Mediation in a Dynamic Context: Arguing for a Request-Oriented Approach and Structuring it

Christophe Rey, Michel Schneider

Mediation in a dynamic context: Arguing for a request oriented approach and structuring it

C. REY, M. SCHNEIDER
LIMOS
University Blaise Pascal
63177 Aubière Cedex (France)

1. Introduction

The interoperability of multiple heterogeneous sources represents an important challenge considering the proliferation of numerous information sources as well in private networks (intranet) as in public networks (internet). The heterogeneity is the consequence of the autonomy: sources are designed, implemented and used in a independent way. The heterogeneity appears for different reasons : different types of data, different representations of data, different management software packages. The interoperability consists in allowing the simultaneous manipulation of these sources so as to join and fusion the data which they contain. In numerous domains, it is necessary to make different sources interoperable : electronic business, environment, economy, medicine, genome.

Interoperability problems appears in a very different way depending on whether sources are structured (data bases), semi-structured (HTML or XML pages), non-structured (any file). The access interfaces also influence the possibilities of interoperability. For example two data bases can be difficult to make interoperable when they are only accessible through specific web interfaces.

An interoperability approach which is studied for several years is based on mediation (Wiederhold 1992, Garcia-Molina et al 1997). A mediator analyses the request of a user, decomposes it into sub-requests for the various sources and re-assembles the results of sub-requests to present them in a homogeneous way. The majority of mediation systems operate in a closed world where one knows a priori the sources to make interoperable. This gives several advantages. At first it is possible to build an integrated schema which constitutes a reference frame for the users to formulate their requests. Then it is possible to supply the mediator with various information which are necessary for the interoperability and particularly for the resolution of the problems of heterogeneity. Different solutions were studied and experimented for the resolution of these problems. Let us quote in particular (Hull 1997, Saltor et Rodriguez 1997, Kedad et Métais 1999).

When one operates in a dynamic world where sources are not selected a priori and can evolve all the time, the elaboration of an integrated schema is a difficult task. It would be necessary to be capable of reconstructing the integrated schema each time a new source is considered or each time an actual source makes some changes. We suggest in this chapter an approach which does not require a preliminary integration of sources schemas but which is request oriented. The ideal request oriented mediation would be the following: the user request is rewritten in the terms of a domain specified through one or several ontologies. Potential sources are identified from the elements of this request. Schemas for each of these sources must be then extracted. The user

request is another time rewritten for every source according to its information capacity (sources not offering a sufficient capacity are no longer considered). Every source is then interrogated. Results are then formatted and integrated: heterogeneities must be solved. Residual conflicts are discovered and resolved from knowledge on sources (schemas) and domain (ontologies). The last stage is an evaluation of the quality of the results by the user. It allows the system to straighten its internal information bases (ontology of the domain, metadata warehouse to store knowledge in sources, data warehouse to store certain sources when necessary).

Of course, many complex problems still remain to achieve such a mediation but this approach presents several advantages. Integration is processed only on the schemas of the results and not on the entire schemas of all potential sources. The rewriting process is more clear because it is divided in successive rewritings steps, which can be studied independently. Moreover, this approach remains compatible with the classic mediation because one can store in the metadata warehouse partially integrated schemas.

The objectives of this chapter are multiples:

- to argue in favour of a request oriented approach for the mediation in a dynamic context in order to increase significantly the performances of mediator systems;
- to propose an organisation of the mediation process in a sequence of well defined stages;
- to show how several research results can contribute to the feasibility of this approach;
- to identify the bottlenecks and to incite future research works.

The chapter is made of two sections. In section 2, we discuss the notion of information: we recall the three types of information sources and the difficulties to put interoperability into practice, we define which sorts of knowledge bases are needed in such a mediation and we characterise the different kinds of information that generate heterogeneities. In section 3, we present the request oriented mediation process and we bring various arguments in favour of its feasibility, on the basis of the most recent investigations in particular in the field of knowledge extraction and knowledge representation.

2. Preliminaries

In this section we clarify in advance some important aspects:

- the various types of sources that one can find on Web,
- the three types of knowledge bases which appear useful in order to rewrite the user request in a machine understandable form
- and the different types of heterogeneities between all the results that have to be integrated to answer the user request in a unified way.

