Combining RDF and Agent-Based Architectures for Semantic Interoperability in Digital Libraries

Norbert Fuhr, Claus-Peter Klas
University of Dortmund, Germany
{fuhr,klas}@ls6.cs.uni-dortmund.de

1 Introduction

From a functional point of view, digital libraries can be viewed as a collection of services. Today, already a large number of digital libraries is available on the Web; thus, federated digital libraries aim at the horizontal integration of these services. Whereas most of these digital libraries perform similar services on different datasets, we expect also a larger variety of services related to digital libraries in the near future. Examples for these services are e.g. alerting/filtering services, personal library services (storing a user's personal collection of digital library services, along with annotations), groupware services allowing for sharing personal libraries, recommendation services suggesting new items to a user based on his current profile.

Due to the heterogeneity of digital libraries, there also is a need for services bridging the semantic heterogeneity gaps, e.g. between different DL schemas (e.g. Dublin Core, MARC), different classification schemes or ontologies (e.g. Library of Congress Classif., Dewey Decimal Classif., DMOZ's Open Directory), different languages or even different media (e.g. text-based search on images).

There is a number of popular approaches towards interoperability that are targeted at achieving syntactic interoperability, i.e. protocols and data formats (e.g. Z39.50, DASL, OAI protocol). However, only a small number of services (searching, browsing, gathering, document delivery) has been considered so far. On a more general level, standardization efforts for Web services like SOAP and WSDL also address syntactic interoperability.

In this paper, we address the issue of semantic interoperability. We assume that we have a DL environment with a a large collection of services, for which a solution at the syntactic level has been chosen. For this setting, we present an approach for dealing with semantic interoperability that uses an agent-based architecture. Especially, we address the issue of semantic heterogeneity, and we consider the quality losses while mapping between heterogeneous interfaces.

The major ideas of our approach are a three-phase selection process and the usage of RDF schema[1] (RDFS) as description scheme for services. Below, we focus on the phases of the selection process and describe the role of RDFS in this process.

[1] http://www.w3.org/TR/rdf-schema/

The general setting is as follows: In order to satisfy a user's information need, an agent formulates a request for a specific type of service (e.g. search for documents fulfilling the given query, browse issues of a given journal, find related documents to a given document). There is a number of agents that may be able to perform this service. For selecting the agent(s) that is/are invoked for answering the request, a three-phase process is invoked:

1. matchmaking: identify candidate services,
2. planning: bridge semantic heterogeneity,
3. contract networks: select best services.

We describe each of these phases in more detail below.

2 1st phase: Matchmaking.

There is a central agent registry where agents have to register their capability. Capabilities of an agent are described in RDFS. For each type of service, we assume that there is a resource representing the service type, as well as resources representing the input and output types of this service type (see the example in figure 1). In order to describe the capabilities of specific services, input and output types have a number of properties. Each of these properties points to concepts of the the corresponding ontology. Thus, a subclass relationship on services is induced by the different values of the properties of the input and output types. For each of the output types, there also may be a small ontology, e.g. for describing subject areas or for characterizing different XML document types or dialects of schemas (MARC, DC).

In addition, there may be a specialization in the service functionality that leads to a subclass relationship (e.g. a service 'find related documents' may be specialized into 'find cited/citing documents' and 'find similar documents').

In order to register a specific service, first the appropriate subclass of the service resource has to be added to the description scheme. Next, the service is registered as an instance of this subclass.

When an an agent (called manager in the following) asks for other agents for performing a certain task, then this task has to be described wrt. to the description scheme used by the registration service. Assuming that all properties have been specified[2], then we have to look for services that subsume the current task. Logically speaking, a task submitted by an agent is transformed into a query wrt. to the RDFS structure (e.g. formulated in RQL [Christophides et al. 00]) that searches for instances implying the task description. In principle, also partial solutions should be considered (e.g. a search for documents published between 1980 and 2000 can be answered by a DL containing documents from 1970-90), but the limited expressiveness of RDFS as well as the complexity of the decision procedure prohibit a practical solution for this problem.

As a result of the matchmaking process, the registry's answer to the manager consist of a list of addresses of agents (along with their capabilities) that can perform the requested task. In turn, the manager sends the request to these agents.

[2] For unspecified properties, the most specific concept $\perp$ subsumed by all other concepts has to be assigned.

