An Architecture for Heterogenous Federated Mediators

Dong Cheng and Nacer Boudjlida

LORIA-University Henri Poincaré Nancy 1
BP 239, 54506 Vandoeuvre Lès Nancy CEDEX(F)
Email: cheng,nacer@loria.fr

Abstract. This paper describes a heterogeneous federated mediators architecture to support querying and cooperation activities. In a lot of situations, the data and program were recorded by different persons, at different times, in different languages, on different fields. All the difference makes the information systems' heterogeneity. The mediation services are expected to play an important role in helping automated processes to access heterogeneous information. They are not Yes/No answers from the mediator, when no single object meets the search criteria, they may be cooperative answers to make a composite answer. In this paper we introduce a mediator-based logical architecture and its implementation.

Keywords: heterogeneous system, mediation, knowledge representation

1 Introduction

The propagation of the network has led to increasing need for interoperability between heterogeneous systems and services. These latest developments in data network domain are new challenges for interoperability technology, as attested by the tremendous work about Semantic Web [1]. However, in this work we intend to define an approach where heterogeneous systems may work together in a context of semantic queries. In this condition, we must match the information that was created by different programmers, at different times, in different languages, and all these things may be described in different knowledge representation technologies. Examples of applications of our work may be searching for Web Services that offer given functions, searching for a component in the context of component-based design and component-based programming, searching for a business partner with a given expertise, looking for an employee whose records and expertise satisfy a given position profile...

In this paper, we highlight a Composite Answer approach in heterogenous environment where systems maybe record the information in different knowledge representation technologies. A significant originality of our approach resides in the type of answers we aim at providing. Indeed, when there is no unique entity to satisfy the search criteria, the systems attempt to determine a composite
answer that satisfy the criteria. Possibly heterogeneous systems may cooperate in the query evaluation process and in the answer composition process. These systems are viewed as a federation.

We are exploring alternative architecture, component and mechanisms to transparently federate such types of systems or services. This paper reports on a first experimentation focusing on the architecture itself. The presentation is structured as follows. In section 2 we briefly review the conceptual federated mediator-based architecture, and the concepts of composite answers in this architecture. In section 3, we describe an implementation and the physical architecture. Conclusions and remarks are in section 4.

2 A Review of Mediation architecture

In our work a mediator-based architecture has been adopted and is described in [7]. It is a dynamic discovery of services or capabilities an “entity” offers. It is very similar to the notion of discovery agency in the Web services architecture. In this architecture, an “entity”, called exporter, publishes (tells) its capabilities at one or more mediators sites (see figure 1). Entities, called importers, send requests (queries) to the mediator asking for exporters fitted with a given set of capabilities.

Several approaches, which are mediator-based, i.e. they are distributed and intelligent production systems, have been proposed over the last decade. A single Mediator is designed to offer an adequate level of decision-making integration, it takes into account the effort needed for the integration of heterogeneous computer systems [8]. The Conflict Resolution Environment for Autonomous Mediation (CREAM) system has been implemented, it provides various user groups with an integrated and collaborative facility to achieve semantic interoperability among participating heterogeneous information sources [9]. The KRAFT (Knowledge Reuse And Fusion/Transformation) architecture provides a generic infrastructure for knowledge management applications. It supports virtual organization using mediator agents [10]. When it was applied to business-to-business electronic commerce, the mediators allows partners to exchange rich business information. As online travel agent application, a nature of competition in electronic markets, when it gets a customer’s travel requirements to reservation system, envision this hypothetical system designed for a group of airlines all of whom serve national routes. Consider what might occur if that system was extended to include international airlines and foreign routes. This travel requirements could be easily satisfied by one travel agent in federated mediators.

In federated mediators, a cooperation environment, we can see the query as being addressed to “the union” of the federated mediators’ knowledge bases. Concretely, this union is explored from “near to near” within the federation, that means from a mediator to an other. Satisfying the query falls into different cases [7]:

- Case1: There exist exporters that exactly satisfy the query;
Case 2: There exist exporters that fully satisfy the query, but their capabilities are wider than those requested;

Case 3: No single exporter fully satisfies the query, but when “combining” or composing capabilities from different exporters, one can fully satisfy the query;

Case 4: Neither a single exporter nor multiple exporters satisfy the query, but there exist some exporters that partly satisfy the query;

Case 5: No single exporter nor several exporters fully or partly satisfy the query.

One notices, that cases 3 and 4 are cases where a composite answer is returned. In the case 4, we have to determine “what is missing?”, Complement Concept, to the individuals to satisfy Q, which means to determine what part of the query is not satisfied by the found individuals. Furthermore, in these cases 2, 3 and 4, we need to notice “what is superfluous?”.

