Ontology for Database Access

Ryusuke Masuoka
Fumihiro Maruyama
{masuoka, superb}@flab.fujitsu.co.jp
NetMedia Laboratory
Fujitsu Laboratories Ltd.

October 25, 1996

1 Abstract

We are applying ACL (Agent Communication Language) agent technologies to practical domains, such as virtual marketplace for distribution industry, access to online databases, and travel planning. In these domains, ontologies to make accesses to disparate databases possible are quintessential, and we are developing one. In this paper, we will describe design criteria which we have found important, with several examples.

2 Introduction

We are developing prototypes of virtual marketplace for distribution industry and of accessing system for online databases under the project "SAGE (Smart AGent Environment)." The project SAGE is aimed at creating an environment for agents to live and communicate and the environment is based on ACL.

In the prototype of virtual marketplace, we will enable retailers and wholesalers to make electronic trancsactions between them. We also plan to realize functionality of market in SAGE.

In the prototype of accessing system for online databases, we will make it possible for users to access to the information in the online databases without knowing gory details of accessing nor which database he/she is accessing.

In both domains, ontologies to make accesses to disparate databases possible are quintessential. A virtual marketplace system has databases on both retailers' and wholesalers' sides. Online databases are of course databases themselves. We are developing a generic ontology for database access to incorporate into both of our prototypes.

In the process, which we are not through yet, of building one, we found criteria for that kind of ontology. Though they are still preliminary, we report those design criteria which we have found important, with several examples in this paper.

2.1 Standardization Processes in Japan and Ontologies

Before we go into the details of criteria, we will touch the subject of standardization processes in Japan and ontologies.

Once Japanese companies used to be slow to adopt industry standards. Now with realization of Electronic Commerce (EC) and Commerce At Light Speed (CALS) in sight, a number of standardization processes are under way in many areas in the industries.

Many of the companies are also introducing data warehouses for their Decision Support System (DSS). A data warehouse is a completely centralized system, opposite to distributed systems like an agent system. But the standardization of units and words, which is strongly recommended in data warehouse systems, is actually preparing the people in those companies to introduce ACL-type agent systems.

We believe developing ontologies strongly depends on efforts from those companies involved on such standardization processes. Only those who are involved in the area can create appropriate models, concepts, and words, on which we can safely build the ontologies. Therefore what we see is that they are quite ready to help us build ontologies and to introduce agent systems.

3 Design Criteria

The followings are design criteria for ontologies for database access.

In designing it, we pursued coherence and other criteria as mentioned in [4] as much as possible, but we put great emphasis on practicality at the same time. The system will be useless with drawbacks such as long latency, even if we could provide totally new functionalities by using agent technologies.

3.1 Model

We model the database as a set of records. A "record" is something that can be asked of values for specified fields. The model of set is used because the set operations between the subsets of a database are necessities, especially for online database access. We considered the possibility of using lists instead of sets, but we judged the list structure is not essential for database.

The following is an expression in KIF, which defines the class of "databases" as mentioned above.

```
(defrelation database (?db) :=
    (and
        (set ?db)
        (exists (?set-of-fields)
            (and
                (set ?set-of-fields)
                (forall (?record ?field)
                    (=>
                        (member ?record ?db)
                        (member ?field ?set-of-fields)
                        (defined (field-value ?record ?field)))))
         )
    )
)
```

In the above definition, "field-value" is a function. We also regard this "field-value" as a relation in a usual manner (c.f. Chapter 8 in [2]).

3.2 Restrictions to Declarative Language

With a declarative language such as KIF (c.f. [2]), one can easily come up with a problem description that can not be solved even theoretically. A problem is practically unsolvable if the problem can not be solved in the reasonable amount of time, These are the situations which we do not want for the practicality of the system. But at the same time, we do not want to lose the expressiveness of KIF.

We have to balance these demands. In order to do so, we need reasonable restrictions to the expressiveness of KIF. The followings are the candidates for such restrictions.

- Use advertise performative in KQML (c.f. [1]) to restrict acceptable expressions for queries to the agent.
- Use user interfaces and restrict the expression of the queries user interfaces dispatch.
- Syntactic restrictions to KIF, such as to limit the kind of relations used in the queries.

