An application of the mediator approach to services over the Web

C. Reynaud & G.Giraldo
Université of Paris Sud, CNRS (L.R.I.) & INRIA (Futurs), Orsay, France

ABSTRACT: This document is about mediation systems integrating services over the Web. First, we give an overview of the mediator approach. In section 3 we focus on integration of services over the Web. We explain which problems arise and we give principles to make the mediator approach applicable in this context. The first principle is a solution to semantic heterogeneity. It exploits results in standardization relative to business transactions. The second principle concerns the scalability of the mediator approach. It emphasizes two points: de-coupling of the various components and strong mediation. In the last section, we present an architecture of a mediation system integrating services, designed in the setting of the PICSEL 2 project. We illustrate with an application of e-commerce in the tourism domain.

1 INTRODUCTION

The huge growth of new Internet technologies has led in the last years to the creation of a great amount of e-commerce applications. However finding relevant services on the World-Wide Web of today remains a tedious and long task for the end-user. In the setting of the PICSEL2 project, a collaboration with France Telecom R&D, our aim is to exploit results coming from standardization in e-commerce transactions in order to automate the construction of service mediation systems and to allow the mediator systems scaling up to the Web.

End-users have to be helped. Let us suppose, for example, that a user needs a service including a flight, a car rental and a lodging. Web booking services able to meet this requirement are numerous but heterogeneous, each one having been designed in a specific way. It is accessible according to particular modalities. It uses a particular vocabulary. Furthermore, one service may entirely fulfill the need of the user, whereas other services may partially fulfill it, and in that case a combination of several services is needed. Such a process is complex and this is the reason why end-users have to be provided with an interface giving them the illusion of a convivial, single, centralized and homogeneous system. Mediator systems are one of the solutions. In the recent years, considerable research has been done about information integration both for structured and semi-structured data sources. Several information integration systems have been implemented and they have proved to be manageable and useful. Given a query expressing the information looked for by a user, a mediator system takes in charge, in the place of the user, the tasks of finding the relevant information sources, interacting with each of these sources to get the useful data, and combining them to compute the appropriate final answer. In this paper, as service integration raises similar problems as sources integration does, we focus on using a mediator approach for integrating services.

Services that we want to integrate are Web services. As a consequence, the mediator approach has to be considered at the Web scale. More particularly, that means that the construction of mediators has to be automated as most as possible and that mediators must be designed as open systems able to integrate very easily new services.

We propose an approach to automate the building of an ontology, a major step in the construction of integration systems. The approach exploits results in standardization relative to business transactions. We propose also to automate the description of the functionalities of services which respect to the standard used to build the ontology, such descriptions being useful for the mediator to compute query plans. By providing an ontology based on standardized specifications, we don't make its design directly depend on the integrated services. That way, we obtain an open mediator system naturally ready to integrate new services whatever they are, since they deal with standardized messages that have been exploited to build the ontology.

The plan of the paper is the following. First, we give an overview of the mediator approach integrating heterogeneous information sources. Then section 3 is about problems specific to mediation systems integrating services over the Web. We explain which problems arise and we give principles to make the mediator approach applicable in this context. In the last section, we present an architecture of a mediation system integrating services, designed in the setting of the PICSEL 2 project. We illustrate with an application of e-commerce in the tourism domain.

2 THE MEDIATOR APPROACH

2.1 *General overview*

A mediator provides a uniform query interface for querying collections of pre-existing data sources that were created independently. These data sources are multiple and distributed. They may be syntactically or semantically heterogeneous while being related to a same domain (e.g., tourism, culture). A mediator system gives users or software agents the illusion that they interrogate a centralized and homogeneous information system.

A mediator system is composed of a single mediated schema, also called ontology. The ontology is a description of a domain of interest. Users pose queries in terms of that domain ontology rather than directly in terms of the source schemas. It is also composed of a set of source descriptions expressing how the content of each source available to the system is related to the domain of interest. Descriptions of source contents are expressed as views over the mediated schema. These source descriptions, also called mappings because they model the correspondence between the mediated schema and the schemas of the data sources, play a central role in the query answering process. A user query that is formulated on a domain ontology is translated by the data integration system into a set of queries against local schemas using the source descriptions. Then the queries are executed against the data sources through wrappers in order to get the answers to the user query.

