An ontology-based mediation architecture for E-commerce applications

O. Corcho[1], A. Gomez-Perez[1], A. Leger[2], C. Rey[3], and F. Toumani[3]

[1] Universidad Politécnica de Madrid (ocorcho@fi.upm.es, asun@fi.upm.es)
[2] France Telecom R&D (alain.leger@rd.francetelecom.com)
[3] LIMOS, Université Blaise Pascal, France (rey@isima.fr, ftoumani@isima.fr)

Abstract. As part of the MKBEEM project, we present an ontology based mediation framework for electronic commerce applications. The framework is based on a mediator/wrapper approach that supports an integrated view over multiple heterogeneous sources. The MKBEEM mediation system allows to fill the gap between customers queries (possibly expressed in a natural language) and diverse specific providers offers. In contrast with many existing mediator based systems, our approach rests on a three-layer knowledge representation architecture which includes an *electronic services* ontology besides the usual domain ontology and sources descriptions layers. At the reasoning level, we propose a new mechanism, namely *dynamic discovery of e-services*, that acts in collaboration with the Picsel mediator system to effectively achieve the MKBEEM mediation tasks.

1 Introduction

The recent progress and wider dissemination of electronic commerce via the world wide web has dramatically increased the diversity and heterogeneities in corresponding systems. To avoid this progress to be hampered by closed markets with incompatible applications that cannot use each other services, there is a need for efficient and flexible mechanisms to handle the interoperability problem. Today, we expect that emerging standards will solve most of the syntactic infrastructure problems. However, dealing with sources from different domains and their semantics differences requires to handle the interoperability problem at the semantic level.

The notion of mediation has been proposed as the principal means to resolve problems of semantic interoperation [13]. Mediation architectures are generally based on the mediator/wrapper paradigm where information sources are "wrapped" to logical views so that their interfaces to the outside world are uniform [13,8]. These logical views are "glued" together through an integrated global schema usually called the domain ontology. In such architectures, a mediator acts as an interface between user queries or application programs and existing information sources: queries against the domain ontology are reformulated in terms of logical views and then sent to the remote sources.

In this paper, we present a **three-layer ontology-based mediation framework** for electronic commerce applications. In contrast with many existing mediator based systems, the proposed architecture includes an *electronic services ontology* besides the usual domain ontology and sources descriptions layers. Roughly speaking, an electronic service (e-service)[1] can be seen as an application made available via Internet and accessible by humans and/or other applications [2,3,6]. Examples of e-services currently available range from weather forecast, on-line travel reservation or banking services to entire business functions of an organization. Whereas in many e-commerce platforms, e-services are associated to providers, we propose to define an e-service as an *integrated provider-independent offer* available on a given e-commerce platform. Then, to effectively handle the mediation tasks, we rely on two reasoning mechanisms:

- the first, called **dynamic discovery of e-services**, allows to reformulate users queries against the domain ontology in terms of e-services. The aim here is to allow the users and applications to automatically discover the available e-services that best meet their needs at any given time,
- the second, called *query plan generation*, allows to reformulate a user query, expressed as a combination of e-services, in terms of providers views. The aim of this second issue is to allow the identification of the views that are able to answer to the query (knowing that, afterwards, the query plans will be translated into databases queries via the corresponding wrappers).

While the second reasoning mechanism, known as *query rewriting using views*, has already been addressed in the literature [11,5,8], the first is a quite new problem for which we propose a solution in this paper.

The rest of this paper is organized as follows. The context of this study, namely the MKBEEM[2] project, is introduced in Section 2. The architecture of the MKBEEM ontology is presented in Section 3. In Section 4, the dynamic identification of e-services is detailed. We conclude in Section 5.

2 The MKBEEM mediation system

The global aim of the MKBEEM project is to extend current electronic commerce platforms to reach a truly pan European and culturally open electronic commerce market. The main technical aim of MKBEEM is to create an *intelligent knowledge based multilingual* mediation service which displays the following features:

- Natural language interfaces for both the end user and the system's content providers/service providers.
- Automatic multilingual cataloguing of products by service providers.

[1] Also called web services.

[2] MKBEEM stands for Multilingual Knowledge Based European Electronic Marketplace (IST-1999-10589, 1st Feb. 2000 - 1st Aug. 2002).

