AN ARCHITECTURAL PROPOSAL FOR A CROSS-LANGUAGE SYSTEM TO FEDERATE MULTILINGUAL DIGITAL LIBRARIES

Nieves R. Brisaboa, Ángeles S. Places, Carme F. Pérez-Sanjulián, Francisco J. Rodríguez
Universidade de A Coruca
Campus de Elvica s/n.
15071 – A Coruca, Espaca
{brisaboa, carme}@udc.es, {asplaces, franjrm}@mail2.udc.es

Abstract. This work presents an architecture to federate pre-existing Documental Databases with documents written in different languages. A user will be able to ask queries to all the federated documental databases using a unique and friendly user interface that will be generated in the language she chooses (among the available ones in the system). The query will be executed over all the relevant databases in the system no matter which language their documents are written in. The query will be automatically translated if it is necessary.

The system uses ontologies to integrate the different database schemas, and it also includes three dictionaries. The first one stores the names of the concepts and some of the possible values for some of those concepts in all the languages present in the system. The second one stores the skeleton of sentences the user will use to express the query. The third one stores the directions, texts, etc. of the user interface.

The architecture described in this paper uses ontologies not only to represent the global schema but also to guide the execution of software modules in the system. Therefore, when a new Digital Library is added to the federation, there is no need to rewrite any code, but only to modify the ontologies and maybe the dictionaries (only if a new Digital Library has documents in a language not yet included in the system). This is possible because of the User Interface Generator, which is a module that generates the user interface code (HTML and Java) every time a user accesses the system. Likewise, ontologies guide the execution of other modules in the system.

Keywords. Federation of Digital Libraries, Cross-language search, Federated Search, Ontologies.

1 Introduction

Europe, which we consider to be our future common homeland, is not a monolingual and monocultural entity. It is a large jigsaw of languages and cultures that means a cultural richness that is necessary to preserve.

Within the European Union, there are more than 50 autochthonous languages in everyday use, even when only 11 are official languages. All the European Union's members have at least one autochthonous linguistic community which differs from the majority of the State and has its own distinct language (or languages) and identity [15]. Among the 370 millions of European citizens, nearly 50 million belong to these linguistic communities that have been speaking these autochthonous languages for thousands of years. They form an integral part of Europe's common cultural and linguistic heritage, and most of them have very rich cultural, literary and folk traditions. Some of these languages belong to the same linguistic family and are similar enough to be mutually understandable,

especially in writing.

In the same way that in the 15th and 16th centuries the printer became the principal way of preserving and spreading literature as much as other cultural heritage, nowadays the Web is becoming the medium for preservation and dissemination of any cultural manifestation. It is clear that only languages with presence into the Web will have the opportunity to survive the English language invasion. All around Europe, there are efforts to create documental databases, supported by the European Union. These databases are real Digital Libraries with documents and literature in different languages, but when those languages have few speakers the isolated efforts are not sufficient, because visitors of such Web sites are scarce and therefore its maintenance is expensive.

Federating those Digital Libraries into a cross-language federation system will increase the number of visitors of all the Web sites into the federation. Such a system will take advantage of the fact that some languages are similar enough to be understood by speakers of other languages (who are able to read but not to write a query directly). On the other hand, the federation will facilitate international researchers to find documents and other cultural manifestations from other cultures, even if they do not understand the language which the documents are written in. For example, some researcher studying poetry written by women in the 19th century will be interested in getting the names of such authors from all the Digital Libraries in the federation, even if she is not able to read the retrieved literary works. Those two facts made the Cross-language Text Retrieval systems gain more and more attention in the last years [9][18].

In this paper we present a proposal for the architecture of a federated database system we are building. In our proposal, we assume that the Digital Libraries are pre-existing and heterogeneous in language and type of corpus as well as in data model and technology. The user, when connecting our system, will choose the language (among those available at that moment) for the user interface and then he will write the query in the chosen language over a friendly interface in Bounded Natural Language [7][20]. The query will be analyzed and redirected to all the Digital Libraries in the federation. The retrieved data and documents will be presented to the user in an interface in the chosen language (but of course the documents will be in their original language, because our system does not translate literary works).

2 System Architecture Overview

Several architectures have been proposed to build systems that work with heterogeneous and geographically dispersed databases [1][2][17][22][24]. The relevance of the Database Federation [21] subject has been increasing at the same time as the number of isolated data sources available in the Web grows. Nowadays, federating and integrating different data sources into a broker system, which provides mechanisms to query these data sources using a unique and intuitive user interface, is an important and current investigation area [14].

