CONTEXTUAL ALIGNMENT OF ONTOLOGIES FOR SEMANTIC INTEROPERABILITY

(WITS/ICIS)

Aykut Firat (a.firat@neu.edu)
Stuart E. Madnick (smadnick@mit.edu)
Benjamin Grosof (bgrosof@mit.edu)

Working Paper CISL# 2004-17

October 2004

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

CONTEXTUAL ALIGNMENT OF ONTOLOGIES FOR SEMANTIC INTEROPERABILITY

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

Stuart Madnick, Benjamin Grosof
Massachusetts Institute of Technology
Cambridge, MA USA
{smadnick,bgrosof}@mit.edu

Abstract

Virtually all of the existing approaches to ontology integration assume that each of the individual ontologies (and the integrated ontology) corresponds to a single set of semantics at a given time. We first claim that this single integrated view assumption is unnecessarily restrictive, and defend the view that ontologies can simultaneously accommodate multiple integrated views provided the accompaniment of *contexts*-- a set of axioms on the interpretation of data allowing local variations in representation and nuances in meaning, and a *conversion function network* between contexts to reconcile contextual differences. Then, we propose an ontology integration methodology based on the alignment of contexts and linking conversion function networks defined between contexts. The flexibility of our approach and methodology is illustrated with the alignment of air travel and car rental domains, an actual example from our prototype implementation.

1. INTRODUCTION

Ontologies are used as a common language in the integration of semantically heterogeneous information sources for specific domains. When there is a need for interoperability across domains, ontologies need to interoperate. Virtually all of the existing approaches to ontology interoperability assume that each of the individual ontologies corresponds to a single integrated view at a given time. We first claim that this assumption is unnecessarily restrictive; and promote the use of domain ontologies to simultaneously accommodate multiple integrated views provided the accompaniment of *contexts*--a set of axioms on the interpretation of data allowing local variations in representation and nuances in meaning--and *a conversion function network* between contexts to reconcile contextual differences. We use the ECOIN information integration framework [1] to exemplify the interplay of ontologies with contexts & conversion function networks in supporting multiple integrated views with an actual example from air travel domain.

Then, we consider multiple systems modeled this way (with the addition of an example from the car rental domain) and discuss an approach to achieve interoperability *between* them without locking users into a single predefined view. The flexibility of our approach and methodology is illustrated with the alignment of the air travel and car rental domains, an actual example from our prototype implementation.

2. SINGLE ONTOLOGY, MULTIPLE MEANINGS

We can think of ontological terms as identities acquiring their exact meanings when situated in a context. Information systems subscribing to the same ontology, then, may actually speak the same "language" in different ways (often for good reasons) depending on context. To support communication while allowing contextual variations, equivalences between meanings need to be found and established between various contexts. In Figure 1, we illustrate a simplified air travel ontology, together with contexts and a (partly shown) conversion function network defined between contexts.

Both *things* and their *properties* are represented as (*semantic) objects* in the simplified ontology; therefore no distinction is made in the representation of *intrinsic* and *mutual properties* through *attributes*[2]. Semantic objects can also have special attributes called *modifiers* to represent the conditions under which the values of objects are observed in source relations: these are shown with dashed borders.

Contexts are shown as modifier-value pairs in the figure, but they are actually axioms in the form of first-order statements which make (possibly dynamic) assignments to modifiers. Modifiers are introduced by the ontology designers; and the context axioms are generated by the owners of the data sources that

