A Natural Language Mediation System for E-Commerce applications: an ontology-based approach

Johannes Heinecke[1] and Farouk Toumani[2]

[1] France Télécom R&D, DMI/GRI/LN, F-22307 Lannion cedex, France,
johannes.heinecke@rd.francetelecom.com

[2] LIMOS, Université de Clermont-Ferrand, F-63173 Clermont-Ferrand, France,
ftoumani@isima.fr

Abstract. This paper describes how ontologies are used to mediate between languages and to infer answers to user questions in the multilingual e-commerce mediation system MKBEEM.[3] As an example, the paper discusses how a complex user request in human language is transformed into an ontological formula and subsequently exploited to identify a service which matches best. The MKBEEM-system prototype is in principle language independent and has been tested for the time being in Finnish, French, English and Spanish.

1 Introduction

The MKBEEM-project integrates knowledge-based processing (Knowledge Representation and Reasoning) and Human Language processing in providing multilingual e-commerce mediation services in order to allow a customer to use her own language, independent of the country where the product/service provider is based in. The consortium aims at proving that the technology concept is robust for given domains, and thereby bringing advances in both technology and services.

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 system's content providers/service providers and the end user.

[3] The project MKBEEM (Multilingual Knowledge Based European Electronic Marketplace, 2000-2002, http://www.mkbeem.com/) is a project funded by the European Commision (IST-1999-10589). The consortium, coordinated by France Telecom R&D (F), consists of VTT Information Technology (FIN), Universidad Politécnica de Madrid (E), National Technical University of Athens (GR), CNRS-LIRMM (F), SchlumbergerSema (E), Société Nationale des Chemins de Fer (F), and Ellos (FIN).

- Automatic multilingual cataloguing of products by service providers.
- On-line e-commerce contractual negotiation mechanisms in the language of the user, which guarantee safety and freedom.

Ontologies have been widely recognized as a central solution for sharing conceptions of goods and services among parties in e-commerce [1,2,3]. A recent survey by IBM and Icon Medialab found that in the Scandinavian countries on average 35% and in Finland up to 60% of purchase attempts failed in eShops. A major cause for this bad usability was that the customers could not find the requested products. Simple string based product search facilities are not enough. "No product available" is an insufficient answer, if the selection includes comparable goods or if the user just happens to use terms that differ from the ones in the catalogue. eShops need to solve the best possible offerings matching the user requirements, like human shopkeepers would do. The required question-answering capabilities can be realized by inferring based on domain ontologies, e.g. product models, and related generic ontologies. Moreover, ontologies can be used to facilitate multilinguality. In the MKBEEM-project, ontologies serve as the central solution for providing multilinguality and intelligent question answering [4]. The main result of this project is a multilingual e-commerce mediation system. It supports three main functionalities:

- *Multilingual cataloguing*, which enables providers to describe in their own language the goods and services that are on sale. Textual descriptions are translated automatically. Facts about products are extracted automatically into a language neutral form that complies with the product models of the domain ontology.
- *Processing of customer language information requests*, which is based on the co-operation between *human language processing* and *ontologies* of the commerce domain, the related products and generic common sense issues. Ontologies bridge between languages and also help in implementing fuzzy information search.
- *Multilingual trading*, which among other things applies an e-commerce ontology in carrying out contract terms adaptation for a particular shopping basket taking into account the countries of the seller and the buyer.

The MKBEEM-system prototype supports currently Finnish, French, English and Spanish. The technology can be easily adapted to other languages as well since all ontological knowledge is language independent. Feasibility tests have been conducted with test users since September 2002 in France and in Finland for mediating clothes, railway tickets, Finnish holiday cottages and French hotel room reservations, and car rental.

2 Technical Approach

In MKBEEM, ontologies are used to 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 [2,5]. Ontologies are used for classifying and indexing catalogues, for filtering user queries, for facilitating man-machine dialogues between users and software agents, and for inferring information that is relevant to the user requests.