The proposed architecture is composed of four isolated layers and we define three exchange languages to communicate them. This architecture is shown in Fig. 1. To understand the system it is necessary to consider the ontological architecture described in the next section. The architecture layers and software modules are:

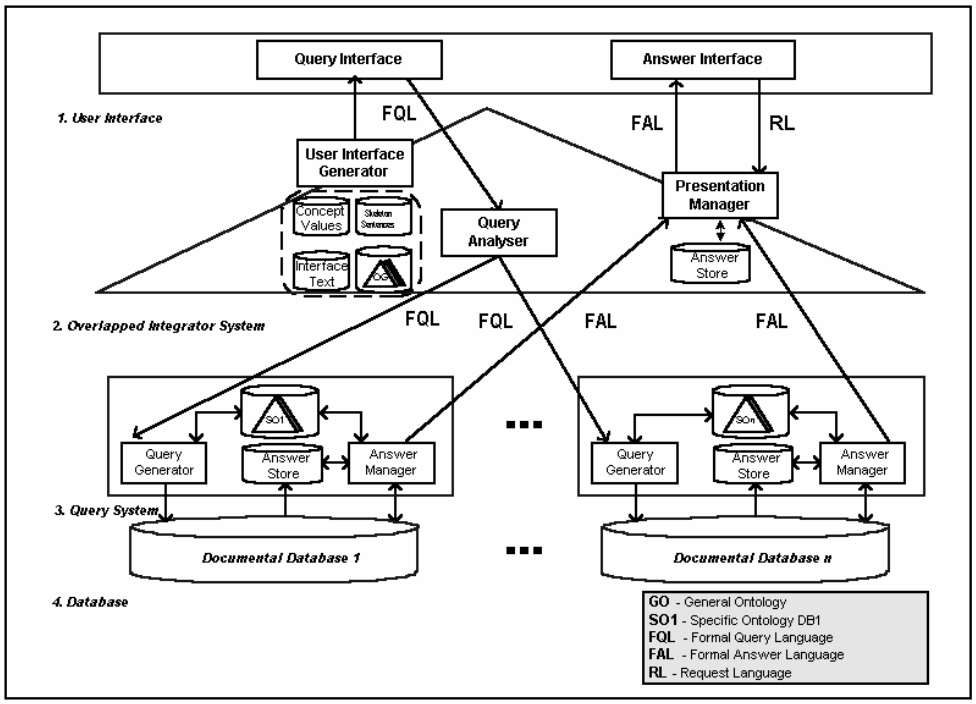

Fig. 1. System Architecture.

1. **Layer 1: User Interface:** The *User Interface* is generated, every time a user accesses the system, by the *User Interface Generator* placed in layer 2.

2. **Layer 2: Overlapped Integrator System:** The *Overlapped Integrator System* is the real broker in our federation. The *General Ontology* and all the dictionaries are placed here. This layer has some modules:

   - ***The User Interface Generator,*** which generates the *User Interfac7e* following the *General Ontology* and using the dictionaries as we explain in the ontology section.

   - ***The Query Analyser*** that analyses the user query and redirects it to Layer 3.

   - ***The Presentation Manager.*** It integrates the answers from all Digital Libraries, so the information can be presented to the user in a convenient way. This module also maintains the session so the user can navigate through the retrieved documents.

3. **Layer 3: Query System:** There is a *Query System* for each Digital Library in the federation. Its task is to query (*Query generator module*) its associated database (or file system that supports the Digital Library) using its *Specific Ontology*. Then, the retrieved document *Ids* are stored (*Answer Store*) so the user can navigate through them (*Answer Manager*) and through those ones retrieved from other Digital Libraries by the same query. Although all the *Query Systems* perform similar tasks, they need to be adapted (query language, operative system, etc.) to the specific associated database.

4. **Layer 4: Document Database:** The databases that can be integrated in our system are pre-existing and independent of it. That is, managing the databases is not a task of our system. Therefore, if a database has Text Retrieval capabilities [3][4][5][16][19], any needed pre-process has to be already performed and those Text Retrieval techniques have to be already implemented. Our *Query System* will connect to the database

using its Text Retrieval features, if they are available, and will ask the user queries in the appropriate DML (Data Manipulation Language).

