A Prototype for Metadata-based Integration of Internet Sources

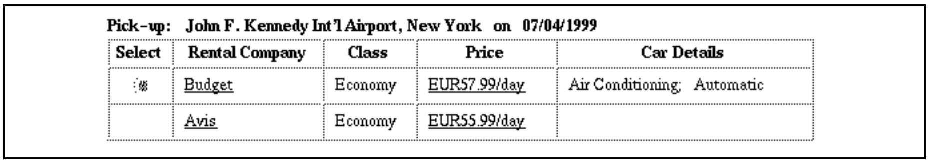
Christof Bornhövd and Alejandro P. Buchmann

DVS1, Department of Computer Science, Darmstadt University of Technology,
D-64283 Darmstadt, Germany
{bornhoev,buchmann}@dvs1.informatik.tu-darmstadt.de

Abstract. The combination of semistructured data from different sources on the Internet often fails because of syntactic and semantic differences. The resolution of these heterogeneities requires explicit context information in the form of metadata. We give a short overview of a representation model that is well suited for the explicit description of semistructured data, and show how it is used as the basis of a prototype for metadata-driven integration of heterogeneous data extracted from Web-pages.

1 Introduction

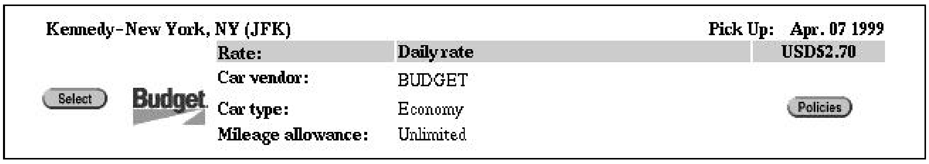
A wide variety of Information is available over the Internet. However, an integration of available data for further *automatic* processing is rarely possible because of a lack of uniform structure and meaning. Semantic context information is, at best, available locally to the institution managing the data source, and is lost when data is exchanged across institutional boundaries. Most sources provide data in semistructured form, and provide no explicitly specified schema with information about the structure and semantics of the data. The following HTML pages from different car reservation systems illustrate this.



Pick-up: John F. Kennedy Int'l Airport, New York on 07/04/1999

Select	Rental Company	Class	Price	Car Details
◉	Budget	Economy	EUR57.99/day	Air Conditioning; Automatic
	Avis	Economy	EUR55.99/day	

Fig. 1. Reservation System A



Kennedy-New York, NY (JFK) **Pick Up: Apr. 07 1999**

Rate:	**Daily rate**	**USD52.70**
Car vendor:	BUDGET	
Car type:	Economy	
Mileage allowance:	Unlimited	

Fig. 2. Reservation System B

To integrate and automatically process these data we must resolve the heterogeneities on the modeling level. This requires explicit knowledge about the structure of and the semantic assumptions underlying the data. For the representation and exchange of this context information [4] we use metadata.

The interpretation of the metadata requires the introduction of a shared vocabulary to reach a common understanding with regard to a given domain. Such a vocabulary is provided by a domain-specific ontology [2, 5]. An explicit description of the relationships between the data of a given source and the represented real world phenomenon is established by a reference to the underlying ontology, i.e., by mapping local representation types and terms to semantically corresponding ontology concepts, and adding context information that explicitly describes the underlying modeling assumptions.

In this paper we introduce a representation model that combines the explicit description of modeling assumptions underlying the available data, with concepts of a flexible, self-describing data model suitable for the representation of semistructured data. On the basis of this model we show how data from the Internet can be integrated, and prepared for further automatic processing.

2 MIX – A Metadata Based Integration Model

Our representation model is called ***M****etadata based* ***I****ntegration model for data* ***X****-change*, or MIX for short. MIX is a self-describing data model in the sense of [6], because information about the structure and semantics of the data is given as part of the available data itself, thus allowing a flexible association of context information in the form of metadata.