- On-line e-commerce contractual negotiation mechanisms in the language of the user, which guarantee safety and freedom.

A detailed description of the MKBEEM project and relevant references related to this project can be found in [1].

Let us consider an example of a simplified *natural language request* scenario trained on the generic MKBEEM architecture to sketch the roles of the different components of the MKBEEM system: an end user submits to the MKBEEM system a natural language query. The query is received by the User Agent component which recognizes the user language and then forwards it to the corresponding Human Language Processing Server (HLP Server). The HLP Server is in charge of *meaning extraction*: he analyzes the input string and converts the query to an *Ontological Formula* (OF) which is a language-independent formula containing the semantic information of the corresponding phrase of human language. The OF is sent to the Rational Agent (RA) which is responsible of the dialogue management between the different components of the MKBEEM system. The RA forwards the OF to the Domain Ontology Server (DOS). The DOS module is responsible of storing, accessing and maintaining the ontologies used by the MKBEEM system. It also provides the core reasoning mechanisms needed to support the mediation services. Continuing with the example, the DOS achieves a *contextual interpretation of the formula* using its knowledge about the application domain. This task consists mainly in the identification of the offers (e-services) delivered by the MKBEEM platform that *"best match"* the ontological formula. The set of solutions computed by the DOS is sent back to the RA which, in collaboration with the *User Agent*, will ask the user to choose one solution and to complete, if any, the parameters that are missing. After this dialogue phase, the retained solution is sent back to the *DOS* to generate the *query plans*. A query plan contains information about the Content Providers Agents (CPA) that are able to answer to the user query. Then, thanks to specific wrappers belonging to CPA a query plan is translated into databases queries which are executed on the remote providers sources.

This example is a typical mediation instance: the user poses queries in terms of the integrated schema (e-services and domain ontology) rather than directly querying specific provider information sources. This enables users to focus on *what* they want, rather than worrying about *how* and *from where* to obtain the answers. Apart from the natural language processing and the wrapping steps, which are no further considered in this paper (cf. [1] for more details), the MKBEEM mediation relies on two rewriting steps achieved by the DOS:

1. A contextual interpretation step: the OF is rewritten into a few E-Services Formula (ESF) which have been dynamically identified as the combinations of e-services that "best match" the OF. Here are mainly used the e-services ontology and the domain ontology layers. In the sequel, we refer to this task as *dynamic discovery of e-services.*

2. A query plan generation step: the ESF are rewritten into a few query plans where the Content Providers Agents that are able to answer the user query are identified. Here is mainly used the sources descriptions layer. In the sequel, we refer to this task as *query plan generation*.

The problem of query plan generation, known as *query rewriting using views*, has already been studied in the research area of information integration [5,8]. That is why, in the MKBEEM mediation system, the DOS relies on the Picsel [8] mediator to handle this task. Picsel is a description logic based information integration system. It uses the $\mathcal{ALN}$-CARIN[3] language as the core logical formalism to represent both the domain ontology and the contents of information sources. The query rewriting algorithm implemented in the Picsel system is proved to be sound and complete. On the contrary, the dynamic discovery of e-services is a new research problem. This is our proposal to solve it, that is detailed in section 4, after the presentation of the three-layer ontology.

3 Architecture of the MKBEEM ontology

The MKBEEM Ontologies are defined as a knowledge structure to enable sharing and reuse of information. They provide a consensual representation of the electronic commerce field in two typical Domains (Tourism and Mail order) allowing the exchanges independently of the language of the end user, the service, or the content provider. Ontologies are used for classifying and indexing catalogues, for filtering user's query, for facilitating man-machine dialogues between users and software agents, and for inferring information that is relevant to the user's request.

The $\mathcal{ALN}$-CARIN language has been chosen as the ontology implementation language due to its inferential capabilities. Its description logic part, the $\mathcal{ALN}$ language, contains the following constructors:

- concept conjunction ($\sqcap$), e.g., the concept description $parent \sqcap male$ denotes the class of fathers (i.e., male parents),
- the universal role quantification ($\forall R\, C$), e.g., the description $\forall child\, male$ denotes the set of individuals whose children are all male,
- the number restriction constructors $\geq n R$ and $\leq n R$, e.g., the description ($\geq 1\ child$) denotes the class of parents (i.e., individuals having at least one children), while the description ($\leq\ 1\ leader$) denotes the class of individuals that cannot have more than one leader.
- the negation restricted to atomic concepts ($\neg A$), e.g., the description $\neg male$ denotes the class of individuals which are not males (females).