2.1. The three types of sources

Sources reached on Web can be three types: structured, semi-structured, not structured.

Structured sources (for example: data bases with HTML interfaces) are very dynamic, since their contents is susceptible to be modified all the time. Access to data are generally made through a specific interface which allows to introduce values for search criteria. These values are free or must be chosen among a predetermined set. This kind of sources correspond to the hidden Web since their data cannot be retrieved by first generation Web search engines. Localisation of such sources can be made through the labels of their interfaces. This task necessitates specific parsers. Determining potential interest of the data needs probe queries on the data base.

Semi-structured sources (for example: XML pages, HTML pages containing lists and\or tables) are also dynamic. But their dynamic character is less stressed. Their contents vary enough

little during time because they are generally updated manually (except for the pages which are generated automatically with an underlying structured source). The semantic tagging which offers XML allows to facilitate considerably the automatic search for zones of interest. By opposition HTML offers only a syntactic tagging and an automatic search for concepts is problematic. In semi-structured sources, repetition of a same structure (list or table) can help its detection.

Not structured sources (for example : on-line files text, HTML pages containing long portions of text) are considered as static (for example the contents of an on-line article is not going to evolve during the time). For these sources the search for concepts and for associated data requires an analysis of the natural language.

Better to illustrate these three types let us consider the example of a bibliographical search. We want to be able to integrate information resulting from various sources as bibliographical data bases, lists of publications stemming from personal pages, complete on-line articles. Interest is for example to look for references to articles (from keywords) and then to look in which other publications refer found articles, either still to look for authors who wrote a lot on the subject.

Concretely, let us suppose that one likes to make a bibliographical search from words "scatter" and "gather". It will be interesting to be able to consult with the same mediator:

- Structured sources such as the DBLP bibliographical data base (http://www.informatik.uni-trier.de/~ley/db/index.html) (Figure 1) or the Waikato data base of the university of New Zealand (http://www.nzdl.org/fast-cgi-bin/library?a=p&p=about&c=csbib);
- Semi-structured sources such as the list of Marti Hearst's recent publications, accessible by his personal home page and of which the article "Reexamining the Cluster Hypothesis: Scatter / Gather on Retrieval Results" evokes quoted terms (figure 2);
- Not structured sources such as the complete text of Marti Hearst's article (figure 3).

So, a mediator who would integrate in a effective way all these sources would allow us i) to execute a search from words "scatter" and "gather" on sites DBLP and Waikato and to get back so two sets of bibliographical references ii) to localize the page of Marti Hearst's recent

Search Result

Query: title = "scatter gather"

1	EE	Douglas R. Cutting, David R. Karger, Jan O. Pedersen: Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections. SIGIR 1993: 126-134 [DBLP:conf/sigir/CuttingKP93]
2	EE	Douglas R. Cutting, Jan O. Pedersen, David R. Karger, John W. Tukey: Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. SIGIR 1992: 318-329 [DBLP:conf/sigir/CuttingPKT92]
3	EE	Marti A. Hearst, Jan O. Pedersen: Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. SIGIR 1996: 76-84 [DBLP:conf/sigir/HearstP96]
4	EE	Peter Pirolli, Patricia K. Schank, Marti A. Hearst, Christine Diehl: Scatter/Gather Browsing Communicates the Topic Structure of a Very Large Text Collection. CHI 1996: 213-220 [DBLP:conf/chi/PirolliSHD96]
5		Susumu Shibusawa, Hiroyuki Makino, Shigeki Nimiya, Jun-ichi Hatta: Scatter and Gather Operations on an Asynchronous Communication Model. SAC (2) 2000: 685-691 [DBLP:conf/sac/ShibusawaMNH00]
6		Sandeep N. Bhatt, Geppino Pucci, Arnold L. Rosenberg: Scattering and Gathering Messages in Networks of Processors. IEEE Transactions on Computers 42(8): 938-949 (1993) [DBLP:journals/tc/BhattPR93]