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

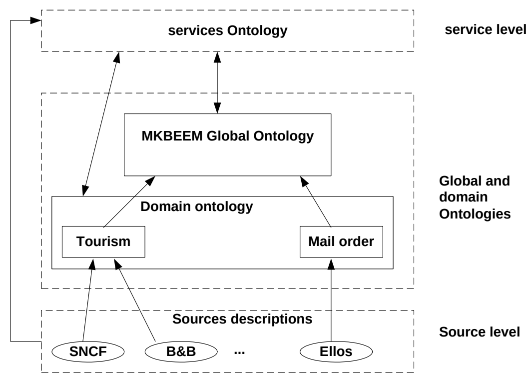

Fig. 1. Knowledge representation in the MKBEEM-system

The *global ontology* describes the common terms used in the whole MKBEEM-platform. This ontology represents the general knowledge in different domains (e.g., date, time) while each *domain ontology* contains specific concepts (e.g., trip) corresponding to vertical domains such as tourism and mail orders. The *service ontology* describes all the offers available in the MKBEEM-platform in terms of classes of services, e.g., service capabilities, non-functional attributes, etc. Service classes are generic in the sense that they are described independently from a specific provider. The source descriptions specify concrete services (i.e. provider offers) in terms of the service ontology. A further ontology is the linguistic domain ontology which assures an unambiguous interpretation of the user requests (see below in section 3).

The MKBEEM-mediation system allows to fill the gap between customer queries and diverse concrete providers offers. In a typical scenario, an end user submits to the MKBEEM-system a natural language query. The query is processed by a HUMAN LANGUAGE PROCESSING SERVER (HLP Server) which is in charge of *meaning extraction*: it analyzes the input string and converts the query into an *ontological formula* (OF) which is a language-independent formula containing the semantic information of the corresponding phrase in human language in terms of the service ontology. The OF is then sent to the DOMAIN ONTOLOGY SERVER (DOS). The DOS 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. 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 (services) delivered by the MKBEEM-platform that "*best match*" the ontological formula. The aim here is to allow the users/applications to automatically discover the available services that best meet their needs, to examine their capabilities and to possibly complete missing information. The set of solutions computed by the DOS is sent back to the user to choose one solution and to complete the parameters, if any, 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 concrete services that are able to answer to the user query. Then, thanks to the technical information provided in the source descriptions, a query plan is translated into specific provider requests which are executed on the remote provider platforms.

Thus, the user poses queries in terms of the integrated knowledge (services and domain ontology) rather than directly querying specific provider information data-bases. This enables users to focus on *what* they want, rather than worrying about *how* and *from where* to obtain the answers.

Apart from the wrapping steps, which is no further considered in this paper (cf. [4] for more details), the MKBEEM-system relies on two mediation tasks, namely human language processing and service identification. These tasks are discussed in detail in the remainder of this paper.

3 Human Language Requests Analysis

Within MKBEEM, we currently cover three basic services of the tourism domain, i.e. train reservation, accommodation reservation, car rental as well as mail ordering of clothing. In all of these cases, human languages allow a wide range of expressions and the related linguistic ontology therefore contains all the necessary information. Another benefit of this is that it helps the user to specify as much parameters as needed in a single request, in natural language, thus avoiding tiresome form-filling. The combination of several requests (e.g. *I want to visit Paris and reserve a hotel next weekend*) is also possible.

To ensure that the generated, language neutral ontological formulas will contain all relevant information given by the user, the user request is treated in several interdependent steps [6].

Since the MKBEEM-prototype is multilingual, the first step is to identify the language of the user request. In the next step, it is analyzed and a language independent semantic graph is created. The linguistic analysis is based on a dependency syntax, a set of language dependent rules comparable to the Semantic Interpretation Rules of Discourse Representation Theory [7] and a set of language independent predicates. To ensure the ontological appropriateness of the generated semantic graph, it is checked by the linguistic domain ontology developed for this purpose.[4] Any inappropriate semantic graph is deleted from the set of possible solutions. Finally, in order to deal with travel dates etc. (especially in the tourism domain), temporal expressions which are relative to the

[4] This is done by PICSEL (ONTOCLASS) [8].