[3] $\mathcal{ALN}$-CARIN is a logical language combining Description Logics and Datalog rules [12].

As all the description logics, $\mathcal{ALN}$ comes equipped with well-defined, set-theoretic semantics. Furthermore, the interesting reasoning problems such as subsumption and satisfiability are decidable for $\mathcal{ALN}$ [7].

The MKBEEM ontologies are structured in three layers, as shown in Figure 1.

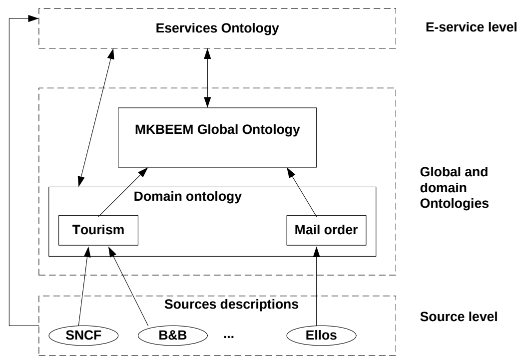

Fig. 1. Knowledge representation in the MKBEEM system.

These three layers are described below.

1. Global and domain Ontologies.
   The global ontology describes the common terms used in the whole MKBEEM platform. This ontology represents knowledge reusable on different domains while each domain ontology contains concepts corresponding to one of the domains of the MKBEEM partners (e.g;, tourism, mail orders, etc.). This decomposition in global and domain modular ontologies has allowed us to reuse many definitions from existing ontologies and more easily create definitions of the domain ontologies.

   Example 1. Table 1 provides the definitions of the concept Date and Time of the global ontology. The travel domain ontology contains, among others, the concept trip which has a departure place (the role depPlace), an arrival place (the role arrPlace) and may have a transport mean (the role transportMean), and the concept accommodation which is located in a place (role placedIn) and has a starting date (the role startDate).

2. E-services ontology.
   All the offers available in the MKBEEM platform are integrated and described in the e-services ontology. An e-service can be seen as a provider-independent predefined query corresponding to an existing offer in the MKBEEM platform.

   Example 2. According to e-services ontology given in Table 1, the MKBEEM platform delivers four offers:

- hotel, which allows to consult a list of hotels.
- apartment, which allows to consult a list of apartments.
- timetable1, which allows to consult a trip given the departure place, the arrival place, the departure date and the departure time.
- timetable2, which allows to consult a trip given the departure place, the arrival place, the arrival date and the arrival time.

  It is worth noting that these e-services are specified independently from a given provider.

3. Sources descriptions.

   At the lower knowledge level, sources descriptions specify the providers competencies, i.e., the description of the contents of the providers information sources. Each content provider agent is identified by its name (e.g., SNCF[4]) and can provide many views, each of which corresponds to a possible query (offer) on the provider information source.

   Example 3. In the example of the MKBEEM ontologies given in Table 1, four providers are defined: SNCF, cheapJet, Expedia and GitesDeFrance. There are two possibilities for querying the SNCF source to have information about train trips: given information about the departure place and the arrival place and either the departure date and the departure time (view timetableD) or the arrival date and the arrival time (view timetableA). The cheapJet source allows to consult a list of plane trips given the departure place, the arrival place, the departure date and the departure time (view planeTrip). The Expedia provider delivers information about hotels (view ExpediaHotel) while GitesDeFrance delivers information about apartments (view Renting)

The following section introduces the e-services dynamic discovery mechanism and illustrates how the proposed modular architecture is used to handle the DOS mediation tasks (i.e., contextual interpretation of ontological formulas and query plan generation).

4 E-services discovery

In the MKBEEM project, the *dynamic discovery* mechanism is used in association with the Picsel system to achieve the reasoning tasks in the DOS. The complementary roles of these two complex logical reasoning constitutes the description logic core for query processing in the MKBEEM system. They are in fact two different instances of the problem of rewriting concepts using terminologies [4].