We are currently taking the second approach, that is to limit the expression of KIF by user interfaces. The following is an example query sent from a user interface to ask the list of providers and prices for lettuces which weigh 500 g or more and which cost less than 1000 yen for eight of them. [1]

[1] This message follows the rules described at the end of Section 3.5.

```
(ask-all
:aspect (?provider ?price)
:content (and
              (is-a ?x agricultural-product)
              (provider-name ?x ?provider)
              (price ?x ?price)
              (product-name ?x ``lettuce´´)
              (>= (weight ?x) 500|g)
              (< (* ?price 8) 1000|yen)
          )
)
```

"price" relation in the above, for example, specifies the value for ":price" field of the record. This is defined by the following.

```
(defrelation price (?x ?price) :=
    (field-value ?x :price ?price))
```

We used english words in the above example for easy understanding. Japanese words are actually used as mentioned in 3.4.

3.3 Easy Implementation

We believe that database agents must be easily implemented. Since we apply these technologies to the practical areas, we can not expect the system constructors to be experts in the area of knowledge sharing research. We need to restrict ontologies to fit the capability of the system constructors. [2]

The restrictions mentioned in the previous subsection may apply. We can also keep the database agents simple by making the facilitator do the part of "solving" the problem into simples ones for the database agents.

3.4 Native Language

The native language should be employed where it is appropriate. In our case, we allow using Japanese words to express the concepts.

This is because there will be much less mistakes in creating the systems if one can use the language of his/her familiarity. It is also true that there are concepts that have appropriate words for only in one language and not others. We believe this will lead to faster deployment of more robust system.

The language coding scheme should be passed on in each KQML message.

[2] The situation may change if we can provide easy-to-setup tools to make legacy databases into agents.

3.5 Rule-based Advertisement

In our system, database agents will advertise their capabilities to the facilitators. Facilitators forward queries to appropriate database agents, based on the knowledge of their capabilities.

Advertising the format itself of the query is too restrictive. We decided that a database agent should advertise what kind of knowledge they have and the rules (or the name of a set of rules) which decide the query is acceptible for the agent or not. Therefore we are limiting the syntax of KIF used in queries to database agents.

For the knowledge part, the database agent essentially tells the facilitator the field names of the record, and what kind of objects the values for the fields are. With the latter (the kind of objects), agents and facilitators can agree on what legal expressions of the values for the field are. This adds more robustness to the system. There are also included natural language descriptions for the fields.

For the rule part, the database agent advertise the rules which can decide which query is a legitimate one for the agent. Or we might make the database agents and the facilitators share a set of rules, and the database agents advertise with a relational sentence that they follow that set of rules.

We took the latter approach. "allows-database-query" in the following example is the relation which tells that the agent follows the rules in the list after the example. The example is an advertisement message for a database agent named "xyz-foods," which is sent from a database agent to a facilitator.

```
(database xyz-foods)

(=> (member ?x xyz-foods)
    (is-a ?x agricultural-product))

(field-definition xyz-foods :JAN-code
    (quote is-text) "JAN (Japan Article Number) code")
(field-definition xyz-foods :price
    (quote is-number-currency)  "price per unit")
(field-definition xyz-foods :color
    (quote is-color)  "color of the product")
...

(allows-database-query xyz-foods)
```

By sending this message, both the agent and the facilitator agree that the agent can handle queries with relations which abide by the following rules.

- Relation of fields and their values
- Arithmetical relation between quantities

- Any logical combination of relational sentences above

4 Summary

In this paper, we gave the design criteria for ontologies for database access which we got in the process of designing our ontology for database access.

We are still working on prototypes and the ontology, and the details are subject to many changes. We believe that we will be able to give more details at the time of the symposium.

References

[1] Tom Finin: "Specification of the KQML Agent-Communication Language", 1994/2, `http://www-ksl.stanford.edu/knowledge-sharing/papers/README.html#kqml-spec`

[2] Michael R. Genesereth and Richard E. Fikes: "Knowledge Interchange Format Version 3.0 Reference Manual", 1992/6, `http://www-ksl.stanford.edu/knowledge-sharing/papers/README.html#kif`

[3] Guy Steele, "Common LISP", 1990, Digital Press.

[4] Thomas R. Gruber: "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", August, 1993.