A System for the Integrated Access to Digital Libraries

Nieves R. Brisaboa, Miguel R. Penabad, Ángeles S. Places, Francisco J. Rodríguez

Departamento de Computación. Facultade de Informática
Universidade da Coruña
{brisaboa, penabad, asplaces, franjrm}@udc.es

Abstract. There has been lately much effort put on the creation of digital libraries containing antique documents, in order to preserve our cultural heritage and provide a broader access to them, but they are usually isolated one from another. This work presents a system, based on the use of XML trees, to federate three existing digital libraries that contain documents from the Spanish Golden Age (16th-18th centuries). This federation offers very valuable advantages to a wide community of researchers, who will be able to access (at present time, three) databases of historic documents through a unique entry point.

Keywords. Digital Library Federation, XML trees, User Interfaces.

1 Introduction

Internet has become one of the most important places to publish any kind of information. It is especially important if we consider documents such as antique books, manuscripts, or any other element of our documental heritage, because their publication using the usual methods is too expensive and not economically viable, These documents are kept in museums and libraries with historical archives with very restricted access, so their Web publication serves two purposes: offering them to the research community, and preserving them. These documents, besides their great beauty, are a magnificent source of information about different cultures like the Spanish Baroque. Through the analysis of these works, researchers can find information about morality, habits, technology, education, or other aspects that are of interest for a wide range of researchers, including for example Anthropologists, Historians, or Philologists. Additionally, and no less important, it helps the preservation of the documents, preventing them from disappearing due to their antiquity and fragility.

With these two goals, much effort has been put to publish these documents on Internet, usually in the form of documental databases or digital libraries. We have

This work was partially supported by CICYT, grants TEL99-0335-C04-02 and TIC2002-04413-C04-04.

created two of them, one for Spanish Emblem Books [2] and one for Early Spanish Press [10]. However, it is common that these digital libraries are isolated, thus making it difficult to jointly use several (often closely related) sources of information. As for our libraries, they are already available on Internet by using separate interfaces.

In order to augment their potential usefulness, it would be desirable to have a system for the users to access, through a unique interface, all available corpora. There has been a lot of effort on the field of database federation [7,11,12]. There are also works especially focused on the federation of documental databases [8]. However, these works lack some aspects we consider crucial. As they only federate databases, they do not consider the user interface, which is important for us to provide a unique Web access point. Specific aspects about documental databases, such as the exploitation of the available Text Retrieval techniques on each database are also usually ignored.

We have built a federated system that considers these aspects, focusing specifically on federating digital libraries, keeping their text retrieval capabilities, and also giving special importance to the user interface to access them. Our system integrates our two digital libraries (their underlying databases), plus a new one containing non Spanish Emblem Books that were translated into Spanish in this period because of their importance. These emblem books were very popular at the time, so their study is very interesting, because they inspired other Spanish authors to write their own emblem books. This system is already functional, but it will be available through Internet only when the database of Translated Emblem Books is completed.

The federated system offers a unique access point to browse through the documents or perform searches. This interface has been developed following some guidelines that will make it intuitive and easy to use, yet powerful and flexible enough to respond to queries over any (or all) databases in the system. This federated system was initially built ad hoc for these databases; however, the design of its architecture allows us to extrapolate the ideas we used to build it, so they can be used to federate any set of documental databases. These ideas are based on the requirements that any federated system must meet: *scalability* (increasing the number of databases in the federation should not decrease the overall system efficiency), *easy adaptation to changes* (adding, dropping or modifying databases must be easy and quasi-automatic) and *user-friendliness*. In order to build a system that meets these requirements, we based our work on the use of a layered structure, and the use of XML trees to perform the schema conciliation of the databases and to help building the user interface.

The rest of this paper is organized as follows. Section 2 describes the three databases included in our federation, showing their interest and how we manage them. Section 3 describes the XML trees and how we use them to help integrating the different databases and building the user interface. Section 4 offers an overview of the architecture of the system. The implementation of the system is described in Section 5, and the last section offers our conclusions.

2 Description of the Databases

We are currently working with three databases that store information about Literature from the 16th-18th centuries: Spanish Emblem books, Early Spanish Press (*Relaciones de Sucesos*), and Non-Spanish Emblem books translated into Spanish at that time. They are two large databases that store digitized pages as well as transcriptions, enriched with a huge amount of information coming from the analysis of the documents by experts on History of Art and Hispanic and Latin Philology.