In the following we illustrate the interest of *e-services discovery* using the MKBEEM ontologies provided in table 1.

[4] SNCF is the french railway company.

Table 1. Example of MKBEEM ontologies.

Global ontology

$$\text{time} \doteq (\leq 1\ \text{hour}) \sqcap (\geq 1\ \text{hour}) \sqcap (\forall\ \text{hour integer}) \sqcap (\leq 1\ \text{minute}) \sqcap (\geq 1\ \text{minute}) \sqcap (\forall\ \text{minute integer})$$

$$\text{date} \doteq (\leq 1\ \text{day}) \sqcap (\geq 1\ \text{day}) \sqcap (\forall\ \text{day integer}) \sqcap (\leq 1\ \text{month}) \sqcap (\geq 1\ \text{month}) \sqcap (\forall\ \text{month string}) \sqcap (\leq 1\ \text{year}) \sqcap (\geq 1\ \text{year}) \sqcap (\forall\ \text{year integer}) \sqcap (\leq 1\ \text{weekday}) \sqcap (\geq 1\ \text{weekday}) \sqcap (\forall\ \text{weekday string})$$

Travel Domain ontology

$$\text{trip} \doteq (\leq 1\ \text{depPlace}) \sqcap (\geq 1\ \text{depPlace}) \sqcap (\forall\ \text{depPlace string}) \sqcap (\leq 1\ \text{arrPlace}) \sqcap (\geq 1\ \text{arrPlace}) \sqcap (\forall\ \text{arrPlace string}) \sqcap (\geq 1\ \text{transportMean}) \sqcap (\forall\ \text{transportMean transportMeanType})$$

$$\text{accommodation} \doteq (\leq 1\ \text{placedIn}) \sqcap (\geq 1\ \text{placedIn}) \sqcap (\forall\ \text{placedIn string}) \sqcap (\leq 1\ \text{startDate}) \sqcap (\geq 1\ \text{startDate}) \sqcap (\forall\ \text{startDate date})$$

E-services ontology

$$\text{hotel} \doteq \text{accommodation} \sqcap (\leq 1\ \text{numberOfBeds}) \sqcap (\geq 1\ \text{numberOfBeds}) \sqcap (\forall\ \text{numberOfBeds number}) \sqcap (\leq 1\ \text{hotelCategory}) \sqcap (\geq 1\ \text{hotelCategory}) \sqcap (\forall\ \text{hotelCategory string})$$

$$\text{apartment} \doteq \text{accommodation} \sqcap (\leq 1\ \text{numberOfRooms}) \sqcap (\geq 1\ \text{numberOfRooms}) \sqcap (\forall\ \text{numberOfRooms number}) (\leq 1\ \text{apartmentCategory}) \sqcap (\geq 1\ \text{apartmentCategory}) \sqcap (\forall\ \text{apartmentCategory string})$$

$$\text{timetable1} \doteq \text{trip} \sqcap (\forall\ \text{depTime time}) \sqcap (\geq 1\ \text{depTime}) \sqcap (\leq 1\ \text{depTime}) \sqcap (\leq 1\ \text{depDate}) \sqcap (\geq 1\ \text{depDate}) \sqcap (\forall\ \text{depDate date})$$

$$\text{timetable2} \doteq \text{trip} \sqcap (\forall\ \text{arrTime time}) \sqcap (\geq 1\ \text{arrTime}) \sqcap (\leq 1\ \text{arrTime}) \sqcap (\leq 1\ \text{arrDate}) \sqcap (\geq 1\ \text{arrDate}) \sqcap (\forall\ \text{arrDate date})$$

Sources Descriptions

Provider	View Name	View Description
SNCF	timetableD	timetable1 $\sqcap$ ($\forall$ transportMean train)
SNCF	timetableA	timetable2 $\sqcap$ ($\forall$ transportMean train)
cheapJet	planeTrip	timetable2 $\sqcap$ ($\forall$ transportMean plane)
Expedia	ExpediaHotel	hotel
GitesDeFrance	Renting	apartment

Example 4. Let us assume that a given user submits to the MKBEEM system the following request expressed in a human language:
Q1: "I'll arrive in Paris on Monday and I look for an accommodation with swimming pool".
The query Q1 is first processed by the HLP Server and converted to the following *ontological formula*:
OF1 : "(trip)(V8), (accommodation)(V7), (arrDate)(V8,C9), (date)(C9), (day)(C9,15), (weekday)(C9,monday), (month)(C9,april), (year)(C9,2002), (arrPlace)(V8,paris), (leisure)(V7,swimmingPool)".

