# Seamless Access to Databases through KQML in an Agent-enriched Web

Lara Cogoi
*Dept. of Mathematics and Computer Science, University of Udine, Italy*
*cogoi@dimi.uniud.it*

Vincenzo Della Mea
*Inst. of Pathology, University of Udine, Italy*
*dellamea@dimi.uniud.it*

Luca Di Gaspero
*Dept. of Mathematics and Computer Science, University of Udine, Italy*
*digasper@dimi.uniud.it*

Vito Roberto
*Dept. of Mathematics and Computer Science, University of Udine, Italy*
*v.roberto@dimi.uniud.it*

## Abstract

*The present paper illustrates a means for integrating the World Wide Web to generic informations sources - mainly databases- through agents.*

*We propose a mediator layer that lets databases facing on the Web be accessed through KQML, with two aims. The first is allowing the manipulation of queries and results by other agents; the second is giving a uniform external appearance to databases and other sources of information. In particular, our description focusses on identifying a set of KQML performatives suitable for information query and manipulation, and proposing an SQL semantic equivalence for them.*

## 1. Introduction

As the World Wide Web grows up, the fraction of hypermedia generated from databases increases, due to the much easier way of producing data in a structured form as can be done by means of DBMS s; the latter are quite often parts of the corporate information systems. This occurs as an automated, dynamic hypermedia construction from database records (e.g. through server scripting) or by letting the user query the database through HTML forms. In such a situation, the problem of integrating multiple remote databases and/or different sources of information is apparent.

The multi-agent paradigm has recently gained public attention, as a means for analysing and implementing complex systems made by heterogeneous dynamic components, often relying upon distributed resources. In such an approach, an agent is an entity having the properties of autonomy, sociality, reactivity and proactivity; each agent is able to communicate with peers in an appropriate language, to fulfill its design aims by taking into account the environment [1]. One of the major areas in which agents have been found useful is that of interoperability to legacy software and data, including existing databases [2]. Another important area is that of information discovery on the Web, to help users in finding sources and screening data. However, both areas are not devoid of problems: legacy systems are not simple to agentify in an automated way, and information discovery must rely on some standards to be effective [3].

Many approaches to agents are based on KQML (Knowledge Query and Manipulation Language) [4], which allows communication grounded on the Speech Acts theory, i.e., by using statements called perfomatives as content carriers. This paper proposes a mediator layer that lets databases facing on the Web be accessed through KQML, with two main aims. The former is allowing the manipulation of queries and results by other agents; the latter is giving a uniform external appearance to databases and other sources of information.

Access to SQL databases through KQML had already been proposed, because it is of great interest in the multi-agent software engineering community. In particular, the CoBase system provides for KQML access to databases, by means of a specific content language [5] called CCL (CoBase Content Language), which is in turn based on a specific ontology (CO, Cobase Ontology). This way, KQML acts as a transport language for intercommunicating entities, and its expressive power is not fully exploited.

MOO-KQML is just a project meant at extending the capabilities of MOO-based collaboratories [6]. It allows to query an SQL database through a specific KQML performative, to which an SQL meaning is assigned. Our proposal is grounded on the latter approach, because it allows to exploit directly the performatives already available for database management, while masking database details.

Our description focusses on identifying a set of KQML performatives suitable for information query and manipulation, and proposing an SQL semantic equivalence for them.