The communication between layers is made by means of three exchange languages defined by us in XML [24] to accomplish that communication goal: *Formal Query Language* (FQL), the *Request Language* (RL) and the *Formal Answer Language* (FAL). These languages are used to give all queries (FQL), documents (FAL) and navigation orders (RL) the same format.

3 Ontological Architecture

One of the most important problems in the database federation is to create a catalogue (global schema) that integrates the diverse database schemas managed by the system. Using this global schema, the broker system can know which information is managed by each member database and where (which databases) the user query must be redirected.

There are many research works that emphasise the use of ontologies as a way to integrate dispersed and independent databases [10][17]. An ontology is a specification of a conceptualisation [11][12][13], that is, a set of concepts and the relationships among them. An ontology describes a domain of interest. An ontology can give a homogeneous description of the different schemas of databases integrated into the system.

In our system the ontologies represent the common concepts, that is, the concepts that any user, even if she is not an expert in the database domain, can perfectly understand. We only represent those common concepts in the ontology because only them are useful to the user interface. In our ontologies, concepts are arranged in tree shapes. There are three kinds of relationships among concepts into our ontologies: *Generalisation/specialisation* (*"is a"* relationship), *description* (*"has"* relationship) and *Aggregation* (*"is part of"* relationship). In the proposed architecture, as shown in Fig. 1, two ontological levels are considered.

- **The *General Ontology*** is placed in the *Overlapped Integrator System*. It is an abstraction of the member database schemas and it integrates all the concepts present in the *Specific Ontologies*. Every concept in the general ontology has associated a database list where that concept is relevant. An example of *General Ontology* is shown in Fig. 2 (without the related database list for readability).

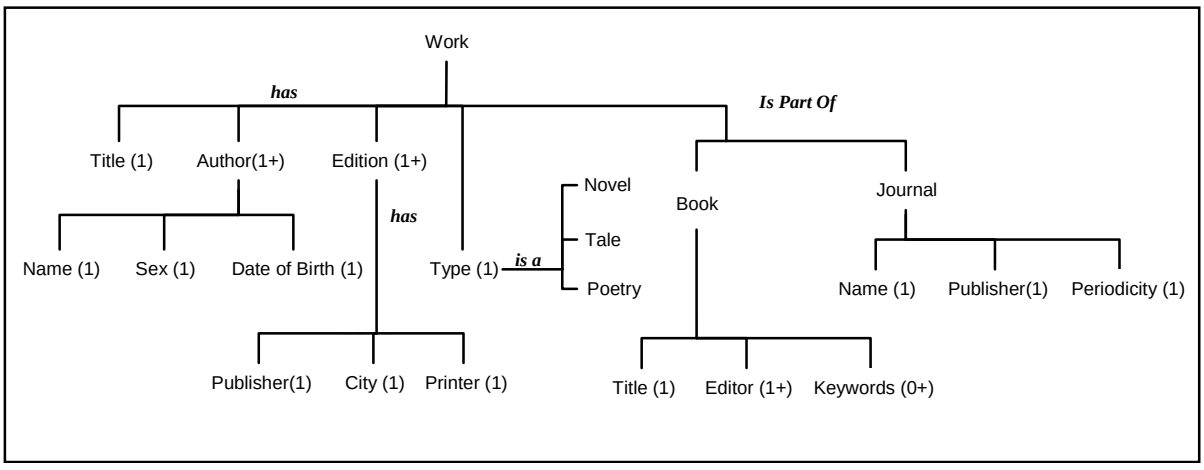

Fig. 2. General Ontology

- **A Specific Ontology** is defined for each documental database, describing some of its concepts. It has a subset of the concepts in *General Ontology.* Every concept in the *Specific Ontology* has associated the expression necessary to access the corresponding data in the associated database. This expression depends on the DBMS. For example, in a relational database, a concept can have the relation and attribute names where the concept is stored. These expressions, associated to the concepts in a *Specific Ontology,* are used by the *Query Generator Module* to adapt the query received in *FQL* to the DML of the DBMS of the associated documental database.

4 System Dynamics Overview

In this section we give an intuitive description of the dynamics of some of the modules of the system (a more detailed description will be found in the full paper).

- **The *User Interface Generator*** reads and follows the *General Ontology*. This module presents the concepts to the user in order to allow her to write a query in an easy way (by expressing restrictions over the concepts). To do its work *the User Interface Generator* uses a set of skeletons of sentences in *Bounded Natural Language* [7][20].