The model is based on the concept of a *semantic object*. A semantic object represents a data item together with its underlying *semantic context*, which consists of a variable set of meta-attributes (also represented as semantic objects) that explicitly describe implicit modeling assumptions. Our approach is based on the notion of context as proposed in [7, 4]. A more comprehensive notion of context can be found in [3].

In addition, each semantic object has a *concept label* associated with it that specifies the relationship between the object and the real world aspects it describes. These concept labels are taken from a commonly known ontology. Thus, the concept label and the semantic context of a semantic object help to describe the supposed meaning of the data.

The common ontology provides an extensible description basis to which data providers and consumers should refer. In specific application domains ontologies already do exist. However, in an imperfect real world, we must allow ontologies on consumer side that are tailored to specific needs and provide for extensibility of the model. Aspects for which no such description standards exist require new concepts to be specified by the corresponding data source, or by the consumer of the data. In our experience consumers are willing to invest into the interpretation of sources and extension of the ontologies if this results in future savings through automatic processing. By providing the means for adding metadata and extending the ontology on the receiver side, we believe that we can claim a reasonable combination of rigor and flexibility that makes the model applicable in real-life situations.

We distinguish between simple and complex semantic objects. *Simple semantic objects* represent atomic data items, such as simple number values or text strings. In contrast, *complex semantic objects* can be understood as heterogeneous collections of semantic objects, each of which describes exactly one attribute of the represented phenomenon. These subobjects are grouped under a corresponding ontology concept. The attributes given for a complex semantic object are divided in those used, similar to a set of key attributes in the relational model, to identify an object of a given concept, and additional attributes that might not be given for all objects of the concept. Attributes used for the identification provide the prerequisite for the identification of semantic objects that represent the same real world phenomenon.

For example, the data given by system A in Fig. 1 may be represented as the complex semantic object $SemObj_A$ of concept *CarOffer* given in Fig. 3. Each offer is identified by the attributes underlined. Additional properties, such as *Extras* are not required for the unique identification and might not be given for each *CarOffer*. In this way, complex semantic objects provide a flexible way to represent data with irregular structure, as it may be given by semistructured sources, or may result from the integration of different heterogeneous data sources.

```
SemObjA = < CarOffer, {
          < Company, "Budget" >,
          < Location, "J.F.Kennedy Int'l. Airport", {<LocationCode, "FullName">} >,
          < VehicleType, "Economy",                 {<TypeCode, "FullClassName">} >,
          < DailyRate, 57.99,                       {<Currency, "EUR">, <Scale, 1>} >,
          < PickUpDay, "07/04/1999",                {<DateFormat, "DD/MM/YYYY">} >,
          < Extras, "Air Conditioned" >,
          < Extras, "Automatic" >                                          } >   } >
SemObjB = < CarOffer, {
          < Company, "Budget" >,
          < Location, "JFK",                        {<LocationCode, "ThreeLetterCode">} >,
          < VehicleType, "Economy",                 {<TypeCode, "FullClassName">} >,
          < DailyRate, 52.70,                       {<Currency, "USD">, <Scale, 1>} >,
          < PickUpDay, "Apr. 07 1999",              {<DateFormat, "Mon. DD YYYY">} >,
          < FreeMiles, "Unlimited" >                                       } >   } >
```

Fig. 3. MIX Representation of Source A and B

For a semantically meaningful comparison of semantic objects we must take their underlying context into consideration. We use *conversion functions* by which semantic objects can be converted among different semantic contexts. These functions can be specified in the underlying ontology, or may be stored in an application-specific conversion library. Based on these mapping functions, semantic objects can be compared by converting them to a common semantic context, and comparing their underlying data values.

The MIX model is capable of representing data from different sources in a uniform way, and on a common interpretation basis. This supports the automatic processing of the data after their integration. Space limitations forced us to describe a short version of MIX. A detailed and formal presentation can be found in [1].

The data provided by the reservation systems introduced in Sec. 1 may be represented as shown in Fig. 3. By avoiding the need to agree on all attributes, both sources can agree on the same meaning for essential aspects of *CarOffer*, even though both sources make different semantic assumptions which result in different semantic contexts for their data.