A mediator system is a system querying data sources. But it is not a repository of integrated sources. Data is not collected from each source, is not integrated with data from other sources and is not stored as it is at a warehouse. As a consequence, a mediator system is not able to directly execute queries. The user queries must be executed against the data sources through wrappers which translate the queries, reformulated into queries against the source schemas, in the language understood by each source.

2.2 *Research results*

Most existing data integration systems (such as Tsimmis, Information Manifold, Infomaster) follow a centralized approach of mediation between distributed and heterogeneous data sources, based on a single mediated schema (Chawathe et al. 1994, Genesereth et al.1997, Kirk et al.1995).

In this setting, works have developed languages for describing the mediated schema, the contents of information sources or the users' queries (Etzioni et al. 1994, Papakonstantinou et al. 1995). Other research works addressed the issue of providing sound and complete algorithms for rewriting queries using views over data sources. The information integration context is typical of the need of rewriting queries using views for answering queries because users of information integration systems do not pose queries directly to the sources in which data are stored but to a set of virtual relations that have been designed to provide a uniform and homogeneous access to a domain of interest. The instances of the rewriting problem that have been considered vary depending on the languages used for describing the queries and the views. The rewriting problem that has been extensively studied concerns the pure relational setting where queries and views are expressed as conjunctive queries over relational atoms (Halevy 2001). Finally, for a few years research works have focused on the use of ontologies for cooperative query answering in a mediator approach. In Bidault et al. (2000), a solution has been proposed for repairing unsatisfiable queries posed to a mediator by exploiting new queries that are satisfiable while being semantically close to the original one.

3 MEDIATION OF SERVICES OVER THE WEB

The use of the mediator approach in order to integrate services involves modifications in existing data sources integration systems. More precisely, the contents of the ontology will have to reflect the new domain of interest, i.e. a description of services. The views describing the contents of the data sources will have to be changed into views describing functionalities of computable services. All these aspects are going to be mentioned in section 4. In this section, we mention the modifications entailed by Web scalability. We first describe problems to be solved. Then we give some principles to make the mediator approach applicable to our new context.

3.1 *Difficulties entailed by Web scalability*

For scaling up the mediator approach to the Web, some difficulties have to be solved. One of them is related to the construction of the ontology, a crucial step when designing a mediator system. If manually done, even supported by convivial tools, it is a long, difficult and laborious modelling task. Techniques to automate this process are absolutely needed if mediators are used on the Web scale. Second, we want to deal with a wide-area integration of services. So, the description of their functionalities, i.e. the construction of views over services, must be made easier. Once again, there is a need for automation. Finally, the Web's deployment requires open systems, i.e. systems that are able to integrate immediately and very easily new services in a domain, without having to make the ontology evolved. To make this feasible, automation of the construction of the ontology must not be based on the integrated services. Ideally, the ontology should be built independently of them.

3.2 Principles for scaling up to the Web

The first principle is relative to semantic integration of services. Works addressed in the Web setting concerning standardization being important, we propose to exploit results obtained in this field, those concerning the Web service infrastructure (ebXML, SOAP) but overall results in standardization relative to business transactions. Organizations which work on these aspects (OTA 2002, UN/EDIFACT 2002) provide standardized specifications for the exchange of structured messages in particular domains. Such results are very interesting because they capture consensual knowledge among trading partners involved in a domain. So, we propose to exploit standardized specifications of structured messages in a domain for constructing the ontology of a service mediation system in the same domain.

Other principles can make the mediator systems scaling up to the Web. First, the principle of independence will introduce flexibility. This principle can be applied according to two ways. First, it refers to the separation of the ontology from the descriptions of the real integrated services. Second, it refers to a de-coupling of the various services. This decoupling includes information hiding based on the difference on internal business and public message exchange protocol interface descriptions. Coupling of the mediator system with services will be achieved via the interfaces. Second, the principle of strong mediation will enable anybody to access to any service in a scalable manner. That means that the mediator must be able to integrate numerous services, a lot of services been not identified as the mediator was designed.

4 ARCHITECTURE OF A SERVICE MEDIATION SYSTEM OVER THE WEB

Principles just described are applied in the setting of the PICSEL 2 project. We developed a service mediation system allowing various services that are distributed over the Web to be integrated. The services are all relative to a same domain. They deal with messages which correspond to standardized specifications delivered by a certain organization (Fig. 1). Public interfaces are associated with each service. They describe the exchanged messages in terms defined by the organization that we have considered. The query engine in the mediator system PICSEL (http://www.lri.fr/~picsel) has been reused.