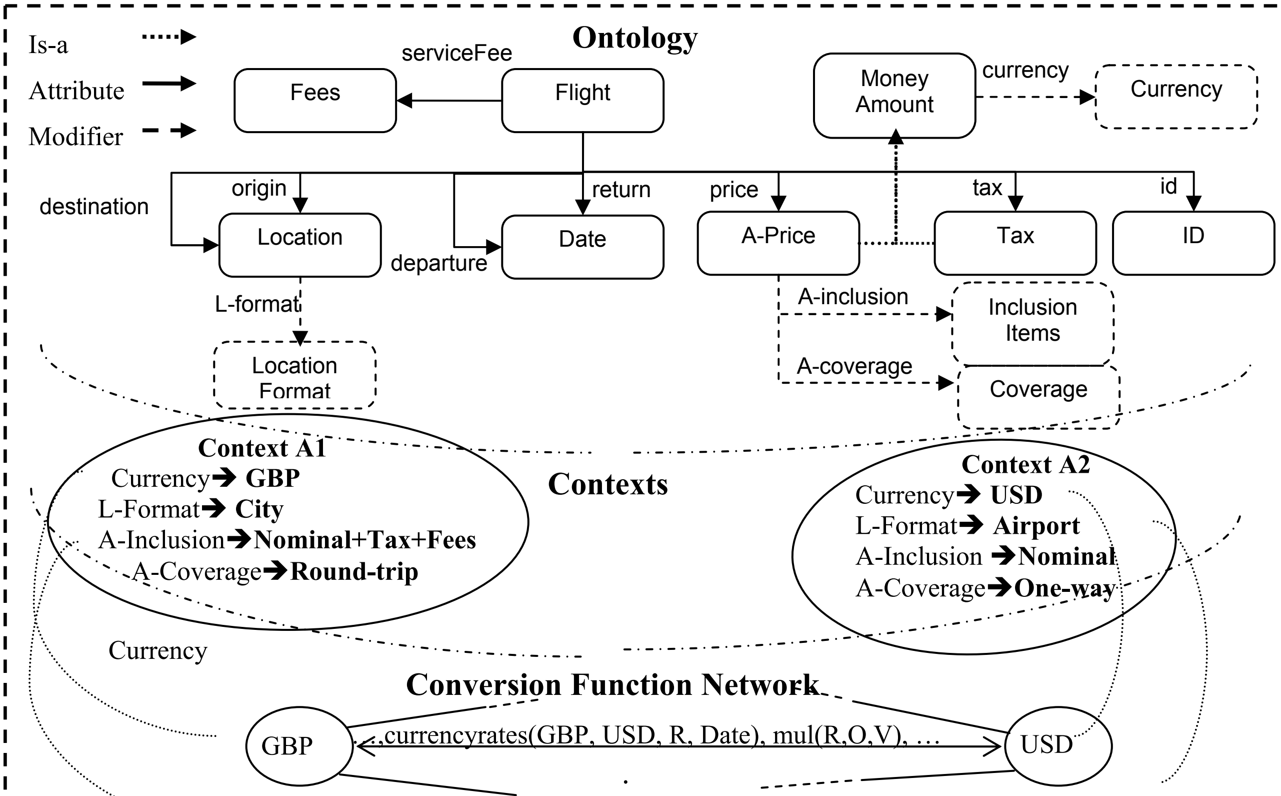


Figure 1. Ontology, Context & Conversion Functions for Air Travel (Simplified)

subscribe to a given ontology. With the use of modifiers, ontological terms can acquire local meanings: for example, in *context A1*, the A-Inclusion modifier value "Nominal+Tax+Fees" and A-Coverage modifier value "Round-trip" indicates that price is the bottom-line price; while in *context A2*, price corresponds to the "Nominal" "One-way" price. Modifiers can have modifiers of their own, thereby relativizing their meaning-- as only terms without modifiers are assumed to have a shared meaning across all contexts. For each modifier, conversion functions are organized as a network and encode necessary translations between different modifier values.

Consider now an integrated system built according to this ontology, context axioms, and conversion functions as shown in Figure 2. The user is in context **A1,** and issues query Q1 (see figure) against the cheaptickets data source which is in context **A2**. This query Q1 is issued in the user context A1, without being concerned with the diverse meanings of the ontological terms may take in the cheaptickets context. The *Context Mediator* (details in [1]) rewrites Q1 into the mediated query MQ1:

```
MQ1: SELECT Airline, (2* (Price+Tax) + 5) * eRate
FROM cheaptickets, currencyrates, (select Airport from cityairport where city= "Boston") cityairport1,
        (select Airport from cityairport where city= "Istanbul") cityairport2
WHERE DepDate = "06/01/04" and ArrDate="07/01/04" and   DepCity= cityairport1.Airport and ArrCity=
cityairport2.Airport and fromCur= "USD" and toCur= "GBP"  and Date= "05/10/04";
```