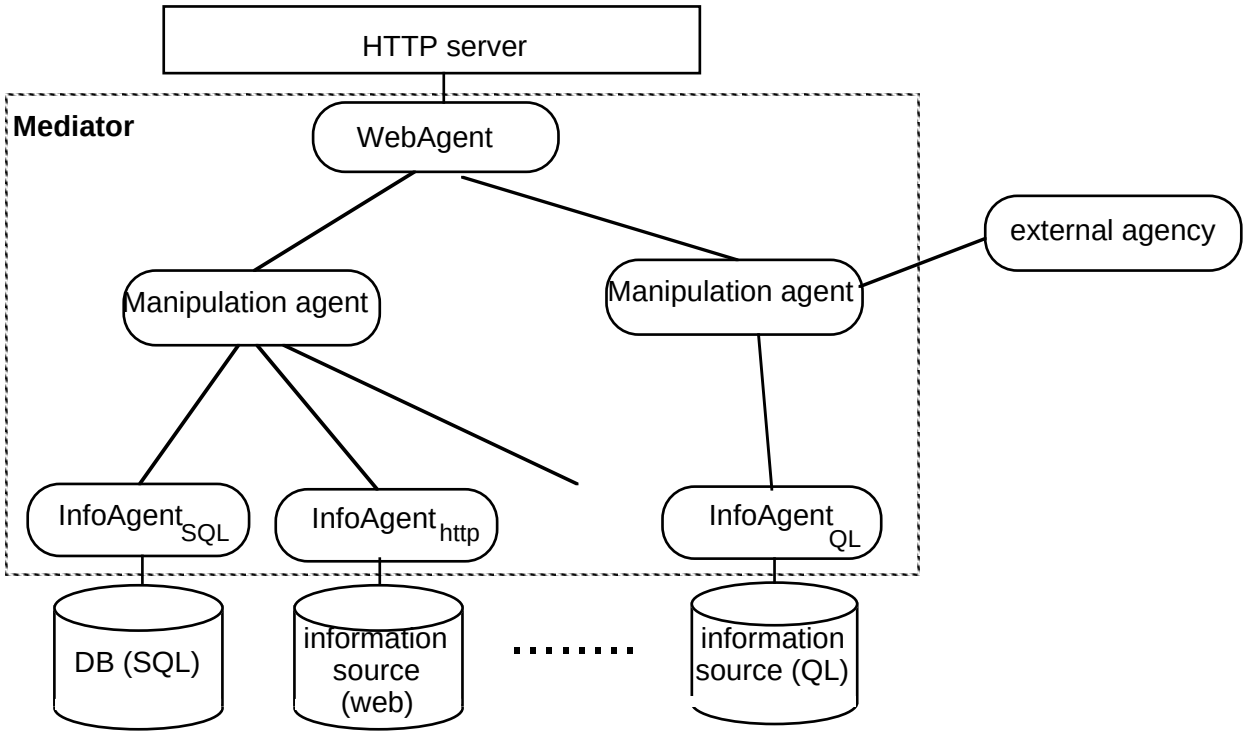


**Figure 1. The mediator layer architecture**

## 2. Mediator agents

In the proposed approach we establish a mediator layer between the http server and information sources. This layer deals with the translation of http requests into KQML messages, and the conversion of agent responses from KQML to HTML; in addition, both kinds of data may be manipulated inside the layer. In order to make it possible, a common query/manipulation language should be adopted on the information sources side.

Figure 1 shows a sketch of the mediator layer architecture. There are a number of agents involved, two of which are mandatory (the WebAgent and at least one InfoAgent), whilst the others allow for optional processing of queries and results.

$Infoagent_{QL}$ is the specific facilitator towards the underlying information source accessible through a query language QL: SQL databases will have a specific $InfoAgent_{SQL}$, web resources will be mediated by means of $InfoAgent_{http}$, etc. Besides such facilitators, any information source becomes accessible through a common language, thus masking their implementation differences.

Between the Web- and InfoAgents there can be manipulation agents, which are able to transform either queries or results, with different aims, e.g., to distribute the query to multiple information sources, even not of the same type. Manipulation agents may know the common query/manipulation language, thus being able to opoerate transparently on different sources. Furthermore, they are in charge of handling different ontological representations among the InfoAgents.

In the rest of the paper, we will focus on InfoAgents and their behaviour.