ACM SIGMOD Anthology - DBLP: [Home | Search |Conferences |Journals]

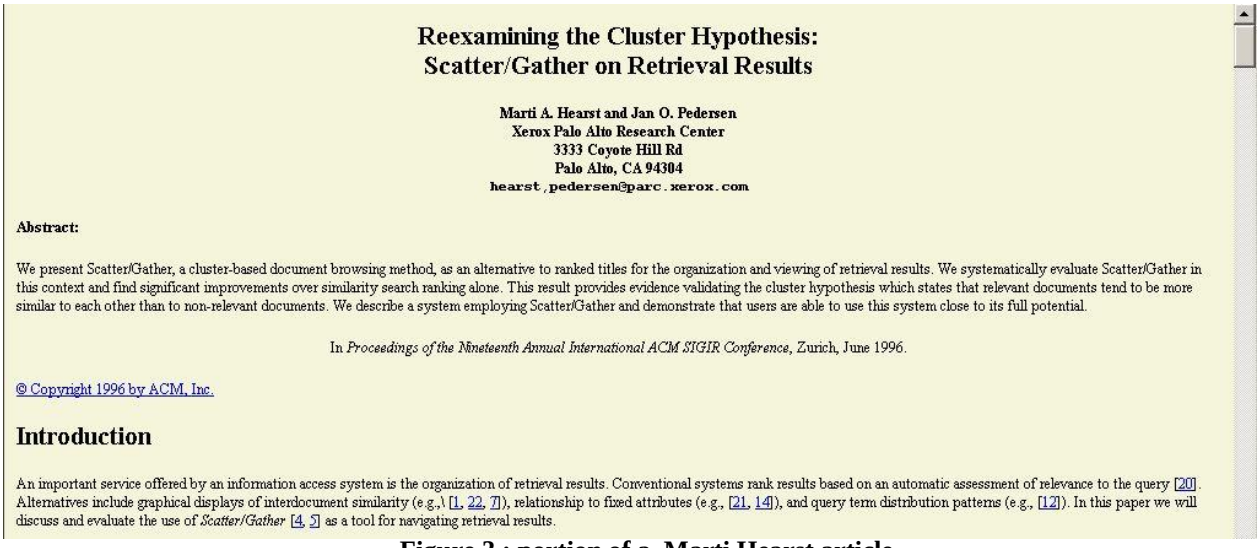
Figure 1 : example of a search result from DBLP (http://www.informatik.uni-trier.de/~ley/dbbin/title)

publications and to find so a third set of interesting references iii) to localize the document

containing Marti Hearst's complete article which can supply interesting information such as email addresses of the two authors, their laboratories, the most often quoted words, … . Very often the user will wish that the three sets of bibliographical references obtained in stages i) and ii) are merged by keeping only a single copy of each reference. This fusion is not coarse. References are not represented under the same format in the three sets. Certain information appears in a set and not in the others. A same information can be coded differently in the sources.

Hearst, M. and Karadi, C. **Cat-a-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy**, *Proceedings of the 20th Annual International ACM/SIGIR Conference* , Philadelphia, PA, July 1997.	postscript (7.8M) postscript (gz) html abstract
Hearst, M. and Karadi, C. **Searching and Browsing Text Collections with Large Category Hierarchies,** *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI), Conference Companion*, Atlanta, GA, March 1997.	html (at sigchi) abstract
Hearst, M. and Pedersen, P. **Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results**, *Proceedings of 19th Annual International ACM/SIGIR Conference*, Zurich, 1996.	postscript (1.3M) postscript (gz) html abstract
Hearst, M. **Research in Support of Digital Libraries at Xerox PARC. Part I: The Changing Social Roles of Documents**, *D-Lib Magazine*, May 1996.	html (at CNRI)
Hearst, M., Kopec, G., and Brotsky, D. **Research in Support of Digital Libraries at Xerox PARC. Part II: Digital and Paper Documents**, *D-Lib Magazine*, June 1996.	html (at CNRI)
Hearst, M., **Improving Full-Text Precision on Short Queries using Simple Constraints,** *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval* (SDAIR), Las Vegas, NV, April 1996.	postscript (1.3M) postscript (gz) abstract