A sentence in *Bounded Natural Language* is a sentence with gaps that the user must to fill in order to express a restriction over a concept. To fill the gap sometimes the user must write the restriction (i.e. the name of the author she is looking for), and sometimes she must choose the desired value from a list (i.e. genre of the author or language of the works he is interested in). The system has a set of skeletons of sentences in *Bounded Natural Language* and the *User Interface Generator* chooses the appropriated skeleton for each concept, completes the skeleton with the concept and then presents the sentence to the user. For example a skeleton can be: *"The <concept must be ....."* . The *User Interface Generator* will choose this skeleton to ask for the *author* or for the *publisher*, etc. Then it will complete the skeleton with the concept name and will present to the user the sentence in *Bounded Natural Language "The author must be ....."*.

As we said, some of the concepts in the ontology can have a predefined set of values (for example genre can be male or female). In those cases the possible values will be presented to the user in a list to made easier for the user to fill the gap in the *Bounded Natural Language* sentence.

In the *Overlapped Integrator System*, three dictionaries can be found: one (Concept Values) for the concept in the ontology and its values when they are a finite set (see Fig 3), a second one (Skeleton Sentences) for the skeletons of the sentences in *Bounded Natural Language*, and the third one (Interface Text) for texts, directions, etc. in the user interface.

- **The Query Analyser Module.** After the user writes his/her query, it is sent to the *Overlapped Integrator System* in FQL format. The *Query Analyser* reads it and searches in the *General Ontology* the concepts the user is interested in (those used to describe the desired documents). Then the *Query Analyser* read the database list associated to each concept to decide which databases the query must be redirected to.

- **The Query Generator Module** adapts the query received in FQL (from the Overlapped Integrator System) to the DML of the DBMS of the associated documental database. To do this transformation, this module uses the information associated to each concept in the Specific Ontology (as well as information about the DML).

English	**Spanish**	**Galician**	**.....**
Genre	**Sexo**	**Sexo**	**.....**
Male	Hombre	Home	.....
Female	Mujer	Muller	.....
Type	**Tipo**	**Tipo**	**.....**
Novel	Novela	Novela	.....
Tale	Cuento	Conto	.....
Journal	Revista	Revista	.....
Date of Birth	**Fecha de Nacimiento**	**Data de Nacemento**	**.....**
.....	.....	.....	.....

Fig 3. Dictionary of concepts and values into the General Ontology

5 Conclusions and Future Work

Our system has some interesting advantages:

- **Increasing system scalability and facility to adapt to changes:**

  In our system, the use of ontologies reduces the changes that have to be done when a new database is added to the federated system. It is necessary to carry out only three tasks: building an "ad hoc" *Specific Ontology* for the new database, completing the *General Ontology* with the new concepts that appear in the new database and updating only the database lists of those concepts in *General Ontology* that are concepts relevant to the new database. It is obvious that only the two last changes will be necessary if a database is dropped. When a new language is added only the dictionaries must be updated.

  Likewise, it will not be necessary change the modules in *Overlapped Integrator System*. The ***User Interface Generator*** will automatically generate the *User Interface* taking into account the new concepts in the *General Ontology***.** Likewise the ***Query Analyser*** will be able to redirect the user queries to a new database using the database list associated to the concepts in the *General Ontology*. The ***Query Generator*** associated to the new Specific Ontology will translate the query in FQL into the DML of the associated database. Therefore, no changes need to be made in the system code except the ones in the *General Ontology*.

  Furthermore, all the ***Exchange Data Languages (FQL, FAL, RL)*** will be automatically modified.

- **Logical and Physical Independence:**

Both Logical and Physical independence are a basic principle of good design in databases. The *Specific Ontology* gives physical independence to the system because changes in any member database (DBMS, table structure, etc.) will only affect the information associated to concepts in its *Specific Ontology.*

Likewise, the *General Ontology* gives logical independence to the system since adding, dropping or modifying databases does not affect the system but only the *General Ontology*.

We are working in the implementation of a prototype, applying the ideas exposed in this paper. Initially, this system will integrate three documental databases that store two different corpora of historic documents in Galician and Spanish languages.

References

1. Abelly, A.; Oliva, M.; Rodrнguez, E.; Saltor. F. The BLOOM Model Revisited: An Evolution Proposal. ECOOP Workshop (Proc. ECOOP Workshops & posters, ECOOP'99, Lisbon, June 1999).

2. Arens, Y., Hsu, C., Knoblock, C. A. Query processing in the SIMS Information Mediator. Advanced Planning Technology, Austin Tate (Ed.), AAAI Press pp. 61-69, Menlo Park, CA, 1996.