## 3. SQL and KQML

Despite of the name, SQL may be used as a query language as well as a data definition and manipulation language. On the other hand, KQML is a content-independent communication language made by expressive performatives, whose semantics has been given in terms of the Speech Act theory [4]. Some performatives have a natural translation as database management instructions (i.e., insertion and retrieval). Being SQL a widely used query language, which already addresses the main needs of data query and manipulation, we started from it to identify the main features of the generic InfoAgent, and in particular to define its language and protocol based on KQML.

The main idea is to give an SQL equivalence to some suitable performatives. In order to do this, four steps have been identified:

i) establishing a terminological correspondence between SQL and KQML;
ii) identifying the KQML performatives whose informal semantics is close to SQL operations;
iii) identifying the performative parameters to which SQL clauses may be associated; and
iv) describing the SQL equivalence of the selected performatives.

The first step, common to all operations, brought to associate the ontology concept to the table term, with the table columns corresponding to the ontology dictionary and the tuples to assertions.
Other steps are specific of each operation, and thus are to be explained separately for insertion, deletion, update and query.

### 3.1. Insertion

The INSERT performative seems naturally adequate for the insertion of tuples into a database, thus it can be associated to the INSERT SQL clause.
The :content parameter may become the bearer of the new information to be inserted, and thus may be associated to the VALUES SQL instruction. The :ontology parameter, which should specify the reference ontology for the data enclosed in :content, is then used to indicate the table to be updated.
This results in the following performative:

INSERT
:content *(listof (= (field field1) (value v1))*
*...(= (field fieldN ) (value vN)))*
:language *KIF*
:ontology *TableName*
:reply-with *IdSenderAgent*
:in-reply-to *may-be-void*
:force *tentative*
:sender *SenderAgent*
:receiver $InfoAgent_{SQL}$

Please notice that the value "KIF" of the :language parameter is only one of the possible content languages, and basing on it the :content parameter has been filled in.
The performative is translated into the following SQL operation:

INSERT
INTO *TableName*
*(field1, field2, ... fieldN)*
VALUES *(v1, v2, ... vN)*

### 3.2. Deletion

KQML defines three different performatives aimed at asking an agent to delete phrases from its virtual knowledge base: DELETE, DELETE-ONE, DELETE-ALL. Unfortunately, the :content parameter should contain a KQML performative and not other kinds of information. To overcome this problem, a new performative has been defined, which allows to ask the removal of all sentences that satisfy the conditions expressed in the :content parameter. Deleted sentences are returned back, following the requests contained in the :aspect parameter.

This results in the following performative:

DELETE-SENTENCE
:content *(condizione)*
:aspect *t-uple*
:language *KIF*
:ontology *TableName*
:reply-with *IdSenderAgent*
:in-reply-to *may-be-void*
:sender *SenderAgent*
:receiver $InfoAgent_{SQL}$

where the condition in the :content parameter is a generic condition expressed in the selected language (e.g., KIF). Such performative may be translated as:

DELETE
FROM *TableName*
WHERE *condition*

### 3.3. Update

Since there are no KQML performatives explicitly devoted to the modification of a virtual knowledge base,

we extended the scope of the INSERT performative in order to accomplish also update tasks. This makes sense because the logical difference between insertion and update is limited, being both operations related to a modification of the virtual knowledge base.

Thus, a performative with meaning associated to the UPDATE SQL clause is as follows:

```
INSERT
  :content (if conditions
             (listof (= (field field1)(value v1))
             ...(= (field fieldN)(value vN))) bottom)
  :language KIF
  :ontology TableName
  :force tentative
  :reply-with IdSenderAgent
  :in-reply-to may-be-void
  :sender SenderAgent
  :receiver InfoAgent_SQL
```

where the KIF *conditions* expression in the :content parameter is a sort of conditional clause, to which the update is subjected, as may be easily understood looking at the corresponding SQL translation:

```
UPDATE TableName
       SET field1= v1, ... fieldN=vN
       WHERE conditions
```