Figure 2 : some publications from the page of Marti Hearst recent publications
(http://www.sims.berkeley.edu/~hearst/publications.shtml)



Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results

Marti A. Hearst and Jan O. Pedersen
Xerox Palo Alto Research Center
3333 Coyote Hill Rd
Palo Alto, CA 94304
hearst,pedersen@parc.xerox.com

Abstract:

We present Scatter/Gather, a cluster-based document browsing method, as an alternative to ranked titles for the organization and viewing of retrieval results. We systematically evaluate Scatter/Gather in this context and find significant improvements over similarity search ranking alone. This result provides evidence validating the cluster hypothesis which states that relevant documents tend to be more similar to each other than to non-relevant documents. We describe a system employing Scatter/Gather and demonstrate that users are able to use this system close to its full potential.

In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference*, Zurich, June 1996.

Introduction

An important service offered by an information access system is the organization of retrieval results. Conventional systems rank results based on an automatic assessment of relevance to the query [20]. Alternatives include graphical displays of interdocument similarity (e.g.,\ [1, 22, 7]), relationship to fixed attributes (e.g., [21, 14]), and query term distribution patterns (e.g., [12]). In this paper we will discuss and evaluate the use of *Scatter/Gather* [4, 5] as a tool for navigating retrieval results.

Figure 3 : portion of a Marti Hearst article
(http://www.sims.berkeley.edu/~hearst/papers/sg-sigir96/sigir96.html)

2.2 The three types of metadata useful to rewrite the user request

In order to transform the user request into requests that can be pose to a set of sources, three kinds of metadata sets would be useful:

- a linguistic ontology which would contain terminological knowledge about the domain (for example, Wordnet could be such an ontology),
- a domain ontology which would contain the logical definition of the domain concepts,
- a metadata warehouse which would store partial information (views for example) about the content of already used sources with respect to the domain concepts defined in the domain ontology.

For example, for our bibliographical research, we could imagine the following ontologies:

A **name** is composed with a firstname and a lastname. Synonyms of **publication** : work, issue, printing, paper An **article** is a part of a written document. A **chapter** is a part of text. One of the senses of document is "text file".	**author** ≡ ∃office Π ∀office.laboratory Π ∃name Π ∃expert Π ∃email Π ∀expert.scientific_domain **publication** ≡ ∃pub-author Π ∀pub-author.author Π ∃date Π ∃title Π ∃keywords Π ∀keywords.scientific_expressions Π ∃type Π ∀type.pub_type Π ∃field Π ∀field.scientific_domain **article** ⊂ pub_type **conf_proceedings** ⊂ pub_type **book_chapter** ⊂ pub_type **databases** ⊂ scientific_domain **distr_sys** ⊂ scientific_domain	**Source A** contains a **view A_1** which gives information about **researchers**. Many **articles** from **databases field** are listed in **source B**. **Source C** has already been asked for **names** of **researchers experts** in **distributed systems**.
Example of a **linguistic ontology** (here is a little part of Wordnet online, see www.cogsci.princeton.edu/~wn/ online/)	Example of a **domain ontology** Formalism used : *ALN* description logic Domain : scientific publications	Example of information that could be stored in the **metadata warehouse** : links between domain concepts and sources

Figure 4: Partial representation of the three types of ontologies

The role of each ontology will be presented further during the step by step process description.

These ontologies can be described through different formalisms. In figure 4, linguistic ontology is described in natural language while domain ontology is described in a formal language. Description through diagrams (ER diagrams for example) are also possible. These descriptions must be considered as equivalent. To communicate with users, a description in natural language is surely more appropriate; but to permit manipulation by programs a formal description is indispensable.

2.3 Data integration: a matter of heterogeneity

As illustrating previously, different sources solicited with a same request will always generate heterogeneous results. Let first examine the different types of information concerning a source (see figure 5 for an example with a relational source) and then we will discuss the corresponding heterogeneities. The information concerning a source can be classified according to two main distinctions. The first distinction is between intentional and extensional information. These two types correspond respectively to the source data and to all other information that are used to organise data. The second distinction lies between semantic and structural information. The first represent information which have a sense with regard to the human user. Seconds have a sense with regard to the computer program which manipulates them or formats them. Among semantic information, one can distinguish two subcategories: terminological semantic information

consisted of terms representing more or less abstracted concepts, and functional semantic information representing constraints, logical or temporal implications, functions.