The service mediation system is composed of an ontology describing the services of the domain. It groups a set of predicates modelling the domain of interest. This ontology provides support for querying. It also connects the available services to each other allowing their combination. The construction of the ontology is a semi-automated process exploiting the standardized specifications published by the organization that we have considered. The mediator system is also composed of views describing the functionalities of the available services. In this setting, these descriptions are automatically generated.

A user query formulated on the ontology is translated by the service mediation system into a set of queries against services using their descriptions. Then the queries are executed by the services. Wrappers are usually needed at this step in data sources integration systems to translate the queries in the language understood by each source. In the setting of this service integration system, the translation is automated.

In a first sub-section we present the specifications that we have exploited. Then we describe the building of the ontology and the generation of the service descriptions. The last part is devoted to the translation of the queries into messages.

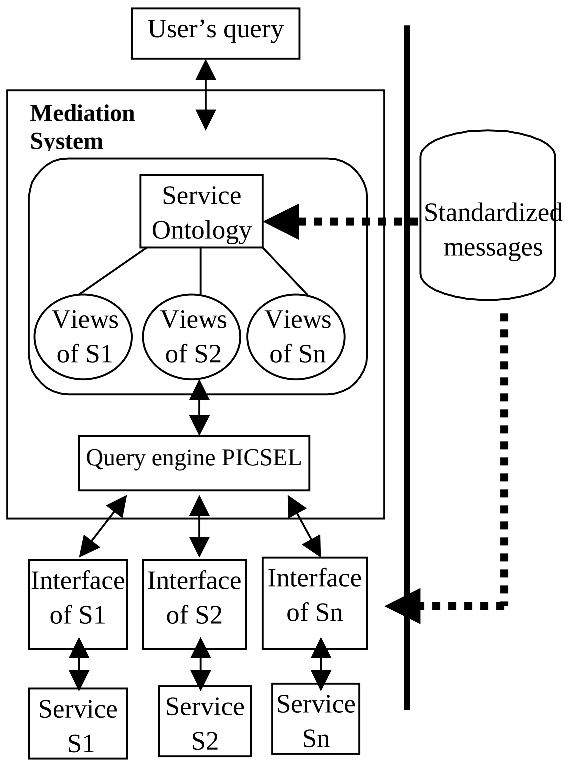

Figure 1. Architecture of the service mediation system developed in the PICSEL2 project.

4.1 The OTA standardized specifications

Our application relies on standardized specifications defined by the Open Travel Alliance (OTA 2002) in the travel industry. OTA is a non-profit organization, comprised of major airlines, hoteliers, car rental companies, leisure suppliers, travel agencies, global distribution systems and other interested parties working to define industry-wide, e-business specifications. These specifications form a common e-business language that will encourage development of systems to create new services to better meet expectations of travelers. The OTA has chosen XML as the vehicle for its specifications. The specifications rely upon the work of other standards developer organizations, specifically the ebXML initiative and the work of W3C.

115 XML-Schemas (Fig. 2) defined by OTA have been exploited in our experiment. These XML-Schemas define the elements to be used in messages when searching for availability and booking in the travel industry. Standardized messages delivered by OTA are of two kinds. There are messages specific to particular sub-domains in the travel industry (online flight booking service, hotel booking service and so on). And there are more technical messages usable for security considerations. We only used the first kind of specifications. These descriptions refer to the vocabulary of the domain useful for formulating queries. Technical descriptions have been eliminated. They refer to terms that are not relevant for querying.

```
<xs:element name="OTA_AirAvailRQ">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OriginDestinationInformation"
type="OriginDestinationInformationType"/>
                ...
        </xs:sequence>
        <xs:complexType
name="OriginDestinationInformationType">
            <xs:complexContent>
                <xs:extension base="TravelDateTimeType">
                    <xs:sequence>
                        <xs:element name="OriginLocation"
type="LocationType">
                        </xs:element>
                        <xs:element name="DestinationLocation"
type="LocationType">
                        </xs:element>
                            ...
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        ...
    </xs:complexType>
</xs:element>
</xs:schema>
```

Figure 2. Part of a XML-Schema describing the message AirAvailabilityRequirement.

4.2 Semi-automated construction of the ontology

The construction of the ontology in the mediator system is automated. The process that has been used exploits OTA specifications. It does not directly depend on the integrated services.

