571 of LNAI. Springer.

Guha R. V. (1991). Contexts: a formalization and some applications, MCC Tech Rep ACT-CYC42391.

Kashyap, V.; Sheth, A.P. (1996) *Semantic and Schematic Similarities between Database Objects: A Context-Based Approach*,VLDB Journal 5(4):276-304.

Lenat, D., R. V. Guha, K. Pittman, D. Pratt, and M. Shepherd. (1990). *Cyc: Towards programs with common sense*. Communications of the ACM 33(8).

McCarthy, John and Buvac, S, 1997. *Formalizing context (expanded notes)*. In: Aliseda, A., van Glabbeek, R. and Westerstrahl, D., Editors, 1997. Computing natural language, Center for the Study of Language and Information, Stanford, CA.

Zhu, H., and Madnick, S. 2004. *Context Interchange as a Scalable Solution to Interoperating Amongst Heterogeneous Dynamic Services*, Proceedings of the Third Workshop on eBusiness (Web2004)