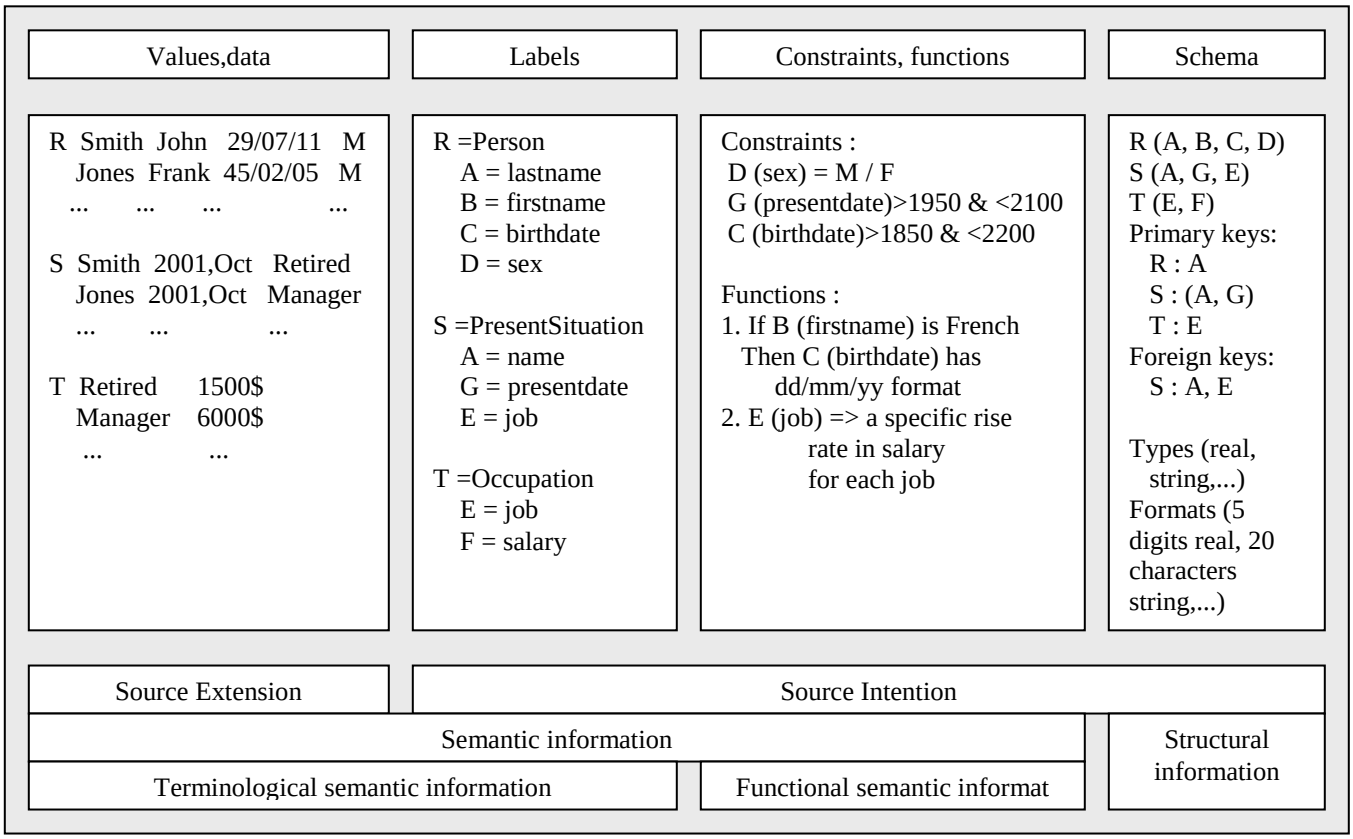


Figure 5: The different types of information on a relational example

This categorisation of available information on sources is very useful to distinguish the different sorts of heterogeneities which can arise. The first distinction between intention and

extension allows to situate the heterogeneities with respect to the sources. The second distinction (with subcategories) allows to classify the sorts of heterogeneities.