The building of the ontology is a two-step process. First, a very simple version is manually built. This process is guided by the OTA specifications. We classify the messages (for example AirAvailabilityRequirement and AirBookRequirement) for which a standardized content has been defined by OTA. Then, we group them into categories (for example AirBookingService). As a consequence, we obtain the two first levels of a class hierarchy (Fig. 3). Names of the classes in level 2 correspond to terms chosen by OTA to name messages. Names of the classes in level 1 correspond to names of categories of messages. The name of the root class denotes the domain of interest (Tourism Service). This class hierarchy is used when querying. Classes in level 2 correspond to services that a user might require. Thanks to this classification, services will not have to be selected among a big set of various services but among services belonging to categories. In a second step, the initial hierarchy is enriched from OTA specifications in a semi-automatic way. Elements useful to enrich the initial ontology (classes, properties, relations) are extracted from the specifications. Then, the acquired elements are structured. Finally, the structured model is represented in the formalism of the mediator PICSEL. The extraction step is semi-automatic. The output is a set of classes, a set of properties which characterize classes and a set of relations among classes. This process is described in Giraldo et al. (2002). The structuration phase is entirely automatic. Terms corresponding to the OTA messages are automatically extracted from the OTA specifications. They are names of classes. Their properties and their relations with other classes are also extracted. The terms corresponding to OTA messages correspond to those of the level 2 in the initial hierarchy manually built, too. That way, the mapping is made easier and the ontology can easily be enriched. At the end of this step, we obtain a network of relations establishing links among classes. The ontology is then represented using the terminological component of CARIN-ALN, the language used in PICSEL (Goasdoue et al 2000) which contains definition and inclusion statements about concepts. The translation process is entirely automatic.

The final ontology (Fig. 4) is composed of a classification of services (each service correspond to an OTA standardized message) precisely defined. Thus, for example, AirAvailabilityRequirement is a service which belongs to the category AirBookingService. It is related to information concerning the origin and the destination, more particularly an origin location, a destination location, a date, a time, specific information relative to the airline company, the class and so on. All these terms are useful to the user to precisely define required airline services.

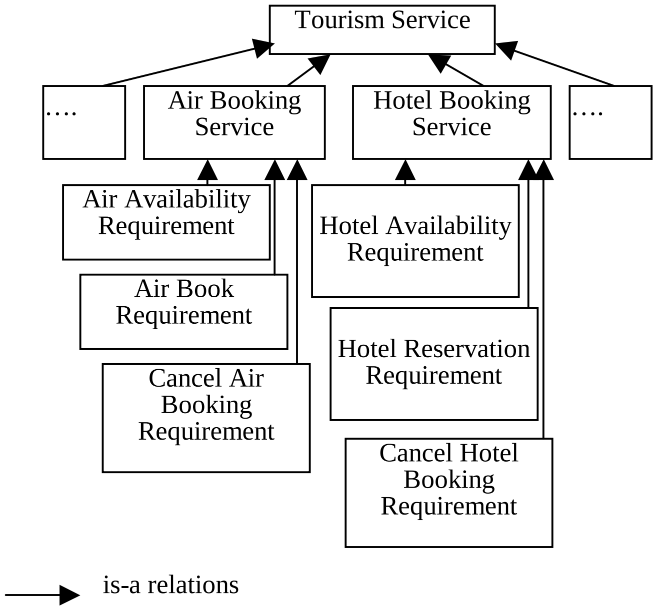

Figure 3. Part of the initial hierarchy of concepts.