In the mediated query MQ1, in addition to city name vs. airport code conflicts, the conflict in the interpretation of price is resolved. The nominal price is converted into a bottom line price by first adding taxes; then multiplying by two as the prices reported by cheaptickets are for each way; and the addition of service fee. Finally, the price is converted from American dollars into British Pounds with the help of the ancillary sources. These are system services required to support multiple meanings within a single ontology. This mediated query is further processed by an Optimizer to produce an efficient plan, and executed by an Executioner which submits subqueries to individual sources, collects the results, and performs any necessary transformations. The final results obtained in the user context are shown to the right:

Airline	Price
British Airways	654
Lufthansa	674

Having illustrated very briefly, how a single ontology can allow the flexibility of multiple integrated views with the use of contexts and a conversion function network, we can now proceed to the next section to consider the multiple ontologies scenario.

User A in Context A1 **AIRFARE**

* Fares are expected to be bottom-line price (round trip, includes taxes and fees)
* Departure and Destination locations are expressed as city names
* Currency is GBP * Today's date: 05/01/04

Q1: SELECT Price FROM cheaptickets WHERE DepartureDate = "06/01/04" and ArrivalDate= "07/01/04" and DepartureCity= "Boston" and ArrivalCity= "Istanbul";

Query

Cheaptickets in Context A2

* All fares are for each way of travel and do not include fees and taxes. * Service fee of $5 is charged
* Departure and Destination locations are expressed as three letter airport codes * Currency is USD
* Lufthansa offers 10% discount if the airfare is bundled with National car rental

cheaptickets

ID (I)	Airline (A)	Price (P)	Tax (T)	DepDate (DD)	ArrDate (AD)	DepCity (DC)	ArrCity (AC)
1	British Airways	495	75	06/01/04	08/01/04	BOS	IST
2	Lufthansa	510	77	06/01/04	08/01/04	BOS	IST
…	…	…	…	…	…	…	…

Ancillary Sources

currencyrates

FromCur	ToCur	eRate	Date
GBP	USD	1.75	05/10/04
EUR	USD	1.25	05/10/04
…	…	…	…

cityairport

City	Airport
Boston	BOS
Istanbul	IST
…	…

Figure 2 Airfare Example Scenario

3. CONTEXTUAL ALIGNMENT OF MULTIPLE ONTOLOGIES

Consider now, the European car rental application based on the ontology shown in Figure 3. The figure