The first two databases correspond to Literature written in Spanish, from the Spanish Golden Age ("Siglo de Oro") and are already available through the Web; the third one stores emblem books that originally appeared in non-Spanish languages (Italian, Latin, or other European languages), and were translated into Spanish during that period. This last database is still being populated and will also be available soon.

Emblem Literature was basically a moral literature, trying to promote moral and ethical norms as well as ideas and concepts about morality in general. Emblem books were formed by emblems, which are types of ideograms that expressed an abstract idea through a picture, accompanied by a motto containing the moral principle. The idea was further explained by an epigram or short poem and a commentary.

The Spanish Emblem Books database [2] stores information for 27 emblem books, containing more than 1800 emblems. These emblems lead to a thesaurus of about 15000 authorities, 7000 exemplas, 16000 onomastics, and 10000 sources for the images. Likewise, we offer about 6500 digitized pages. We have not yet reliable information about the quantity of information available for the last database, Translated Emblem Books, because we are still populating it.

Spanish early press documents ("Relaciones de sucesos" in Spanish) come from the 15th-17th centuries, and were the precursor of the current journalism. They related an event with the goal of informing and entertaining the public. According to the subject of the story, there were different types of reports, like *Festive events*, such as monarchic or religious events, *extraordinary events* like miracles or strange events (the current sensationalist press), or *Bullfight events*, predecessor of the sports press.

The Early Spanish Press database [10] stores information about these reports, including catalogues of reports, thesauri of epithets, illustrations, the different editions of a given report, and the libraries where they can be found nowadays. The digitized pages of all available reports are also stored in the database. The current content of the database includes information about more than 1800 reports, with 280 illustrations, and a thesaurus of about 1000 epithets used in them. There is also information about the 22 libraries where the original reports can be found.

All the works considered in these databases constitute a very rich and complex source of information related to the customs of those centuries in Europe in general and in Spain in particular, because they provide data about society, morality, customs, news, knowledge and conventions. A characteristic, common to all these works, is that they are very useful to a wide range of researchers from a variety of disciplines (History of Literature, History of Art, Anthropology, Sociology, Philology, etc.).

3 System Overview

The architecture of our system is based on the use of XML files, named Concept Trees and Mapping Trees, that store key information that controls how all modules of the system work. Concept Trees representing all relevant "searchable" concepts of the component databases are used to help the user build the query. Each underlying database has a Mapping Tree that is used to translate the query to the appropriate database query language.

3.1 Architecture of the System

The architecture of the system described in this paper is composed of four separate layers. The communication among them is carried out by using two exchange, intermediate languages, exclusively defined for this purpose. This architecture, shown in Fig. 1, is fully described in [6]. What follows is only a brief overview:

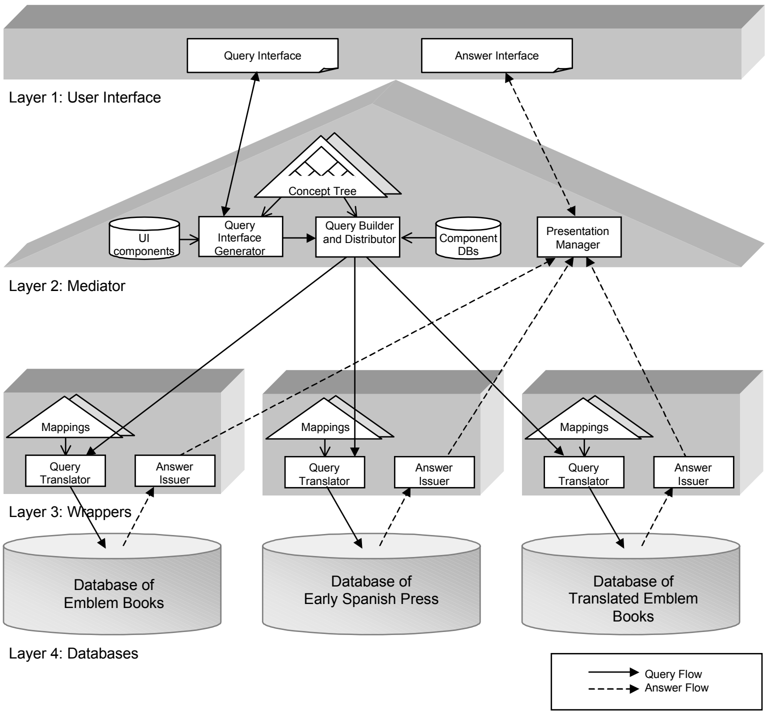

Fig. 1. Architecture of the System

1. *Layer 1. User Interface:* The user interface is generated every time a user accesses the system, so it is not a real layer of the architecture. However, we represent it as a (conceptual) layer in order to properly describe the interaction of the users with our system, because all interactions (queries and answers) are made through the user interface.

2. *Layer 2. Mediator:* The Mediator generates the user interface following the Concept Tree, as we shall see in the next section. After the user expresses his query, the Mediator analyses and redirects it to the Wrappers of the databases involved in that query. After the query is executed in the pertinent databases, the Mediator shows the user an answer Web page with a summary of the results obtained from the databases, as well as the appropriate links to the answer interfaces of these digital libraries, so users can navigate through the obtained documents.
3. *Layer 3. Wrappers:* Every Wrapper is associated to a particular database. Its tasks are to query it and retrieve the answers. For these purposes, this layer uses the Mapping Tree of the database. Although all the Wrappers perform similar tasks, they need to be adapted to the specific associated database.
4. *Layer 4. Documental Database*: The databases that can be integrated in our system are preexisting and independent of it. Moreover, in our case we have that two of the three considered databases are currently available through Internet, with a particular user interface that enables to use them as Digital Libraries. The fact of federating these Digital Libraries in our system does not mean that they cannot answer other queries coming from their own interface.
   Likewise, we must note that if a database has Text Retrieval capabilities, our system is capable of exploiting them, but any needed preprocess (such as indexing) has to be already performed and those Text Retrieval techniques have to be already implemented. That is, managing the databases is not a task of our system.

3.2 Trees

The execution of all the modules in our architecture is guided by the information stored in a set of XML files, denoted Trees here because or their hierarchical representation. All Trees are composed of nodes that represent “searchable” concepts existing in the databases. That is, not all concepts of a database, but only those that can be used to perform searches, are included. These concepts are described by properties or attributes, and arranged in a tree in order to represent the relationships among them. For our domain of interest, digital libraries, we have considered relevant only the following two types of relationships:

- Generalization/Specification: It represents the typical “is a” relationship. For instance, a “Work” is a “Thesis”, a “Book” or a “Journal”.
- Description: It can be represented as a “has” relationship. It is used to represent that a concept is described by other (sub)concept. For example, a Work “has” an Edition, which in turn is described by the attributes “Year of edition”, “Publisher”, and “Promoter”.

We deal in our system with two types of trees, designed for two different purposes: Concept Trees, which are placed in the Mediator, and Mapping Trees, which are placed in the Wrappers (see Fig. 1):

- Concept Trees: They are abstractions of the schemas of all component databases. The root concept of these trees is the object (concept) that can be retrieved by a query. There will be as many concept trees as different concepts can be retrieved.

When the user expresses a query, he must decide which concept wants to retrieve, selecting the Concept Tree that will be used to express the query.
This type of tree is used to generate the user interface, offering the user to navigate through all the concepts on it, or to establish the query constraints for any of them. Thus, each attribute has associated the different ways a user can establish the search conditions for the attribute. Depending on each case, it can be a Bounded Natural Language [4, 5] sentence, or a Cognitive Metaphor [4], as we shall see in Section 4.1. If the attribute describes a characteristic with a finite number of possible values, the list of values is also stored. For example, for the “Stanza” attribute in the Concept Tree of Fig. 2. This attribute stores the list “Quatrain, Sonnet, Tercet …” with all the possible values for this attribute.

– Mapping Trees: A Mapping Tree is defined for each documental database, and describes only the concepts of its associated database. Every concept and attribute in a Mapping Tree has associated the expression necessary to access the corresponding data in the associated database, which is completely dependant on the database DBMS.
For example, a concept represented in a Mapping Tree for a relational database can have the relation and attribute names where the concept is stored, or a more complex expression, like a complete SQL SELECT statement. If the database is capable of using some kind of Text Retrieval technique, the information associated to a concept like “content” or “topic” will be the directions to call the Text Retrieval algorithm implemented in the database. Table 1 shows the XML fragment of the Mapping Tree for the Emblem Book database that shows the mapping information associated to the attribute “Topic”.