3. Baeza-Yates, R.; Navarro, G. Integrating contents and structure in text retrieval. ACM SIGMOD Record, 25(1):67-79, Marzo 1996.

4. Baeza-Yates, R.; Navarro, G.; Vegas, J.; Fuente, P. A model and a visual query language for structured text. En Berthier Ribeiro-Neto (Eds.) Proc. of the 5th Symposium on String Processing and Information Retrieval, pбginas 7-13, Santa Cruz, Bolivia, Sept 1998. IEEE CS Press.

5. Baeza-Yates, R.; Ribeiro-Neto, B. Modern Information Retrieval, Addison-Wesley, 1999.

6. Brisaboa, N.R., Durбn, M.J., Lalнn, C., Iglesias, E.L, Lypez, J.R., Paramб, J., Places, A.S. y Penabad, M.. Integrating the access to documental databases on the web. The Fifth World Conference on Integrated Design and Process Technology IDPT 2000. Proceedings of The Fifth World Conference on Integrated Design and Process Technology IDPT 2000. Dallas, Texas. Junio 2000.

7. Brisaboa, N.R.; Durбn, M.J.; Penabad, M.R.; Places, A. S. "A Collaborative Framework for a Digital Library". VI International Workshop on Groupware CRIWG'2000. Madeira (Portugal).Proceedings of the Sixth International Workshop on Groupware CRIWG'2000. IEEE Computer Society Press. Octubre 2000

8. Chandrasekaran, B.; Josephson, R. What are ontologies, and why do we need them? In IEEE Inteligent systems, 1999.

9. Cross- Language Evaluation Forum. http://www.iei.pi.cnr.it/DELOS/CLEF/

10. Elmasri, R. WebOntEx: Extracting Ontologies from Web Pages. Conferencia invitada en las V Jornadas de Ingenierна del Software y Bases de Datos (JISBD'2000). (Publicado Abstract). Valladolid, Noviembre 2000.

11. Gruber, T. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. IJHCS, 43 (5/6): 907-928. 1994.

12. Gruber, T. http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

13. Guarino, N. (ed.), Formal Ontology in Information Systems. Proceedings of FOIS'98. Amsterdam, IOS Press, pp. 3-15, Trento, Italy, 6-8 June 1998.

14. Hasselbring, W.; van den Heuvel, W.-J.; Houben, G.J.; Kutsche, R.-D.; Rieger, B.; Roantree, M.; Subieta, K. Research and Practice in Federated Information Systems. Report of the EFIS'2000 International Workshop. ACM SIGMOD RECORD Web Edition. Volumen 29, Nъmero 4. Diciembre 2000.

15. Euromosaic: The production and reproduction of the minority language groups in the European Union. ISBN 92-827-5512-6. Luxembourg. 1996.

16. Excalibur.Informix Corporation. http://www.informix.com/

17. Mena, E., Illarramendi, A., Kashyap, V., Sheth, A. OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. Published in the journal Distributed And Parallel Databases (DAPD). 1998.

18. Miller, G. WordNet: A lexical database for English. Communications of the ACM, 38(11), 1995.

19. Context.Oracle Corporation. http://www.oracle.com/

20. Penabad, M.; Durбn, M.J.; Lalнn, C.; Lypez, J.R.; Paramб, J; Places, A. S.; Brisaboa, N.R.; Using Bounded Natural Language to Query Databases on the Web. Information Systems, Analysis and Synthesis ISAS ' 99. Proceeding of the Information Systems, Analysis and Synthesis ISAS'99Orlando (Florida), Julio - Agosto 1999.

21. Sheth, A. P., Larson, J. A. Federated databases for managing distributed, heterogeneous, and autonomous databases, Computing Surveys 22:3 (1990), pp. 183-236.

22. Subrahmanian, V.S., Adali, S., Brink, A., Emery, R.,. Lu, J., Rajput, A., Rogers, T., Ross, R., Ward. C. HERMES: A heterogeneous reasoning and mediator system. Technical report, University of Maryland, 1995.

23. Sudarshan Chawathe, Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey Ullman, and Jennifer Widom. The TSIMMIS project: Integration of heterogenous information sources. 16th Meeting of the Information Processing Society of Japan pp. 7-18, Tokyo, Japan, October 1994.

24. World Wide Web Consortium. Standard XML http://www.w3.org/XML