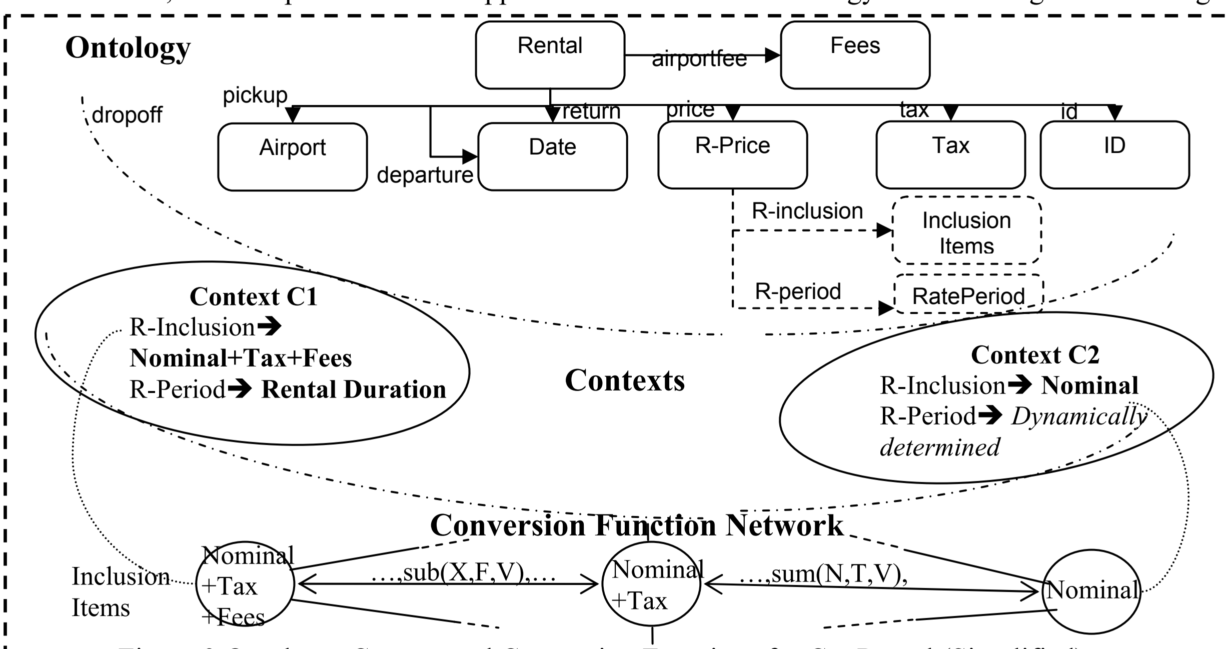


Figure 3 Ontology, Context and Conversion Functions for Car Rental (Simplified)

also contains two sample context definitions, and illustrates the conversion function network between contexts. In the car rental scenario, illustrated in Figure 4, user C in context **C1** poses a query Q2, which is mediated into MQ2 (similar to Q1→MQ1) :

```
MQ2: SELECT Price * 34.65
FROM cheaprentals, (select Airport from cityairport where city= “Istanbul”) cityairport
WHERE Class= “Economy” and PickDate = “02/06/04” and DropDate= “01/07/04” and Pickup=
cityairport.Airport and DropOff= cityairport.Airport;
```

In MQ2, the daily rates given by the sources are converted into the bottom line price requested by the user by multiplying the price by total rental days, the airport concession fee and sales tax ratios (30 * 1.1 * 1.05=34.65). The results are shown on the right. Note that in the car rental domain there is a shared understanding that the currency is Euros, and the dates are expressed in European styles; therefore modifiers were not used during the ontology design for date and the monetary amounts like price, tax, and fees; which emerges as an issue in the merging process. Consider now a third user as shown in Figure 6 who wants to query both domains together. With query Q3, the user wants to see bottom line prices in Euros including any bundling discounts (refer to Figure 2 & 4 to see the bundling discounts); expresses dates in American style, and locations as city names. The two systems are merged in Figure 5 to achieve this purpose.

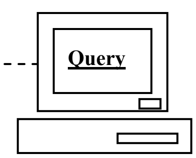
Airline	**Price**
Hertz	831
National	998

Several things need to be noted concerning the process of merging, although details will be skipped because of space limitations. First of all, our approach is a hybrid of ontology merging and alignment approaches [3]. Like ontology alignment approaches we use (articulation) axioms to align ontologies, and like ontology merging approaches we produce a new but virtual ontology out of two ontologies. This virtual ontology inherits all of the underlying ontology elements except when these terms are deemed to be equivalent--in which case only one term is upward inherited referring to both. All the axioms used in the merging become part of the new virtual ontology; new terms, relationships, modifiers, etc. can be added. A new *Price* type is added in the merged ontology, which is defined as the supertype of A-Price and R-Price, and the subtype of *MoneyAmount*. This new term is also assigned a new modifier *bundling* to denote whether the bundling discount included in the price or not in a given context. Before merging, the term *Date* in both ontologies did not have any modifiers, but now that there are multiple views of *Date* at the same time (European vs. American) a new modifier *d-format* is introduced in the merged ontology. For all of these new modifiers, context axioms are appended with new axioms to make value assignments.

User C in Context C1 **CAR RENTAL**

* Rentals are expected to be bottom-line price (includes taxes, and fees)
* Rates are for the rental duration

Q2: SELECT Price FROM cheaprentals
WHERE Class= “Economy” and
PickDate = “02/06/04” and
DropDate= “01/07/04” and
Pickup= “IST” and DropOff= “IST”;



Cheaprentals in Context C2

* Rentals do not include fees and taxes.
* Rates are daily
* National offers 10% discount if the car rental is bundled with a Lufthansa airfare