Structural heterogeneities arise when the same concept is represented by different structures in the sources. They can concern attributes, entities, relationships, primary and foreign keys. For example a date can be represented by means of a unique attribute by juxtaposing day / month / year or with three different attributes. In this category appear also *heterogeneities of sizes and types*. For example, a postal code can be represented with a character string in a source and with an integer in another one; an attribute "sex" can be represented by 0/1, or ' M ' / ' F ' or 1/2.

Semantic heterogeneities concern the meaning of manipulated concepts. These heterogeneities are more difficult to resolve than the previous ones because they often require to know the context of the sources. The *numeric semantic heterogeneities* take place between two numerical values which the mediator can compare only by having information about their contexts. For example, let us suppose that one has two grades 6,1/10 and 12,20/20 coming from two sources indicating the evaluation of the same exam. A priori one can think that this situation corresponds to a size heterogeneity, because these two values represent conceptually the same grade. Really, the heterogeneity can be of semantic nature because second format authorises a bigger precision. The *terminological semantic heterogeneities* arise from problems of synonyms and homonyms. Taking the example of (Knoblock et al 1998), to compare "Vatican" and "Holy See" requires to have information which are not necessarily available in one of the two sources in conflict. On the other hand, as (Haas et al 1999) mentions, the " Hotel of the Station " can be present in several sources without identifying the same establishment (many cities in France possess a " Hotel of the Station "?). The *functional semantic heterogeneities* arise when the mediator has to compare functions and constraints of sources in conflict. For example, one can suppose that a source possesses function " if temperature < 5°C then climate = cold ", while the other one possesses function " if temperature < 10°C then climate = cold ". In these conditions, which temperature value can the mediator associate with the term "cold" ? In our approach, the domain ontology has the role to help the mediator to solve such dilemma.

3. Presentation of the approach

The overall approach comprises nine stages (Figure 6). In this section we define each stage, we precise how existing research results can contribute and we characterise open problems.

Stage 1: Helping the formulation of the request

Formulation of a request can be made in different manners : keywords forms, natural languages, formal request language for semi-structured data (for example LOREL) or for structured data (for example SQL).

Let us first discuss formulation in a formal request language like OQL. We will illustrate the subject with a small example: the search for publications on data bases with their titles, the names of their authors and the e-mail addresses of these authors. Having some knowledge about the domain ontology (see figure 4 for a partial description), a user can issue a formulation involving concepts of the domain and relationships between these concepts like the following one:

```
SELECT struct(a: x.Title, struct(b: y.Name, c: y.Email))  FROM x in Publication,
    y in Author(Publication)  WHERE x.Field = 'Database'
```