Then, given such an ontological formula, the *e-services discovery* is used by the DOS to identify the corresponding relevant service(s) in the ontology of e-services. This task is achieved in two steps:

1. Converting an ontological formula F into a concept description Q_F
   This task depends on the structure of the ontological formula and on the expressive power of the target language. In the context of the MK-BEEM project, the current ontological formulas generated by the HLP Server have relatively simple structures that can be described using the small description logic $\mathcal{FL}_0 \cup \{(\geq nR)\}$. This logic contains the concept conjunction constructor ($\sqcap$), the universal role quantification constructor ($\forall R.C$) and the minimal number restriction constructor ($\geq nR$). In this case, we can achieve this task by computing the so-called *most specific concept* [7] corresponding to the ontological formula.

   Example 5. The concept description Q_{OF1} corresponding to the ontological formula $OF1$ given in the previous example is:
   $$Q_{OF1} \doteq \mathsf{trip} \sqcap \mathsf{accommodation} \sqcap (\geq 1\ \mathsf{arrPlace}) \sqcap (\forall\ \mathsf{arrPlace\ string}) \sqcap (\geq 1\ \mathsf{arrDate}) \sqcap (\forall\ \mathsf{arrDate}\ (\mathsf{date} \sqcap (\geq 1\ \mathsf{day}) \sqcap (\forall\ \mathsf{day\ integer}) \sqcap (\geq 1\ \mathsf{year}) \sqcap (\forall\ \mathsf{year\ integer}) \sqcap (\geq 1\ \mathsf{month}) \sqcap (\forall\ \mathsf{month\ string}) \sqcap (\geq 1\ \mathsf{weekday}) \sqcap (\forall\ \mathsf{weekday\ string}))) \sqcap (\geq 1\ \mathsf{leisure}) \sqcap (\forall\ \mathsf{leisure\ string})$$

2. Selecting the relevant e-services.
   This problem can be stated as follows: given a user query Q_F and an ontology of e-services T, find a description E, built using (some) of the names defined in T, such that E contains as much as possible of common information with Q_F and as less as possible of extra information with respect to Q_F. We call such a rewriting E a *best cover* of Q_F using T. Therefore, our goal is to rewrite a description Q_F into the closest description expressed as a conjunction of (some) concept names in T.

A best cover E of a concept Q using T is defined as being any conjunction of concept names occurring in T which shares some common information with Q, is consistent with Q and minimizes, in this order, the extra information in Q and not in E and the extra information in E and not in Q. Once the notion of a best cover has been formally defined, the second issue to be addressed is how to find a set of e-services that best covers a given query. This problem, called *best covering problem*, can be stated as follows: given an ontology T and a query description Q, find all the best covers of Q using T.

More technical details about the best covering problem can be found in [9,10]. To sum up, the main results that have been reached are:

- The precise formalization of the best covering problem in the framework of languages where the difference operation is semantically unique (e.g., the description logic $\mathcal{FL}_0 \cup \{(\geq nR)\}$).

- A study of complexity showed that this problem is NP-Hard.
- A reduction of the best covering problem to the problem of computing the minimal transversals with minimum cost of a weighted hypergraph.
- Based on hypergraph theory, a sound and complete algorithm that solves the best covering problem was designed and implemented.

Example 6. Continuing with the example, we expect the following result to be returned by the DOS:

	Identified services	**Rest**	**Missed information**
Solution 1	timetable2, apartment	leisure	arrTime.hour, arrTime.minute numberOfRooms, apartmentCategory
Solution 2	timetable2, hotel	leisure	arrTime.hour, arrTime.minute numberOfBeds, hotelCategory

These solutions correspond to the combinations of e-services that best match the ontological formula OF1. For each solution, the DOS computes the extra information (column *Missed Information*) brought by the e-services but not contained in the user query. The column Rest contains the extra information (leisure) contained in the user query and not provided by any e-services. This means that, in the proposed solutions the requirement concerning the leisure is not taken into account.