The process of integrating data represented on the basis of MIX takes place in two steps. First, the semantic objects have to be converted to a common context, which can be specified by the application interested in the data, using appropriate conversion functions.

```
$ = {  < LocationCode, “ThreeLetterCode” >,
       < DateFormat, “DD.MM.YYYY” >,
       < TypeCode, “FullClassName” >,
       < Currency, “EUR” >,
       < Scale, 1 >                              }
```

Fig. 4. Common Representation Context

In the second step, semantic objects which represent the same real world object are identified by comparing their identifying attributes, and are fused into a common representation. For example, using context $ in Fig. 4, as the common context and conversion functions for the aspects specified in $ they may be classified as representing the same offer. Therefore, they are integrated into one semantic object by unification of their attribute sets. Properties described in both objects that are equivalent are represented only once as shown in Fig. 5, where $SemObj_A$ and $SemObj_B$ have been merged into $SemObj_{AB}$.

```
SemObjAB = < CarOffer, {
               < Company, “Budget” >,
               < Location, “JFK”,            {<LocationCode, “ThreeLetterCode”>} >,
               < VehicleType, “Economy”,     {<TypeCoding, “FullClassName”>} >,
               < DailyRate, 57.99,           {<Currency, “EUR”>, <Scale, 1>} >,
               < PickUpDay, “07.04.1999”,    {<DateFormat, “DD.MM.YYYY”>} >,
               < Extras, “Air Conditioned” >,
               < Extras, “Automatic” >,
               < FreeMiles, “Unlimited” >                                    } >   } >
```

Fig. 5. Unified MIX Representation

3 Metadata-based Prototype

To evaluate the MIX model we implemented an application independent Java framework that manages and exchanges semantic metadata. The implementation follows the mediator approach introduced in [9] and is shown in Fig. 6. The bottom layer of the architecture consists of autonomous data sources that may be structured or semistructured. The current prototype supports the mapping of relational databases and XML documents.

Wrappers map local data structures from the source to the concepts specified in the ontology. As stated earlier, the preferred solution is to use standardized ontologies, which exist for some domains, at the source. However, the framework provides the means to extend the ontologies both by the source or the consumer and even to provide an ontology entirely on the consumer side. This makes sense when a consumer interacts regularly with a fixed set of semistructured sources that do not always adhere to an ontology. In the current prototype, data wrappers are implemented as Java classes. These are registered with the federation manager and can be loaded dynamically to transfer data from the source. Wrappers that are not available at the federation manager can be loaded from designated wrapper servers. In this way the architecture allows a flexible management of a large number of data sources that may change frequently.

The federation manager keeps a metadata repository that includes the ontology information as well as information about the wrappers. When the federation manager receives a request the information in this repository is used to identify the appropriate wrapper classes, which are cached by the federation manager. The wrapper classes return semantic objects to the federation manager who converts them to the semantic context specified by the application. Semantically identical objects are fused as explained in Sec. 2. The federation manager then returns the unified semantic objects in the form of Java objects. The application views the data on the level of concepts from a domain-specific ontology without being aware of their local organization.

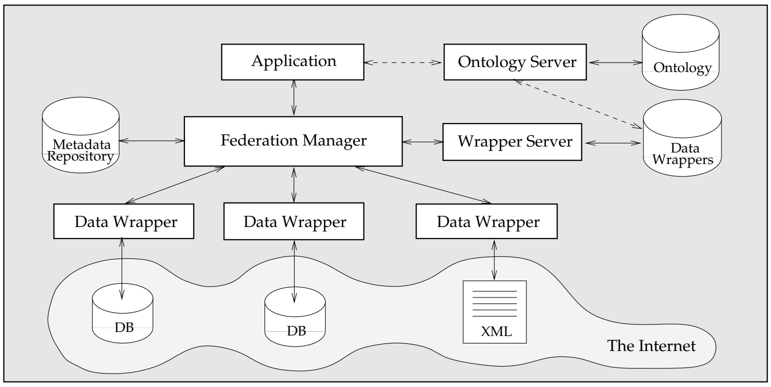


Fig. 6. The MIX Integration Framework

The ontology server stores and manages the domain-specific vocabulary underlying the federation. The concepts of the ontology are given as precompiled Java classes that can be downloaded by the application and the wrappers. To ensure the consistency of the ontology, new concepts are integrated by extending or specializing existing concepts in a predefined way to avoid ambiguous specifications or homonyms. The consistency of a given concept specification is ensured through the language constructs of Java.