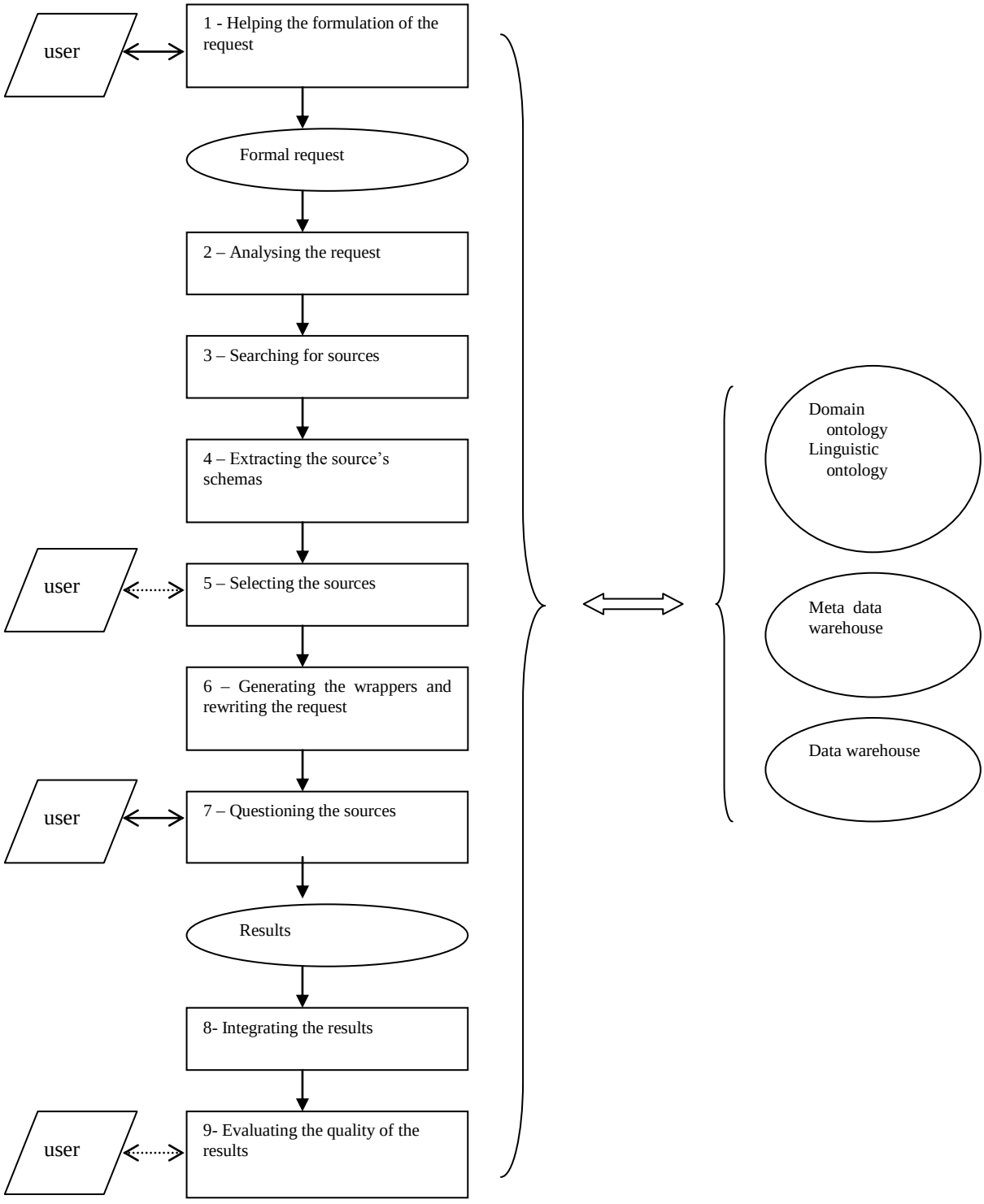


Figure 6 : The overall approach

The system controls this formulation in order to force the user to use only names and values of the domain (through, for example, a set of predefined choices via lists or other HTML form elements). More exactly the system must control that Publication, Author are main concepts of the domain, that Title, Name are other concepts linked to the previous ones (attributes or slots), that Author(Publication) is an allowed form giving the authors (their idendifiers) of a publication (in the present case it is a functional view of the relationship linking a Publication and an Author in the domain ontology). Instead of Publication it can permit the use of a synonym (paper for example) or of a more specific concept (book_chapter for example). It must control also that "database" is an acceptable value for the publication field.

A formulation in natural language is obviously more interesting for the user but its analysis by the system is much more difficult. This analysis necessitates the use of elaborated metadata and natural language inference techniques. Certain number of works focused on this formulation, possibly by restricting it to facilitate its analysis (Grishman 1997, Turmo et al 1998).

Stage 2: Analysing the request

The request analysis must first allow to discover the various types of objects of the domain concerned by the request and the possible links (associations, specializations, …) between these types of objects. The types of objects prefigure the sources which it will be necessary to seek and links prefigure the joins which it will be necessary to make between results. For example, the previous request involves two different types of objects (Publication, Author) with an association between these two types (a publication has authors). One will so be able to interrogate sources which include, either publications, or persons, or both publications and persons. It will be necessary to assemble (join) the results obtained from these sources. The analysis must also allow to discover the conditions of selection. Finally the analysis can also try to determine the indispensable elements and the non indispensable elements. In the previous example, the name of an author is essential because it establishes the support of the join. So, it should necessarily appear in the selected sources. This stage can be considered as the first part of the rewriting of the request (cf *Stage 6*).