To continue with the example, assume that the user chooses the first solution (timetable1, apartment). Then, he is asked to complete the missed information: the arrival time (hour and minutes), the apartment category and the number of rooms in the apartment. The result is a global query Q, expressed as an E-service Formula (ESF), that will be sent to the Picsel system to identify the Content Provider Agent (CPA) that are able to answer to this query. In our example, the execution of the query Q using the ontologies depicted in Table 1 leads to the unique query plan:
$[SNCF\_timetableA, GitesDeFrance\_Renting]$.

Implementation and validation The *e-services discovery* algorithm has been implemented as an integrated component in the MKBEEM prototype. This prototype is built as a set of Enterprise Java Beans (EJB) components that interact with each other. There are also some components dedicated to the interaction with the user interface built with Java Server Pages (JSPs) or Servlets. Finally, some of these components have the functionality of interacting with the remote (or locally duplicated) databases in the provider information systems. The MKBEEM prototype has been validated on a pan-European scale (France and Finland), with three basic languages (Finnish, English and French) and one optional language (Spanish), in two distinct end-user fields: 1) Business to consumer on-line sales, and 2) Web based travel/tourism services.

5 Conclusion

In this paper, we have presented an ontology-based mediation architecture structured in three layers: global/domain ontology, e-services ontology and provider sources descriptions. In this context, a new reasoning mechanism, called *dynamic discovery of e-services*, is proposed to allow the users to automatically discover the available e-services that best meet their needs. This reasoning mechanism is used in association with the Picsel mediator system to effectively handle the mediation tasks. The modularity of this architecture together with the associated reasoning mechanisms allow to make the whole system provider-independent and more capable to face the great instability and the little lifetime of e-commerce offers and e-services. The proposed architecture has been successfully experienced in the context of the MKBEEM project.

References

1. MKBEEM web site : http://www.mkbeem.com.
2. Data Engineering Bulletin: Special Issue on Infrastructure for Advanced E-Services. 24(1), IEEE Computer Society, 2001.
3. The VLDB Journal: Special Issue on E-Services. 10(1), Springer-Verlag Berlin Heidelberg, 2001.
4. F. Baader, R. Küsters, and R. Molitor. Rewriting Concepts Using Terminologies. In *KR'2000, Colorado, USA*, pages 297–308, Apr. 2000.
5. C. Beeri, A.Y. Levy, and M-C. Rousset. Rewriting Queries Using Views in Description Logics. In *ACM PODS , New York, USA*, pages 99–108, Apr. 1997.
6. F. Casati and M-C. Shan. Models and Languages for Describing and Discovering E-Services. In *ACM SIGMOD, Santa Barbara, USA*, May 2001.
7. F. Donini and A. Schaerf M. Lenzerini, D. Nardi. Reasoning in description logics. In *Gerhard Brewka, editor, Foundation of Knowledge Representation*, pages 191–236. CSLI-Publications, 1996.
8. F. Goasdoué and M-C Rousset V. Lattès. The Use of CARIN Language and Algorithms for Information Integration: The PICSEL System. *IJICIS*, 9(4):383–401, 2000.
9. M.S. Hacid, A. Leger, C. Rey, and F. Toumani. Computing concept covers: A preliminary report. In *Workshop on Description Logics. Toulouse, France*, Apr. 2002.
10. M.S. Hacid, A. Leger, C. Rey, and F. Toumani. Dynamic discovery of e-services: A description logics based approach. In *Proceedings of the 18th French conference on advanced databases (BDA), Paris*, pages 21–25, Oct. 2002.
11. A.Y.Levy, A .Rajaraman, and J.J. Ordille. Querying Heterogeneous Information Sources Using Source Descriptions. In *VLDB'96, Mumbai (Bombay), India*, pages 251–262. Morgan Kaufmann, Sep. 1996.
12. A.Y. Levy and M-C. Rousset. Combining Horn Rules and Description Logics in CARIN. *Artificial Intelligence*, 104(1-2):165–209, 1998.
13. G. Wiederhold and Michael R. Genesereth. The Basis for Mediation. In *CoopIS, Vienna, Austria*, pages 140–157, May 1995.