time of utterance (deictic elements like *now*, *today*, *in two hours*, *in five days*, *next Monday*, *at ten to eleven pm*) or incomplete or varying dates (*the 12th of April*, *on Good Friday* are transformed into the corresponding absolute temporal expression (if no exact time is specified, it is not generated):

temporal expression	transformation
now	*17.06.2003 13:56*
today	*17.06.2003*
in two hours	*17.06.2003 15:56*
in five days	*22.06.2003*
next Monday	*21.06.2003*
at ten to eleven pm	*17.06.2003 22:50*
the 12th of April	*12.04.2004*
on Good Friday	*9.04.2004*

The next step is the transformation of the internal semantic representation into the ontological formula, which is also understood by other modules. The concepts (and roles) differ considerably from the linguistic ontology due to the fact that linguistic expressions and semantic nuances are present in the semantic representation, which are not needed in the ontological formula. So for instance temporal/modal information (*I want to*/*I would like to*/*we willwe have to*) must be eliminated by the transformation. Further, different lexemes expressing a move (*go*/*arrive*/*depart*/*travel*/*be in*/*visit*) need to be mapped on the concept trip, which is the only move-concept of the service ontology (see below)

As an example we take a typical user request, like the following example 1:

> Example 1. *"I'll arrive in Paris on Monday evening and I look for an accommodation with swimming pool."*

The request inquires information on public transport to Paris on (next) Monday evening (uttered on Tuesday, 17th June). After analyzing the sentence and processing the relative temporal information, we obtain an internal, language independent, semantic representation:[5]

Semantic representation 2. (simplified)

coord(coord1=x3005, coord2=x3006) &
arrival(destination=x3009, origin=u3010, situation=x3005, agent=x3013) &
speaker(theme=x3013) &
Paris(town=u3015, location=x3009) &
weekday˜monday(date=x3005, wday=u3014) &
monthday˜23(date=x3005, day=u3069) &
month˜june(date=x3005, month=u3070) &
year˜2003(date=x3005, year=u3071) &

[5] In this representation, we use predicates (in *italic*) and arguments (between parentheses) which indicate the semantic roles of the predicates. The predicates are linked via the variables of the arguments. For instance, the agent of the predicate arrival (x3013) is the speaker. Variables may link more than two predicates: x3005 links the situation of arrival with the date (Monday, 23rd June) and the time (18:00).

hour˜18(time=x3005, hour=u3072) &
minute˜0(time=x3005, minute=u3073) &
staying(agent=x3021, situation=x3006, place=x3022, means=x3023,
leisure=x3024) &
speaker(theme=x3021) &
accomodationorg(city=x3022, theme=x3023, leisure=x3024) &
swimmingPool(type=x3024).

As users are not directly concerned by the organization of data provided by information systems (in our case train, car rental, tourism), the main difficulty is to map efficiently the *user concepts* (*go*, *arrive*, *depart*, *take a train*, etc.), identified by the HLP, onto the information system (IS) concepts. Since some user requests are complex utterances, mixing motion verbs with absolute or relative time and space representation, the linguistic ontology is first used to constrain the parser during the construction of the linguistic formulas and to reduce the ambiguity ([9], cf. also [10]). In a next step irrelevant information (from an application point of view) must be pruned to produce a new formula compliant to the DOS (cf. section 4), devoted identify the service and to plan the data-base queries.

The linguistic ontology has been designed using the experience and knowledge gained in a previous project using description logics (Picsel[6]), and which tools have been enriched to fit the needs of linguistic analyzer.

Usually, ontologies are organized as directed graphs and use multiple inheritance. In consequence the more general concepts subsume the more specific. In contrast to superordinates which are less specific concepts, the greatest common subsumee (GCS) are more complete. Our experience, however, shows, that IS concepts are rather GCS than superordinates.