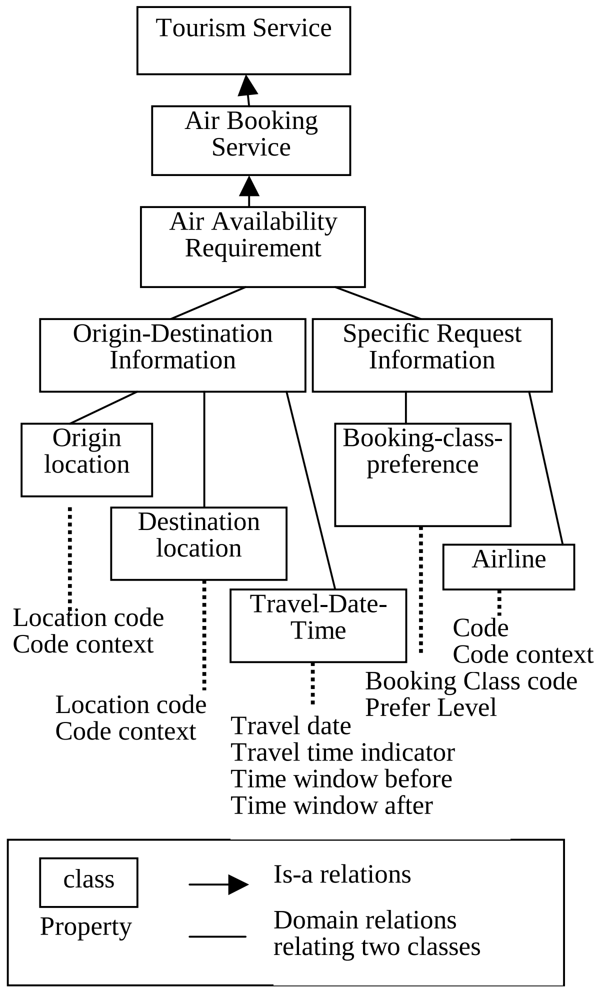

Figure 4. Part of the initial ontology enriched with classes, properties and relations.

4.3 Automated generation of service descriptions

In PICSEL the description of the content of each information source in terms of its views contains a set of logical implications $v_i(X) \Rightarrow p(X)$. These implications establish a link between each view v_i and the domain relation p whose instances can be found in the source.

The use of PICSEL to integrate OTA services (i.e. services able to handle OTA messages) leads to define for each service as many views as messages the service is able to deal with. Each service declares these messages in its public part (Fig. 1). The logical implications defining the views can then be generated in an automatic way from the names of the messages extracted from the public part of a service. For example, S_1-v_1 $\Rightarrow$ AirAvailabilityRequirement and S_1-v_2 $\Rightarrow$ AirBookRequirement could be two implications defining views with v_1 et v_2 the names of the views that are respectively associated with AirAvailabilityRequirement and AirBookRequirement in the ontology, two classes corresponding to names of OTA messages the service S_1 is able to deal with. So, these implications say that service S_1 fulfills both searching for availability and booking in the airline industry.

These view definitions are not sufficient. A user must be able to precisely define the service he is looking for. For example, he must be able to express that he wishes to have a flight from the airport CDG. Expressing such a request is perfectly possible using the ontology. But, for making the departure airport be taken into account by the mediator engine and be rewritten in terms of views, additional views have to be defined: views associated with all the relations in the ontology manipulated by user to precisely describe his requirement. Thus, we have to define views for all the elements composing the messages services are dealing with. We will have, for example, S_1-v_3 $\Rightarrow$ OriginDestinationInformation and S_1-v_4 $\Rightarrow$ OriginDestinationInformationAssociate with v_3 and v_4 two names of views that are respectively associated with OriginDestinationInformation and OriginDestinationInformationAssociate (a name of a role in CARIN-ALN). These implications are useful to indicate to the service S_1 data concerning the departure and the arrival airport of the flight required by the user.

Data composing OTA messages are represented in the ontology (Fig. 4). When the ontology has been translated into CARIN-ALN, these data have been represented by relations. The classes have been represented by unary relations, the roles (relations specific to the domain or properties) by binary relations. So, it is perfectly possible to automatically generate views associated with all the relations directly or indirectly related to a class corresponding to a message (a class in level 2 in the initial hierarchy, Fig.3).

4.4 Automatic generation of messages

Let Q(x) : Flight+HotelAvailabilityRequirement (x) be a query concerning both the availability of a return flight From Paris to Athens on May 2003 and of a room in a hotel in Athens for a trip of one week.

To prevent the user from having to navigate the whole ontology in order to select the terms he might desire to express his need, we use the query formulation help module of PICSEL (Reynaud et al. 2002). It proposes predefined complex queries and guides the user in the expression of the constraints to be fulfilled. Q is one of the predefined queries. The constraints refer to the place where the hotel is located, the departure and arrival place, the departure date of the flight and the duration of the trip. Their expression makes query formulation an interactive process.