* Airport concession recovery fee %10
* Sales tax is 5%

ID(I)	**Company (A)**	**PickUp (PU)**	**DropOff (DO)**	**PickDate (PD)**	**DropDate (DD)**	**Price (P)**	**Class (C)**	**Rate Period (RP)**
1	Hertz	IST	IST	02/06/04	01/07/04	23.99	Economy	Daily
2	National	IST	IST	02/06/04	01/07/04	28.79	Economy	Daily

Figure 4 Car Rental Example Scenario

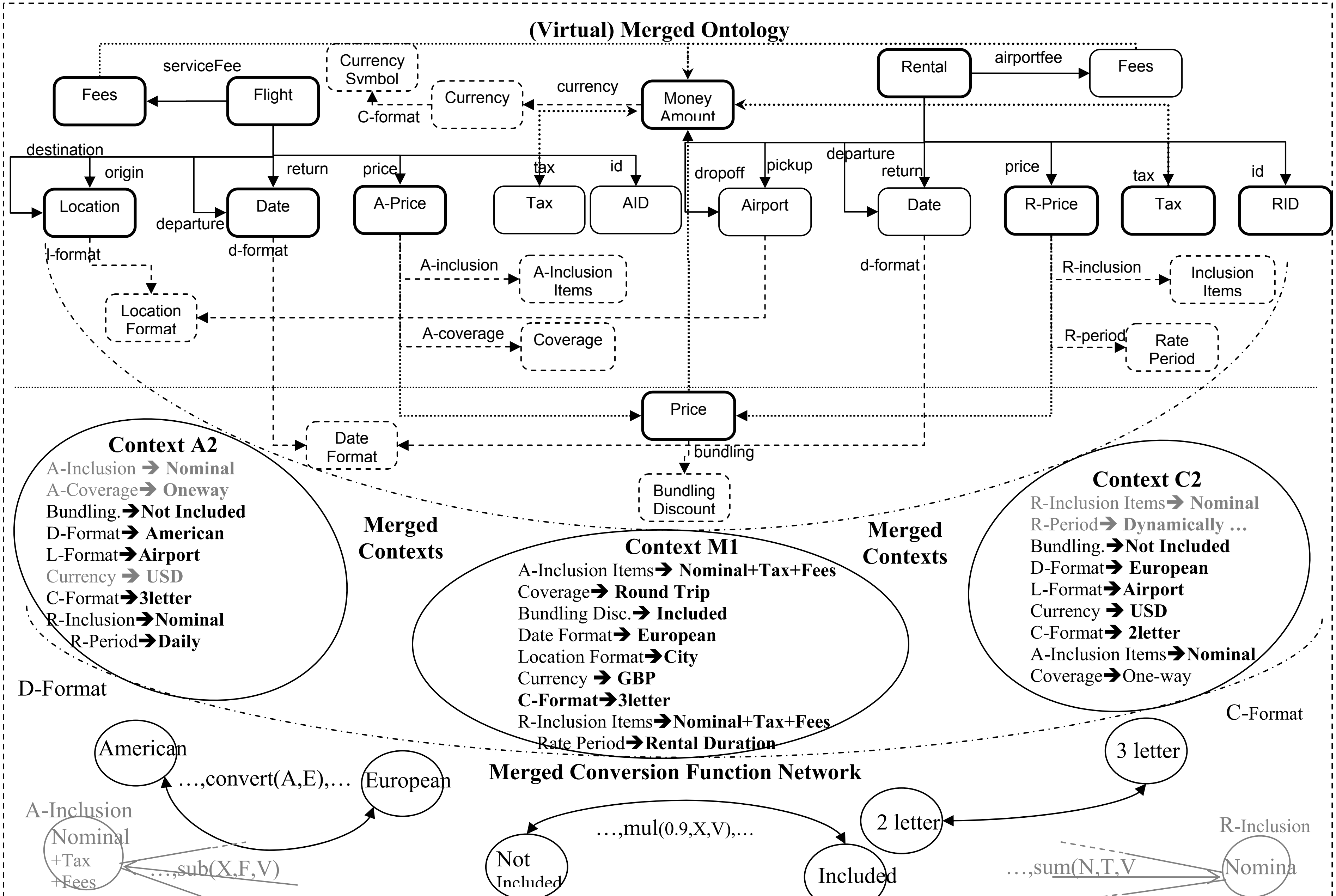


Figure 5. Merged Ontology, Contexts, and Conversion Function Networks

AIRFARE & CAR RENTAL

User Merger in Context M1

* Both Rentals and Fares are expected to be bottom-line & bundle price
* Date is expressed in American style
* Both Rental and flight locations are expressed as city names
* Currency is Euros