### 3.4. Data query

For implementing data query, the chosen performative is ASK-ALL, which serves to retrieve all assertions satisfying the conditions inside the :content parameter, in the form specified by the :aspect parameter. In addition to this, the :ontology parameter may be used to specify the tables to be searched. Thus, a performative of this form:

```
ASK-ALL
      :content conditions
      :aspect (listof field1 ... fieldK)
      :language KIF
      :ontology tables
      :reply-with IdSenderAgent
      :receiver InfoAgent_SQL
```

can be translated into the following SQL operation:

```
SELECT field1 ... fieldK
      FROM tables
      WHERE conditions
```

A prototype system implementing the proposed approach has been developed and experimented in the framework of an ongoing project for the authoring and publishing of multimedia medical cases [7]. All the proposed agents were developed using JAMES [8]: a Java implementation of a KQML speaking agent model.

## 4. Discussion

The agent approach in the integration of heterogeneous information sources has been proposed by Genesereth et al. since 1995 [9,10]; a system called Infomaster was presented, that creates a virtual data warehouse accessing distributed heterogeneous information sources on the Internet. The Infomaster architecture consists of a single centralized facilitator that deals with requests coming from the WWW as well as from other agents, and dispatches them to a specific information source wrapper (similar to our InfoAgents). Their main effort was in the harmonization of different information sources that has been achieved through maps of data convertions. The main difference with respect to our approach is that we provide an additional level between the facilitator and the wrappers that allow further data manipulations. This level may be used for the translation among the (generally) different ontological representations of the specific InfoAgent. As recognized by Hwana and Ndumu [3], however, the problem of automatically mapping different ontological representations still remains central, not only in our approach but also in the whole agent community, together with other ontology issues.

The choice of KQML as an access language for information sources is related to the likely growing diffusion of agents inside the WWW environment. By deploying the proposed architecture, the user interface remains based on current web browsers, without introducing any new software to the user. However, such an approach will perhaps give a computational overhead that is worth to be studied. An agent that accepts the KQML performatives as defined is not so different from any other agent, because the underlying database is quite similar to the usual virtual knowledge base. Further work is needed for developing other InfoAgents and exploring thus the effective capabilities of the proposed approach.

## 5. References


[1] M.J. Wooldridge and N.J.Jennings, "Intelligent agents: theory and practice", *Knowledge Engineering Review* 10, 1995, pp. 115-152.

[2] N.J.Jennings and M.J.Woolridge, *"Agent Technology Foundations, Applications, and Markets"*, Springer-Verlag, Berlin, 1998.

[3] H.S.Nwana and D.T.Ndumu, "A perspective on Software Agents research", *Knowledge Engineering Review*, 14, 1999, pp 1-18.

[4] T.Finin and Y.Labrou, "A semantics approach for KQML - a general purpose communication language for software agents", In Proc. of CIKM94, 1994.
[5] W.W.Chu, H.Yang, K.Chiang, M.Minock, G.Chow, and C.Larson, "CoBase: A Scalable and Extensible Cooperative Information System", *Journal of Intelligence Information Systems*, 6, 1996.
[6] D.Seabrook and J.Messina, "Project MOO-KQML-Database", URL:http://enterprise.aacc.cc.md.us/~rhs/report.html, 1997.
[7] V.Della Mea, V.Roberto, A.Conti, L.Di Gaspero, and C.A.Beltrami, "Internet Agents for Telemedicine Services", *Medical Informatics & Internet in Medicine (formerly Medical Informatics)* 24, 1999, pp. 179-186.
[8] A. Conti, V. Della Mea, L. Di Gaspero, V. Roberto, "MANTHA: Agent Based Management of Hypermedia Documents", in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems* (ICMCS99). Firenze (Italy), June 1999.
[9] D.F. Geddis, M.R. Genesereth, A.M. Keller and N.P. Singh, "Infomaster: A Virtual Information System", *Intelligent Information Agents Workshop at CIKM'95*, December 1995.
[10] M.R. Genesereth, A.M. Keller and O.M. Dushka, "Infomaster: An Information Integration System", in *Proceedings of the ACM SIGMOD Conference*, May 1997.