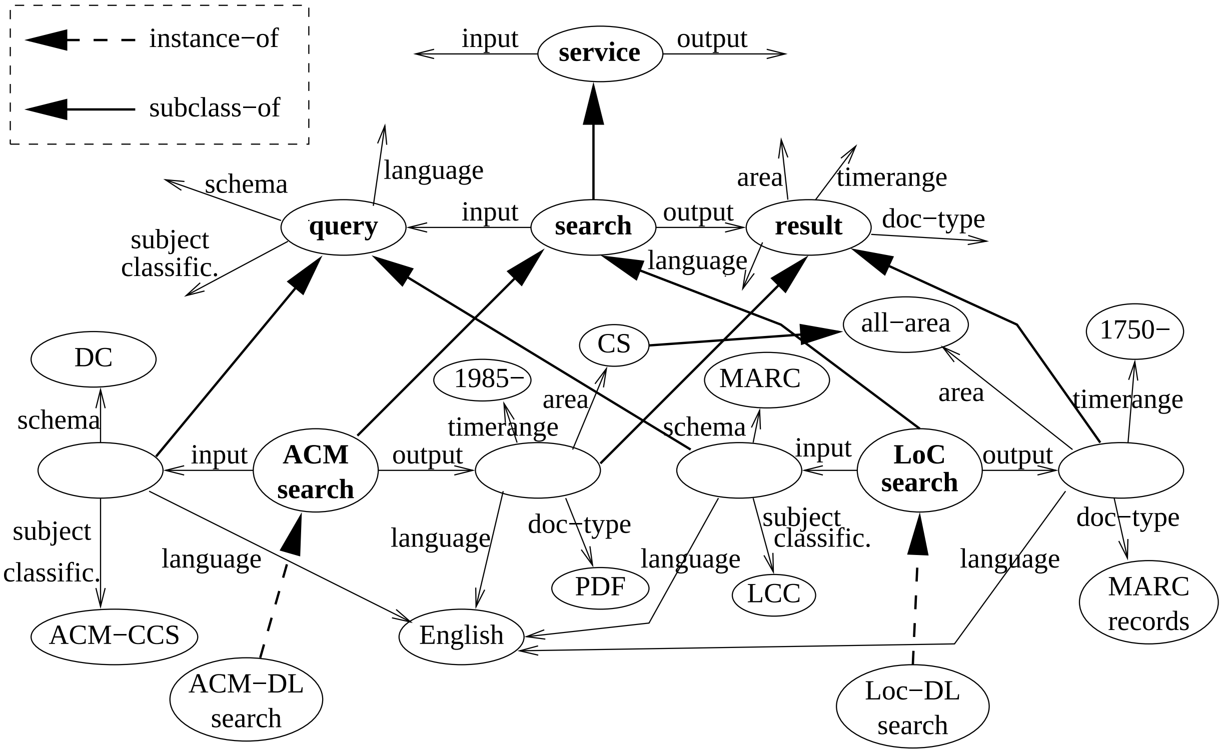

Figure 1: RDFS example of search service description

3 2nd phase: Planning.

Once an agent (called contractor in the following) receives a request, it is the task of the planning component to deal with semantic heterogeneity. That is, the properties of the request may not match the internal capabilities of the agent or of those agents that are further invoked for performing the task (e.g. the query may be in German using Dublin Core and ask for German output, whereas the agent only understands English using MARC and returns English documents).

For resolving the heterogeneity problem, there are other agents specialized in performing the corresponding mappings (e.g. language translation, schema mapping, transformation between subject classification schemes). For planning the invocation of these mapping agents, different organization schemes are possible:

1. Local planning: Each service has a powerful planning component that plans the transformation between the properties of the request and its own internal capabilities.
2. Global planning: There is a mediator agent for each type of service that performs the planning for all agents offering this type of service.
3. Distributed planning: There can be an arbitrary number of mediating agents for a type of service, each being able to resolve certain heterogeneity problems, and knowing a certain number of agents for this type of service.

Each of these strategies has its advantages and disadvantages, e.g. in terms of overall effort, scalability, reliability, or update behavior. In any case, the planning component must be able to