Q3: SELECT Airline, Company, t.Price + r.Price as total
FROM cheaptickets t, cheaprentals r
WHERE DepDate = "06/01/04" and ArrDate= "07/01/04" and DepCity= "Boston" and ArrCity= "Istanbul";
Pickup="Istanbul" and Dropoff="Istanbul" and PickDate="06/02/04" and DropDate="07/01/04";

Figure 6. Combined Airfare & Car Rental Example Scenario

The conversion function network likewise may need adjustments. In our example, three new functions were added to the network for the new modifiers bundling, d-format, and the c-format (a modifier for modifier currency). With this new merged ontology, Q3 can now be mediated into the following mediated query MQ3:

```
MQ3: SELECT "Lufthansa", "National", ((2 * (t.Price + Tax )+5) * eRate + r.Price  * 34.65) * 0.9 as total
FROM   cheaptickets t, currencyrates, cheaprentals r, (select Airport from cityairport where city= "Boston")
          cityairport1, (select Airport from cityairport where city= "Istanbul") cityairport2
WHERE DepDate = "06/01/04" and ArrDate="07/01/04" and   DepCity= cityairport1.Airport and
         ArrCity= cityairport2.Airport  and fromCur= "USD" and toCur= "EUR"  and Date= "05/10/04" and
         Airline="Lufthansa" and Company="National"and Class= "Economy" and PickDate = "02/06/04" and
         DropDate=  "01/07/04"  and Pickup= cityairport2.Airport and DropOff= cityairport2.Airport
UNION SELECT Airline, Company, ((2 * (t.Price + Tax )+5) * eRate + r.Price  * 34.65)  as total
FROM cheaptickets t, currencyrates, cheaprentals r, (select Airport from cityairport where city= "Boston")
          cityairport1, (select Airport from cityairport where city= "Istanbul") cityairport2
WHERE DepDate = "06/01/04" and ArrDate="07/01/04" and   DepCity= cityairport1.Airport and
         ArrCity= cityairport2.Airport and fromCur= "USD" and toCur= "EUR" and Date= "05/10/04" and
         (Airline<>"Lufthansa" or Company<>"National") and Class= "Economy" and PickDate = "02/06/04" and
         DropDate=  "01/07/04"  and Pickup= cityairport2.Airport and DropOff=cityairport2.Airport
```

Airline	Company	total
British Airways	Hertz	1747
British Airways	National	1913
Lufthansa	Hertz	1775
Lufthansa	National	1747

which gives the illusion of a single system, while reconciling all the conflicts, and taking care of emerging situations such as the bundling (note the multiplication by 0.9 in the first subquery to take care of the bundling situation); and returns the results in the user context M1. The user has a tie between 'British Airways & Hertz' and 'Lufthansa & National'.

4. CONCLUSION

In this paper, we first illustrated how a single ontology can accommodate multiple views with the accompaniment of contexts, and system services that can translate between different contexts using a conversion function network. Then we briefly described a new kind of ontology alignment process based on aligning context definitions between ontologies. The virtually merged application creates the illusion of a single system that can access sources across domains; accomplishes cross fertilization of contexts and conversion functions; and offers value added benefits beyond what the underlying applications can provide. The work reported here introduces a novel approach to ontology interoperability that does not lock the users into a single pre-defined view, but supports multiple integrated views across domains.

5. REFERENCES

[1] FIRAT, A. 2003. Information Integration Using Contextual Knowledge and Ontology Merging, Ph.D. Thesis, Massachusetts Institute of Technology.
[2] WAND, Y., STOREY, V., AND WEBER, R. 1999. An ontological analysis of the relationship construct in conceptual modeling. ACM Transactions on Database Systems 24, 494 -528.
[3] FRIDMAN NOY, N., MUSEN, M. A. 2000. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. AAAI/IAAI 2000: 450-455.