Let us now assume that S_1, S_2 and S_3 are three services being made available by the PICSEL mediator. S_1 deals only with AirAvailabilityRequirement messages, S_2 deals only with HotelAvailabilityRequirement messages and S_3 deals with both of them. Answering queries in PICSEL resorts to find conjunctions of views, called rewritings, which, together with the ontology and the descriptions of the services, entail the initial query. A rewriting of a query can be viewed as a specialized query that can be directly executed by the services and which provides relevant answers to the initial query. In our example, the rewritings show that answering the query Q requires either to have access to the unique service S_3 either to integrate two services: S_1 S_2, S_1 S_3 or S_2 S_3. In the various combinations, data concerning hotel availabities provided by one of the services is intersected with data concerning flight availabilies provided by the other service to be sure to obtain a package including a return flight and a hotel stay with the same start date and the same end date. We plan to automate the generation of XML documents corresponding to the messages that must then be executed by the available services. Indeed, the components and the structure of the messages are known. They have been defined by OTA using XML-Schema. Furthermore, the rewritings given by the query engine PICSEL provide the values of the elements to insert into the various XML documents. For example, they indicate that the value of the destination location in the XML document describing the message to be performed by S_1 is Athens and that the value of the month in the travel date is May.

5 CONCLUSION

In this paper we have investigated the problem of integrating services over the Web. The approach based on the mediator systems is a first step to make these systems scaling up to the Web.

The specifications defined by OTA are central in our approach. They allow semantic integration and independence between the ontology in the mediator and the available services. Moreover, we shown that such an approach relying on OTA specifications makes automation of the construction of various components of a mediator feasible. It allows a semi-automated construction of the ontology, the automated description of the functionalities of the available services and we hope to be allowed to generate the content of the messages having to be executed by the various services.

Our future work will be devoted to the study of end-user requirements in comparison with professional needs. OTA specifications have been defined for a professional use. Our objective is to adapt the mediation system built from OTA specifications to make it more easily usable by end-users.

6 REFERENCES

Bidault A. & Froidevaux C. & Safar B. 2000. Repairing queries in a mediator approach. *In 14th European Conference on Artificial Intelligence*, pp. 406-410, Berlin.

Chawathe S. & Garcia-Molina H. & Hammer J. & Ireland K & Papakonstantinou Y. & Ullman J. D. & Widom J. 1994. The TSIMMIS project: Integration of heterogeneous information sources. *In 16th Meeting of the Information Processing Society of Japan*, pp. 7-18, Tokyo.

Data InterChange Standards Association. 2002. http://www.disa.org

Etzioni O & Weld D. 1994. A Softbot-Based Interface to the Internet. *Communications of the ACM*, 37(7):72-76.

Genesereth M. R. & Keller A. M. & Duschka O. M. 1997. Infomaster: an information integration system. In Joan M. Peckman (ed), proceedings, *ACM SIGMOD International Conference on Management of Data*: SIGMOD 1997: May 13-15, 1997, Tucson, Arizona, USA, volume 26(2) of SIGMOD Record (*ACM Special Interest Group on Management of Data*), pp. 539-542, New-York, NY 10036, USA. ACM Press.

Giraldo G. & Reynaud C. 2002. Construction semi-automatique d'ontologies à partir de DTDs relatives à un même domaine, *Journées Francophones d'Ingénierie des Connaissances*.

Goasdoue F. & Lattes V. & Rousset M.-C. 2000. The use of CARIN language and algorithms for Integration Information: the PICSEL system, *International Journal of Cooperative Information Systems*, 9(4):383-40.1

Halevy A. Answering queries using views: A Survey. 2001. *VLDB Journal*.

Kirk T. & Levy A. Y. & Sagiv Y. & Srivastava D. 1995. The Information Manifold. In C. Knoblock & A. Levy (eds*), Information Gathering from Heterogeneous, Distributed Environments*, AAAI Spring Symposium Series, Stanford University, Stanford, California.

Open Travel Alliance. 2002. www.opentravel.org.

Papakonstantinou Y. & Garcia-Molina H. & Widom J. 1995. Object exchange across heterogeneous information sources. In P. S. Yu and A. L. Chen (eds.), *11th Conference on data Engineering*, pp. 251-260, Taipei, Taiwan, IEEE Computer Society.

Reynaud C. & Safar B. 2002. Aide à la formulation de requêtes dans un médiateur. *13ème congrès francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle*, 8-10 janvier 2002, Angers.

United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport. 2002. United Nations Economic Commission for Europe, www.unece.org/trade/untdid.

Wiederhold G. 1992. Mediators in the architecture of future information systems, *Computer*, 25(3):pp.38-49.