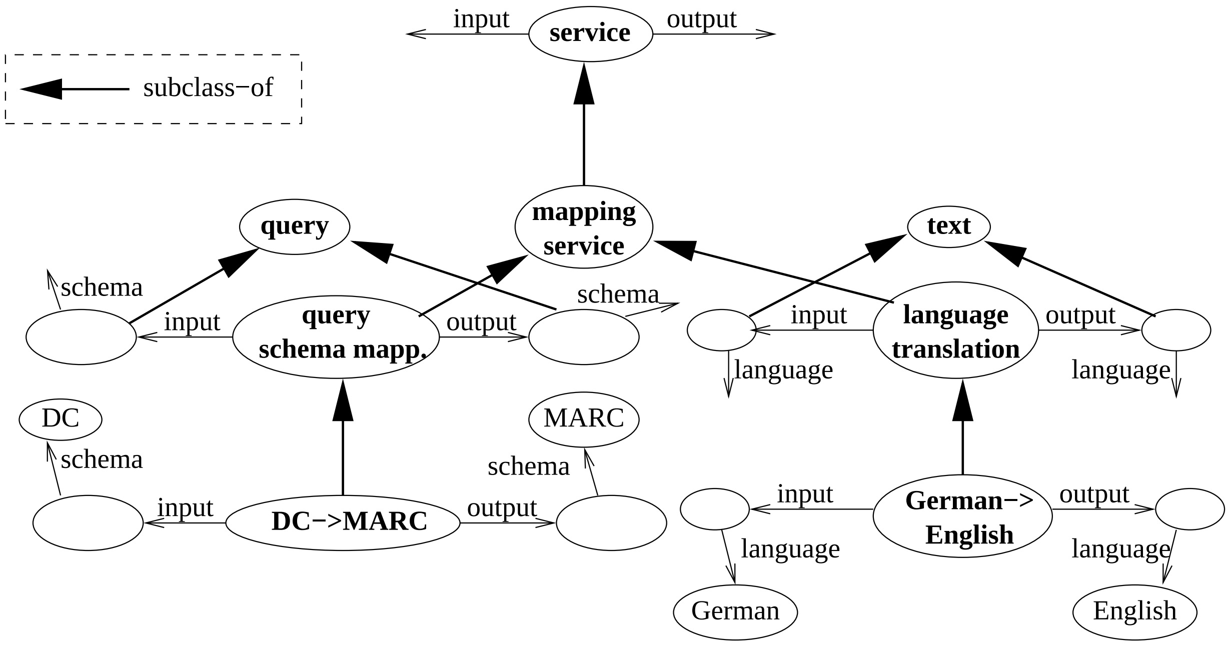

Figure 2: RDFS example: description of mappings services

find the appropriate mapping services, for which again matchmaking can be used.

As major problem with semantic heterogeneity, any mapping involved may be imprecise and thus lead to quality loss. For example, cross-language information retrieval yields worse performance than monolingual retrieval, and mappings between different classification schemes or bibliographic schemas usually cannot be precise (see [Nottelmann & Fuhr 01]). Thus, even if a plan for invoking a service with a rather uncommon interface exists, the resulting quality may be so poor that it will be employed only in cases when no alternatives exist. This quality issue is addressed in the third phase.

4 3rd phase: Contract networks

For a task specified by an agent acting as manager, any agent (contractor) who is willing to fulfill this task replies with a bid for this task. Then the manager selects one or more contractors with the best bids and establishes a contract with these agents. For applying this concept, the planning component (in any of the 3 organization schemes mentioned above) first has to acquire bids from the mapping services as well as from the agents performing the actual service. By adding its own effort, the agent can finally make a bid to the original manager. In principle, a bid is either a pair or a list of the pairs (result quality, cost). Given the bids from the different contractors, the manager makes his final choice (see e.g. [Fuhr 99] for an optimum selection strategy under certain constraints) and assigns the task to one or more agents.

5 Conclusions and outlook

In this paper, we have outlined a three-phase selection process for dealing with semantic heterogeneity. In contrast to standard database approaches, we also allow for uncertain mappings between heterogeneous services, and we consider the tradeoff between quality and effort. The procedure proposed may seem to be too complex for real applications; however, we view it as a starting point that makes explicit the different kinds of decisions involved. Of course, different kinds of shortcuts may be used in practical implementations.

We are planning to implement this approach in our agent-based DL system DAFFODIL[3], a system for accessing a federation of digital libraries and services in the area of computer science [Fuhr et al. 00]. The system integrates more than a dozen of digital libraries with quite different capabilities and also provides a fairly large number of different service types.

References

Christophides, V.; Plexousakis, D.; Karvounarakis, G.; Alexaki, . S. (2000). Declarative Languages for Querying Portal Catalogs. In: *DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries*, pages 115–120.

Fuhr, N. (1999). A Decision-Theoretic Approach to Database Selection in Networked IR. *ACM Transactions on Information Systems 17(3)*, pages 229–249.

Fuhr, N.; Gövert, N.; Klas, C.-P. (2000). An Agent-Based Architecture for Supporting High-Level Search Activities in Digital Libraries. In:*Proceedings 3rd International Conference of Asian Digital Library*, pages 247-254. KAIST, Taejon, Korea.

Nottelmann, H.; Fuhr, N. (2001). Learning probabilistic Datalog rules for information classification and transformation. In: *Proceedings CIKM 2001*. (to appear).

[3]`http://www.daffodil.de`