To request data, an application builds SQL-like queries using the concepts of the ontology. The federation manager returns semantic objects, i.e. objects that are augmented with metadata. The application can then use these objects without further processing. The connection of an application to the corresponding ontology is established at compile time through the use of import statements which load the necessary concept classes into the local directory path.

As a sample application, the prototype includes an object browser that displays the semantic objects returned by the federation manager and allows the navigation of the object structure. In this way the attributes of complex semantic objects, as well as additional context information can be displayed interactively.

4 Related Research

Because of space limitations we mention only three approaches closely related to MIX. Our concept of a semantic object extends the data model discussed in [7] with regard to complex, maybe irregularly structured data objects. They assume a common vocabulary. MIX makes this vocabulary *explicit* and provides both for the *exchange* of vocabularies, and their *extensibility.*

XML [8] provides a flexible model for the representation and exchange of data similar to MIX. However, XML does not enforce a semantically meaningful data exchange per se, since different providers can define different tags to represent semantically similar information. Furthermore, XML does not support the integration of heterogeneous data. In contrast, MIX supports an explicit representation of semantic differences underlying the data, and specifies how this data may be converted to a common representation.

OEM [6], as well as MIX, is a self-describing data model for the representation of semistructured data. However, OEM objects are identified via system-wide object identifiers, and are based on source-specific labels. In contrast, MIX objects have certain attributes associated which support their identification based on their information content. Finally, OEM is tailored mainly to the representation of data with irregular structure. In addition to this, MIX also supports an explicit representation of the semantics underlying the data, and provides conversion functions to convert data between different contexts.

5 Conclusion

We presented a way of integrating data sources from the Internet which is not claimed to be generally applicable, but provides a fairly simple solution for many application domains. We use MIX in a project for the integration of travel data. The prototype of a Java-based implementation exists for MIX and the MIX integration environment. Current research is concerned with the extension of the representation of conversion functions, and with the extraction of MIX representations for a wider range of semistructured data.

References

1. Bornhövd, C.: *Semantic Metadata for the Integration of Web-based Data for Electronic Commerce*, Proc. Int'l. Worksh. on E-Commerce and Web-based Information Systems, Santa Clara, 1999
2. Gruber, T.: *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, International Journal of Human and Computer Studies, 43(5/6), 1995
3. Kashyap, V.; Sheth, A.: *Semantic and Schematic Similarities between Database Objects: A Context-based Approach*, VLDB Journal, 5(4) 1996
4. Madnick, S.E.: *From VLDB to VMLDB (Very MANY Large Data Bases): Dealing with Large-Scale Semantic Heterogeneity*, Proc. VLDB Conf., Zurich, Swizerland, 1995
5. Mena, E.; Kashyap, V.; Illarramendi, A.; Sheth, A.: *Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure*, Proc. Int'l. Conf. on Formal Ontology in Information Systems, Trento, Italy, 1998
6. Papakonstantinou, Y.; Garcia-Molina, H.; Widom, J.: *Object Exchange Across Heterogeneous Information Sources*, Proc. Int'l. Conf. on Data Engineering, Taipei, Taiwan, 1995
7. Sciore, E.; Siegel, M.; Rosenthal, A.: *Using Semantic Values to Facilitate Interoperability Among Heterogeneous Information Systems*, ACM TODS, 19(2), 1994
8. W3C: *Extensible Markup Language (XML) 1.0*, Feb. 10, 1998
9. Wiederhold, G.: *Mediation in Information Systems*, ACM Comp. Surv., 27(2), 1995