An approach to find out “the best” answer was proposed using Description Logics [11]. The determination of the complement is based on the subsumption relationship using a Normalize-Compare process. The implementation of this process uses an array of Boolean (called “Table_Of_Test” further) to record the results of the subsumption relationship evaluation. In the figure 2, $C_1, C_2, C_3, \ldots, C_n$ denote the query concept under its normal form and $D_1, D_2, D_3, \ldots, D_m$ denotes the concepts “known from” the mediators, i.e. every $D_j$ has to be viewed under its normal form $D_j^1, D_j^2, \ldots, D_j^m$.

Then $TableOfTest[D_j, C_i] = true$ means that $D_j \subseteq C_i$. When the value returned by the function $Subsumes(C, D_j)$ is “false” (i.e. the concept $D_j$ does not fully satisfy the concept $C$), therefore we need to determine a possible complement of $D_j$ relatively to $C$. Using the Table_Of_Test it is easy to get the complement of the concept $D_j$ relatively to the concept $C$: $Comp(C, D_j) = and_{k=1}^{m} C_k [TableOfTest[D_j, C_k] = false]$ \footnote{In the following, we will use $T[i, j]$ instead of TableOfTest[i, j]}. That means that the complement is given by the conjunction of all the atomic concepts for which the corresponding values in the “Table Of Test” are “false”.

The composition of the truth values determines the cases of satisfaction. Consider a table $ORoD[1..n]$ as $ORoD[i] = \bigvee_{j=1}^{m} T[D_j, C_i]$. $ORoD[i] = true$
\[\begin{array}{|c|c|c|c|}
\hline
C_1 & C_2 & \ldots & C_n \\
\hline
D_1 & \text{False} & \ldots & \text{True} \\
D_2 & \text{False} & \text{True} & \ldots & \text{True} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
D_m & \text{False} & \text{False} & \ldots & \text{False} \\
\hline
\text{ORoD} & \text{False} & \ldots & \text{True} & \text{True} & \text{False} \\
\hline
\end{array}\]

\textbf{Fig. 2. The “Table Of Test”}

means that the concept \(C_i\) is satisfied by at least a \(D_k\). If the conjunction of the values of \(\text{ORoD}\), noted \(\text{ANDoS}\), is true (i.e. \(\bigwedge_{i=1}^{n} \text{ORoD}[i] = \text{True}\)), it means that all the \(C_i\)s are satisfied and therefore the query. When \(\text{ANDoS}\) is false, the logical disjunction of the values of \(\text{ORoD}\), noted \(\text{ORoS}\), enables to determine a possible partial satisfaction: if \(\text{ORoS} = \bigvee_{i=1}^{n} \text{ORoD}[i] = \text{True}\), it means that there exist some \(C_k\) that are satisfied. If both \(\text{ORoS}\) and \(\text{ANDoS}\) are false then no atomic concept \(D_j^k (j \in 1..m)\) satisfies a \(C_i\).

For example, a formulaic query of travel requirement is in \textit{Description Logics}.

\[Q = (\text{and TRAVEL}) \quad C_1\]
\[(\text{all depart NANCY/FRANCE}) \quad C_2\]
\[(\text{all arrive WUHAN/CHINA}) \quad C_3\]
\[(\text{some step NANCY/FRANCE}) \quad C_4\]
\[(\text{some step WUHAN/CHINA}) \quad C_5\]

It exists a mediator system 1, it well knows all routes in France and some international airlines. So it may find a route as:

\[D_1 = (\text{and TRAVEL}) \quad D_1^1 \sqsubseteq C_1\]
\[(\text{all depart NANCY/FRANCE}) \quad D_2^1 \sqsubseteq C_2\]
\[(\text{all arrive BEIJING/CHINA}) \quad D_3^1 \not\sqsubseteq C\]
\[(\text{some step NANCY/FRANCE}) \quad D_4^1 \sqsubseteq C_4\]
\[(\text{some step PARIS/FRANCE}) \quad D_5^1 \not\sqsubseteq C\]
\[(\text{some step BEIJING/CHINA}) \quad D_6^1 \not\sqsubseteq C\]

System 1 does not know any route in China, this travel requirement can’t be fully satisfied. But we can find another mediator system 2, it well knows all routes in China. So it easy to find a route as:

\[D_2 = (\text{and TRAVEL}) \quad D_1^2 \sqsubseteq C_1\]
\[(\text{all depart BEIJING/CHINA}) \quad D_2^2 \not\sqsubseteq C\]
\[(\text{all arrive WUHAN/CHINA}) \quad D_3^2 \sqsubseteq C_3\]
\[(\text{some step BEIJING/CHINA}) \quad D_4^2 \not\sqsubseteq C\]
\[(\text{some step WUHAN/CHINA}) \quad D_6^2 \sqsubseteq C_5\]