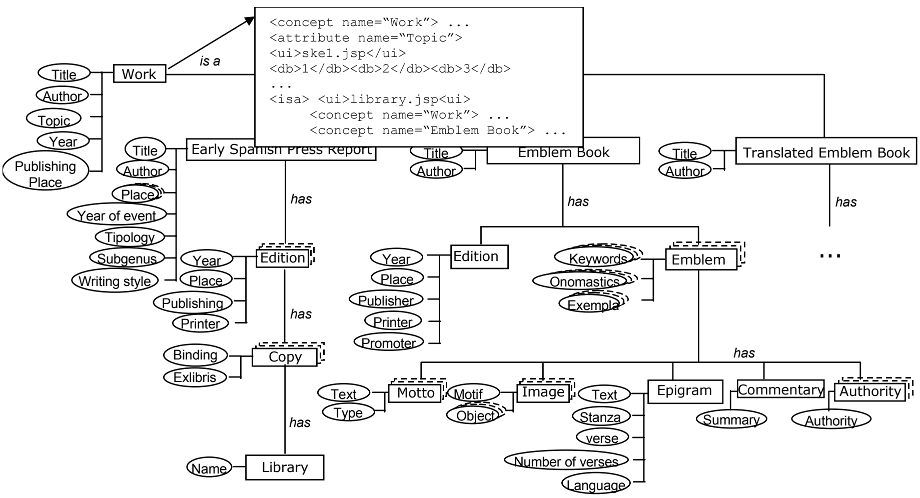

Fig. 2. A Concept Tree

4 Implementation of the System

4.1 Query Interface Generator

We believe the user interface is a crucial aspect for the success of any Web system. Therefore, our architecture is highly concerned about the design of an intuitive, friendly and easy to use interface. We have proposed in [4] the following three techniques to design user interfaces that, being combined and systematically used, can guarantee the success of the interface:

- Use of *Cognitive Metaphors* [4]: Web pages are built taking elements from the real world. This technique is widely used, not only for the design of Web pages but for any computer interface.
- Use of *Bounded Natural Language* [4, 5]: It consists on offering the user a set of natural language sentences with gaps. The user must choose the sentences of interest and fill the gaps on them in order to express the query.
- *Navigational Approach* [4]: The main idea is to present the user a single query interface, where the user can obtain a first set of results. From these results the system will allow the user to navigate through them and obtain information about the items that he clicks. So, instead of a form full of text fields to express the constraints of the attributes to build a query, and when it is possible, this approach makes available the possible values of these attributes and allows expressing the constraints (refining the query) with simple clicks. Also, showing the answers sorted by some attribute, the user can select one value without previous knowledge of it, and browse and locate the results of interest.

We also believe that having several complexity levels for different user profiles is very useful, so we offer unsophisticated interfaces to allow expressing simple queries for general users, and slightly more complicated interfaces to allow experts expressing more complex queries.

The Query Interface Generator offers the user a "controlled" natural language (Bounded Natural Language or BNL) or Cognitive Metaphors (applying or not the Navigational Approach) to express the queries. The Query Interface Generator dynamically builds the query interface using the Concept Trees, providing the user with the mechanisms to express conditions over attributes, and to navigate through the Concept Tree to choose the attributes the user is interested in.

Basically, it takes as its first input the root concept of the Concept Tree previously selected by the user, and operates depending on the concept. If the concept has a specialization (an "*is-a*" relationship), the appropriate procedure allows the user to choose one of its specializations or work with the general concept without considering its specializations. The query interface generated to perform this selection is based on a sentence in BNL or, if it exists, on the Cognitive Metaphor associated to the is-a relationship. If the relationship is a Description relationship (*has*), it offers the user the possibility of considering as many attributes and subconcepts as the user wants, thus going down the Concept Tree. Again the interface is generated by using a sentence in BNL or the cognitive metaphor associated to the "has" relationship.

Finally, the Query Interface Generator permits to establish restrictions over the chosen attributes. To do this, the Query Interface Generator shows the user the Bounded Natural Language or the Cognitive Metaphor associated to these attributes in the concept tree.