As outlined in [9] we use a common formalism for information representation (ontology). The ontologies are represented in Picsel, where concepts are unary predicates and roles binary predicates joining two concepts or a concept and a constant. The common inter-module communication language is Carin-$\mathcal{ALN}$[7] which is in the framework of Description Logics. As a consequence the HLP must transform utterances into formulas (using the inter-module communication language)

Picsel ontologies are organized as directed graphs and use multiple inheritance. Thus in Picsel (and other DLs) the more generic concepts subsume the more specific ones. In natural languages, however, more general concepts

[6] "Picsel is an information integration system over sources that are distributed and possibly heterogeneous. The approach which has been chosen in Picsel is to define an information server as a knowledge-based mediator in which Carin is used as the core logical formalism to represent both the domain of application and the contents of information sources relevant to that domain." [11]:383, [8].

[7] Carin is a family of theoretical languages for knowledge representation, Carin-$\mathcal{ALN}$ is the most expressive description logic for which subsumption and satisfiability are polynomial [11].

combine features of more specific ones. In consequence, the greatest common subsumees (GCS) are the best candidates to represent these more general concepts. Our experience shows that information system applications should rather use GCS than concepts of the linguistic sub-ontology (LSO) in order to keep the power of inheritance and to manage a more generic notion at the same time.

Discrepancies between the semantic representation (of the user request) and the main ontology must thus be bridged: The semantic representations (graphs) are using the LSO (i.e. concepts and roles defined in the LSO). To obtain the ontological formula, we need to rewrite this representation in service ontology (SO) terms. In order to achieve this, the principal rewriting rule is to replace the LSO concept (as found during the syntactical-semantic analysis) by the GCS concept of the SO.

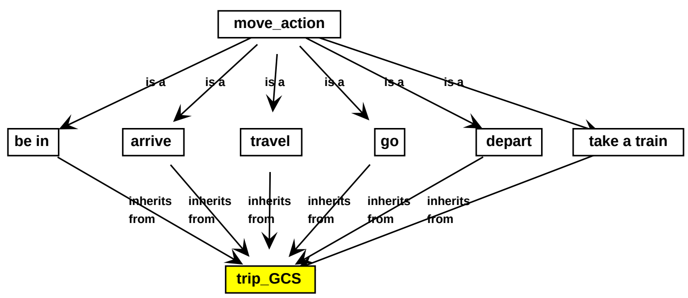

Fig. 2. Links between linguistic ontology and service ontology

As figure 2 shows, the service ontology concept trip_GCS[8] inherits all existing information from the (linguistic) concepts go, arrive etc., which express the meanings of the verbs in question. The motion verb therefore can be rewritten using the GCS (in this case trip_GCS). The resulting formula can now be interpreted correctly within the service ontology. Taking our example 1 (page 5), the semantic representation 2 (page 6) is thus transformed into the corresponding ontological formula 3 in service ontology terms. (cf. figure 3).[9]

---

[8] The suffix _CGS is used only for clarity.

[9] An ontological formula is a particular kinds of conjunctive queries expressed on unary and binary predicates. Roughly speaking, the ontological formula 3 of our example defines two concepts:

- trips, specified through the variable V5609, whose destination is *Paris* (denoted by properName_Paris) and whose arrival date is: *Monday, 23th of June 2003* at *18:00*.
- accommodations, specified through the variable V5610, offering a *swimming pool* among their leisure facilities.

Hence, this ontological formula 3 expresses the fact the user is interested by the instances of these two concepts.

Ontological formula 3.

(trip)(V5609),
 (arrPlace)(V5609, properName_Paris),
(date)(C63),
 (weekday)(C63, monday),
 (day)(C63, 23),
 (month)(C63, june),
 (year)(C63, 2003),
 (arrDate)(V5609, C63),
(time)(C64),
 (hour)(C64, 18),
 (minute)(C64, 0),
 (arrTime)(V5609, C64),
(accommodation)(V5610),
 (leisure)(V5610, swimmingPool)