We put the \(D_1\) and \(D_2\) into a \textit{Table Of Test}. 

```
<table>
<thead>
<tr>
<th>$D_1$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>$D_2$</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>

We get $ANDoS$=True, it means that this travel requirement can be fully satisfied by a composite answer of $D_1$ in system 1 and $D_2$ in system 2.

3 Implementation and Architecture

An experimental platform has been developed in Java where some services, like testing the subsomption relationship, determining the complement concept, and making a composite answer, have been implemented. As the drawing in figure 1, the federated mediators work in distributed and heterogeneous environment. The federated mediators architecture conforms to the Reference Model for Open Distributed Processing (RM-ODP). As shown in figure 3, the mediator architecture consists of three main parts: a local repository, a reasoning processor, a syntax translation.

![Diagram](image)

**Fig. 3.** The Layered Mediator Architecture

In this implementation, we accept the Description Logical language $AL-$ to reason and represent knowledge. It is used to represent the entity’s “capability” in the local repository, and Java objects of $AL-$ are used in all algorithms implementation. All services were implemented in a Description Logic reasoning processor, using the approaches and the algorithms that were outlined in the previous section.
3.1 The export sketch and inter-action

The exporter sends its capability description in $AC-$ to the Reasoning Processor, as showed in figure 4. In this situation, exporter accesses the Mediator server through a Web Server CGI program, such as "classical" Web client/server architecture. The servers accept the $AC-$ concepts written in DAML+OIL, an ontology language in XML [16]. The $AC-$ concepts are encoded in DAML+OIL, then this DAML+OIL text is transmitted to the Web Server’s URL. The CGI program may manage the export/import task. It uses the Syntax Translator to translate the DAML+OIL text into the $AC-$ concept object. Then it puts the concept object into the Reasoning Processor. The Reasoning Processor uses the Classification approach to add this concept into the knowledge hierarchy ($TBox$ and $ABox$) in the local repository.

![Fig. 4. The UML Sequence of Export Process](image)

3.2 The import sketch and inter-action

The Reasoning Processor processes the query input from Importer like the process of Exporter’s capability concept, as showed in figure 5. Then this normalized query concept is put into a $TBox$ object. The algorithms in last section are implemented in the $TBox$ class. The services’ result outputs are also encoded in DAML+OIL XML is also used to exchange information and concepts between the mediators when mediators’ cooperation is needed. In Composite Answer Situation Case 3 and 4, we need transmit the complement concept to the service partners. The Web CGI program make a new Import on complement concept to partner server, as we discussed in section 2. Moreover, we deliberately ignored the search of the actual individuals ($ABox$) that satisfy a query, i.e. in the current implementation, only $TBoxes$ are considered.
4 Conclusion Remarks

The federated mediators is a heterogenous environment, it means different hardware and software, different knowledge representation systems. Thanks to W3C's work, it is possible to communicate between heterogeneous hardware and software by the family of standard from W3C(as XML, SOAP, WSDL, OWL, Web Services, Semantic Web, etc.)

The Syntax Translator may translate the $\mathcal{AC}\$ concept object to multiple knowledge representation languages in different message format. Which language will be accepted by federated mediator server, DAML+OIL or OWL! Which communication standards will be accepted, SMTP or HTTP? etc. The translator may read the necessary information from mediator server's a description in WSDL.

Complete semantic interchange is a hard problem between heterogeneous knowledge representation systems. In this work, we did not address semantic translation between any knowledge representation language. The Syntax translator is a limited translator on semantic query, and only work at the syntax level. We see in last section, the Complement Concept is under a conjunctive form: $C = ( \text{and } C_1, C_2, \ldots, C_n)$. Each $C_i$ is an atomic concept where each atomic concept is identified by a term. The term is supposed to exist in a common lexical ontological dictionary (such as WordNet [13], EuroWordNet [4]). Moreover, we suppose facilities for mapping between terms. (Some term mapping approaches in lexical level are described [6]). In our work, heterogeneous systems cooperation thanks to the complement concept which is under the simplest form, conjunction. The conjunctive form is adopted in most knowledge representation languages,
so the complement concept may be understood by the other partner systems. So these systems may be in different knowledge representation languages.

From the experimental platform, we believe that heterogeneous systems may interoperate by virtue of composite answer approach. We believe that the hard problems do not go away even if we solve low-level issues such as natural language analysis, complex term mapping, and ontologies integration. Further, we suggest that the root causes can be understood better in terms of complement.

In the future, based on our initial experience on limited heterogeneous knowledge representation technology and common mathematical background of heterogeneous knowledge representation technology [21], two interrelated approaches paid attentions to our work. The first is to find some limited translation approaches between any two knowledge representation languages. The second is to measure the mismatch, then describe it in a common knowledge structure, and find the algorithm to get the “best” composite answer.

References