4.2 Query Builder and Distributor

As the user is expressing the query, by navigating through the concepts on the Concept Tree, the query is being internally stored as an XML document. Depending on the considered concepts, the query can be sent to all the Wrappers or only to one of them. In the example that follows, the whole XML query will be sent to the Wrappers of all the databases in the federation because all databases are involved. If the user had chosen “Emblem Book” in the first step, the query would consider only concepts in the Emblem Book subtree and that query would be sent only to the Emblem Book database. An example of a XML query is shown in Table 2 (a).

4.3 Query Translator

XML queries must be translated into the language of each database (in our case, all of them use SQL). To carry out this task the Query Translators read the XML query and take, for each concept or attribute, the fragment of the SQL statement (stored in the associated Mapping Tree) needed to translate the condition over that concept or attribute.

Recall that Table 1 shows a fragment of the Mapping Tree for the Emblem Book database, which will be used to translate the XML query of the Table 2(a). There, the `mapping` tag can be distinguished for the attribute “Topic”. Notice that each mapping is made up of other three elements named by `select`, `from` and `where` tags. These elements contain the fragments to be added to corresponding part of the final SQL statement.

Table 1. Fragment of Mapping Tree for the Emblem Book Database

```
<attribute name="Topic">
<mapping>
<where>#limit# = (select count(*)
                 from clave cl
                 where em.cod_obra = cl.cod_obra
                 and em.cod_emblem = cl.cod_emblem
                 and cl.clave like #value#</where>
<optional end=")">and cl.clave like #value#</optional>
</mapping>
</attribute>
```

Different degrees of gray in Table 2 depict the process followed by the Query Translator of the Emblem Book Query System to build the final SQL statement. Each degree indicates a fragment of XML and its corresponding SQL.

Table 3 shows the final SQL Early Spanish Press database. The process of building these two SQL queries is similar to the one for Emblem Literature database. Since the Spanish Press database is managed by Oracle 9i, which has text retrieval capabilities, the SQL query includes a “Contains” clause. This clause is offered by the

Oracle *inter*Media package included in Oracle since version 8i and allows for different types of text retrieval searches.

Table 2. XML query and the generated SQL for the Emblem Literature database

(a) XML	(b) SQL

`<query>`
` <concept="Work">`
`   <attribute="Topic">`
`       <contains limit="5"/>`
`       <value cons="sin"/>`
`       <value cons="cleric"/>`
`       <value cons="Inquisition"/>`
`       <value cons="stake"/>`
`       <value cons="witch"/>`
`    </attribute>`
`    <attribute="Year">`
`        <between/>`
`        <value cons="1500"/>`
`        <value cons="1650"/>`
`    </attribute>`
`  </concept>`
`    </query>`
	`select cod_obra, cod_emblem`
	`from obra ob, emblema em, edicion ed`
	`where ob.cod_obra = em.cod_obra`
	`and ed.cod_edic = ob.cod_edic`
	`and`
	`5 = (select count(*)`
	`      from clave cl`
	`      where em.cod_obra = cl.cod_obra`
	`      and em.cod_emblem = cl.cod_emblem`
	`      and (cl.clave like "sin"`
	`           or cl.clave like "cleric"`
	`           or cl.clave like "Inquisition"`
	`           or cl.clave like "stake"`
	`           or cl.clave like "witch"))`
	`and (ed.cod_edic >= 1500`
	`and ed.cod_edic =< 1650)`

Table 3. Query to Early Spanish Press database

```
select tituloabre, cod_edic
from relacion rel, edicion ed
where rel.tituloabre = ed.tituloabre
and contains(rel.titulo, 'sin & cleric &
             inquisition & stake & witch', 10) > 0
and (rel.fecha_acon >= 1500 and rel.fecha_acon =< 1650)
```

4.4 Presentation Manager

The Presentation Manager displays the first answer page just when the Builder and Distributor Module sends the query to the appropriate wrappers. This Web page presents the list of digital libraries where the query is to be executed. For each digital library, the following attributes (using unqualified Dublin Core fields) are specified: the name or title of the database; a description of the data stored in the database; a surface address, a telephone number or a contact e-mail; the date of the last update; the main URL of the digital library; and the query that is to be executed in the database (it is possible that the database does not include some of the concepts appearing in the query).