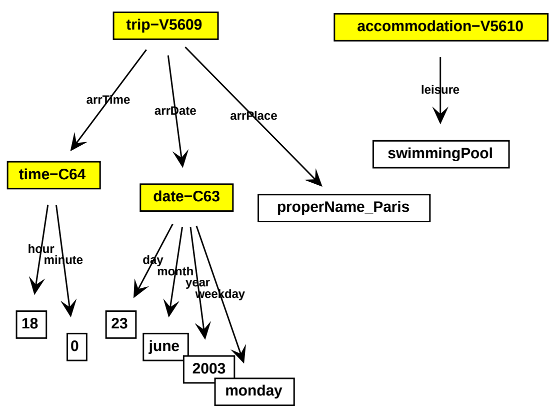

Fig. 3. Visualization of the ontological formula used for the service identification

4 Service Identification

In MKBEEM, service identification is achieved by means of a *dynamic service discovery* reasoning mechanism. Dynamic service discovery 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 [12].

The following example illustrates the interest of the *service discovery* reasoning mechanism.

Let us consider an e-commerce platform that delivers the following 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 journey given the departure place, the arrival place, the departure date and the departure time.
- timetable2, which allows to consult a journey given the departure place, the arrival place, the arrival date and the arrival time.

Let us assume that, according to architecture of the MKBEEM-ontology, these offers are formally described in a given service ontology. Consider now, the example 1 (above page 5) and the ontological formula 3 (page 8) created by HLP Server. Now the *service discovery* is used by the DOS to identify the corresponding relevant service(s) in the service ontology. 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 MKBEEM-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* [13] corresponding to the ontological formula.
   The concept description Q_{OF1} corresponding to the ontological formula $OF1$ given in the previous example is:

$$
\begin{aligned}
Q_{OF1} \doteq\ & \mathsf{accommodation} \\
& \sqcap (\geq 1\ \mathsf{leisure}) \\
& \sqcap (\forall\ \mathsf{leisure\ string}) \\
& \sqcap \mathsf{trip} \\
& \sqcap (\geq 1\ \mathsf{arrPlace}) \\
& \sqcap (\forall\ \mathsf{arrPlace\ string}) \\
& \sqcap (\geq 1\ \mathsf{arrDate}) \\
& \sqcap (\forall\ \mathsf{arrDate}\ (\mathsf{date} \sqcap (\geq 1\ \mathsf{day}) \sqcap (\forall\ \mathsf{day\ integer}) \\
& \qquad \sqcap (\geq 1\ \mathsf{year}) \sqcap (\forall\ \mathsf{year\ integer}) \\
& \qquad \sqcap (\geq 1\ \mathsf{month}) \sqcap (\forall\ \mathsf{month\ string}) \\
& \qquad \sqcap (\geq 1\ \mathsf{weekday}) \sqcap (\forall\ \mathsf{weekday\ string}))) \\
& \sqcap (\geq 1\ \mathsf{arrTime}) \\
& \sqcap (\forall\ \mathsf{arrTime}\ (\mathsf{time} \sqcap (\geq 1\ \mathsf{hour}) \sqcap (\forall\ \mathsf{hour\ integer}) \\
& \qquad \sqcap (\geq 1\ \mathsf{minute}) \sqcap (\forall\ \mathsf{minute\ integer})))
\end{aligned}
$$

2. Selecting the relevant services:

This problem can be stated as follows: given a user query Q_F and an ontology of 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 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 [14,15]. To sum up, the main results that have been reached are:

- The precise formalisation 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 ([16]).
- 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.

Continuing with the example, we obtain the following result from the DOS:

	identified services	**rest**	**missing information**
Solution 1	timetable2,	leisure	depPlace
	apartment	—	numberOfRooms, apartmentCategory
Solution 2	timetable2,	leisure	depPlace
	hotel	—	numberOfBeds, hotelCategory

Table 1. Results from the DOS

These solutions correspond to the combinations of services that best match the ontological formula *OF1*. For each solution, the DOS computes the extra information (column *missing information*) brought by the 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 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 (timetable2, apartment). Then, he is asked to complete the missing information: the departure place, the apartment category and the number of rooms the user wants in the apartment. The result is a global query Q, expressed as a service formula, that will be sent to the Picsel system to identify the providers which are able to answer to this query.

5 Conclusion

In this paper we have described the successful implementation of multilingual mediation system, based on knowledge which is coded in ontologies. It shows, how after the identification of the language, a user request is analysed and transformed into an language independent ontological representation. This representation is used to identify the service/product the user wants to consult/buy by the help of service ontologies. Existing parameters are extracted, missing ones request in a subsequent step. Finally the data base of the appropriate content provider is contacted to present the user the results of his initial requests.

Acknowledgements

The authors would like to thank all their colleagues in the MKBEEM consortium for the excellent cooperation as well as the European Commission for supporting the reported work. We are grateful to three anonymous reviewers for valuable hints for improvement of an earlier version of this paper. All remaining errors are our own.

References

1. Fensel, D.: *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, Heidelberg (2001)
2. Léger, A., Michel, G., Barrett, P., Gitton, S., Gómez-Pérez, A., Lehtola, A., Mokkila, K., Rodrigez, S., Sallantin, J., Varvarigou, T., Vinesse, J.: Ontology domain modeling support for multi-lingual services in E-Commerce: MKBEEM. In: *Proceedings of European Conference on Artificial Intelligence (ECAI 2000), Berlin. Workshop on Applications of Ontologies and Problem-Solving Methods*. (2000)
3. Gómez-Pérez, A.: Requirement, Choice of a Knowledge Representation and Tools. Technical report, Universidad Politécnica de Madrid, Madrid (2001)
4. MKBEEM: The MKBEEM-project. http://www.mkbeem.com/ (2000-2002)
5. Corcho, O., Gómez-Pérez, A., Léger, A., Rey, C., Toumani, F.: An Ontology-based Mediation Architecture for e-commerce applications. In: *Intelligent Information Systems 2003 (IIS2003), Zakopane, Poland*. Advances in Soft Computing, Heidelberg, Springer (2003)
6. Lehtola, A., Heinecke, J., Bounsaythip, C.: Intelligent Human Language Query Processing in MKBEEM. In: *Proceedings of the Workshop on Ontologies and Multilinguality in User Interface. HCII 2003, Creta, Greece*. (forth)

7. Kamp, H., Reyle, U.: *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Studies in Linguistics and Philosophy 42. Kluwer, Dordrecht (1993)
8. Picsel: The Picsel-project. http://www.lri.fr/~picsel/ (1999-2001)
9. Heinecke, J., Cozannet, A.: Ontology-Driven Information Retrieval. a proposal for multilingual user requests (2003) Paper given at the Workshop on Ontological Knowledge and Linguistic Coding at the 25th annual meeting of the German Linguistics, Feb. 26-28. 2003.
10. Guarino, N.: Towards a common ontology for multilingual information access and integration. Paper held at the Third Meeting of SIG on Intelligent Information Agents. Barcelona (1999)
11. Goasdoué, F., Lattes, V., Rousset, M.C.: The Use of CARIN Language and Algorithms for Information Integration. The Picsel-Project. *International Journal of Cooperative Information Systems* **9:4** (2000)
12. Baader, F., Küsters, R., Molitor, R.: Rewriting Concepts Using Terminologies. In: *Proceedings of the Seventh International Conference on Knowledge Representation and Reasoning, Colorado, USA*. (2000) 297–308
13. Donini, F., M. Lenzerini, D. Nardi, A.S.: Reasoning in description logics. In Brewka, G., ed.: *Foundation of Knowledge Representation*, CSLI-Publications (1996) 191–236
14. Hacid, M., Leger, A., Rey, C., Toumani, F.: Computing concept covers: A preliminary report. In: *International Workshop on Description Logics (DL 2002). Toulouse, France.* (2002)
15. Hacid, M., Leger, A., Rey, C., Toumani, F.: Dynamic discovery of e-services: A description logics based approach. In: *Proceedings of the 18th French conference on advanced databases (BDA), Paris*. (2002) 21–25
16. Hacid, M.S., Léger, A., Rey, C., Toumani, F.: Dynamic Discovery of E-services. A Description Logics Based Approach. Technical report, LIMOS, Clermont-Ferrand (2002)