This Web page is updated as the Presentation Manager receives (from the Answer Issuer) the summary of the results obtained from the databases, showing for each digital library the number of results that match the query. Once this answer is obtained, users can access each of the digital libraries to display the documents matching his/her query, using the digital library's own interface.

Note the difference between the queries and the answers: while queries are performed in an integrated way, answers are always displayed using the digital libraries own interface. The main reasons that lead us to take this decision are the following:

- The answer interface of each one of the digital libraries is specifically designed for its data (see, for example, the Emblem Literature answer interface [2]). Therefore, those answer interfaces will always be more intuitive and user friendly than any other interface we can build to display (in an integrated way) the results from the databases.
- The federated databases are heterogeneous, so there will not appear, in any case, duplicates of the results obtained from any query coming from different databases.

There is another reason for this decision, even when it does not happen in our system (because we have also built each independent digital library): the administrators of the digital libraries might not be willing to offer their database information through an answer interface different from their own. Furthermore, there will be cases when the direct link to their answer interface showing the results of the query will not be available, and the user will have to repeat the query using the interface of the digital library to be able to display the results. In this case, our system is useful only to locate interesting sources of information about the topic of interest.

5 Conclusions

We have described in this paper a system to federate a set of digital libraries, initially designed to federate three digital libraries containing antique Spanish documents, but adequate to federate any set of related documental databases. The main idea of the system, the use of XML trees, gives a number of advantages, the main ones being those that make our system to accomplish the three aims shown in the introduction that any federated system should achieve: a system that is scaleable and easy to adapt to changes, with a friendly user interface created using the three techniques presented in this work.

References

1. Baeza-Yates, R. Ribeiro-Neto, B. Modern Information Retrieval Addison-Wesley, 1999.
2. Biblioteca Virtual de Literatura Emblemática. http://rosalia.dc.fi.udc.es/cicyt
3. Brisaboa, N.R., Penabad, M.R., Places, A.S., Rodríguez, F.J. Problems and Solutions to federate Digital Libraries. *Poster in 5th European Conf. on Research and Advanced Technology for Digital Libraries* (ECDL'2001). Darmstadt, Alemania, 2001.
4. Brisaboa, N. R., Penabad, M. R., Places, A.S., Rodríguez, F. J. Tools for the design of user friendly Web applications. *Lecture Notes in Computer Science (LNCS 2115), Springer Verlag* (EC-WEB'2001), pp. 29-38. Munich, Alemania 2001.
5. Brisaboa, N.R., Penabad, M.R., Places, A.S., Rodríguez, F.J. A Document Database Query Language. *Lecture Notes in Computer Science (LNCS 2405), Springer Verlag* (BNCOD'02), pp. 183-198. Sheffield, England. Julio 2002.
6. Brisaboa, N. R., Penabad, M. R., Places, A. S., Rodríguez, F. J.. Ontologías en Federación de Bases de Datos. Novática, Núm. 157. Information Retrieval and the Web, Ricardo Baeza-Yates, Peter Schauble (Eds.). ATI. 2002.

7. Busse, S., Kutsche, R.-D., Leser, U. Strategies for the Conceptual Design of Federated Information Systems In M. Roantree, W. Hasselbring, and S. Conrad (eds.), *Engineering Federated Information Systems, Proc. of the 3rd Workshop EFIS 2000*, pp. 23-32. 2000.
8. Gonçalves, M. A., France, R. K., Fox, E. A., Doszkocs, T. E. MARIAN Serching and Querying across Heterogeneous Federated Digital Libraries. *DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries 2000*. 2000.
9. 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.
10. Relaciones de Sucesos. http://rosalia.dc.fi.udc.es/Relaciones
11. Samos, J., Abelló, A., Oliva, M. Rodríguez, E., Saltor, F., Sistac, J. Araque F., Desgado, C., Garví, E., Ruiz, E. Sistema Cooperativo para la Integración de fuentes Heterogéneas de Información y Almacenes de Datos. Novatica ATI, 1999.
12. Chawathe,S; Garcia-Molina, H.; Hammer, J.; Ireland, K; Papakonstantinou, Y; Ullman, J; Widom, J.. The TSIMMIS project: Integration of heterogenous information sources. 16th Meeting of the Inf. Proc. Soc. of Japan, pp. 7-18, Tokyo, Japan, October 1994.