To the request one can associate a conceptual schema which represents the types involved in the request and their relationships. So, for the previous request, the conceptual schema would consist of the two types Publication and Author and of an association between these two types (Figure 7). Cardinalities are inferred from the domain ontology.

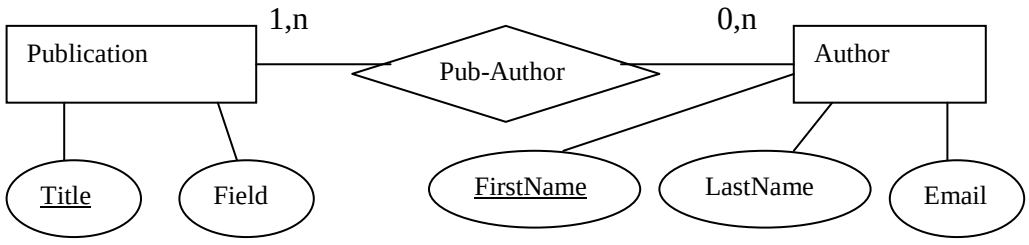


Figure 7 : Schema of the request

request ≡ publication Π ∃pub-author Π ∀pub-author.(author Π ∃name Π ∃email) Π ∃title Π ∃field

Figure 8 : Same request expressed in ALN description logic

Stage 3: Searching for sources

This step has the goal to search sources which are capable of providing some of the objects identified in the previous stage. One must at first determine if some of the sources referenced in the meta data ware house can agree. For these sources one already has sufficient knowledge (conceptual schema or equivalent information) to situate easily their contribution. Search will also (and especially) concern the entire of the Web or a subset of the Web (for example the Web of a particular country). In that case, one can imagine a search procedure which consists to make the connection between concepts of the request and those of a source. Possibilities offered with search engines can be exploited. More exactly one will try to write a search expression involving main concepts of the request obtained in stage 1 (those associated to the types of objects). In case of unsuccessful search, one will be able to try again with a subset of main concepts (when request contains several types of objects). For example with the request of stage 1 one will tempt at first a search with the expression "publication AND author". In case of not convincing result one will try again with the expression "publication OR author". One can exploit with some profit the information coming from the domain ontology. One will begin by considering first the concepts of the higher level (publication and its synonyms) and then the concepts of more specialised levels (article or conf_proceedings or book_chapter). It is interesting to notice that some works (Bergamaschi 1997, Craven et al 1998, Carciolo et al 2000, Cohen 2000) are devoted to the extraction of concepts and of connections between concepts in textual sources. These works could usefully contribute to improve the efficiency of search engines and justify the growing notion of "Semantic Web" (Berners et al 2001).

Stage 4: Extracting the source's schemas

In this stage, one tries to extract the schema of each from the sources identified in the previous stage. For poor structured sources, the main difficulty is to separate metadata and data. Let us note that even there, the domain ontology will be able to bring a substantial aid.

To extract the schema from a structured source is relatively coarse if this one is directly accessible (the data dictionary will supply in this case almost all the useful knowledge). If the source is accessible only through a specific interface (for example an HTML form), the task complicates and it is necessary to analyze the form by means of a suited parser. In this form appear only metadata. One will be able to confirm the interest of these metadata by comparing it with the request concepts (the domain ontology supply with synonyms and homonyms for each concept). It is necessary then to try to extract relations between these metadata. The form layout can give very useful information. To raise residual ambiguities one can interrogate the source and look how are organized the data. For example data mining techniques can be applied on these data to extract functional and inclusion dependencies (see for example Lopes et al 2001).

To extract the schema from a semi-structured source is a task which can be automated. Appeal to XML contributes to formalize and to simplify treatments. An XML document is either

valid, well then the data schema is obtained from the associated DTD, or well formed, in which case a parser can easily extract the schema. Several works were interested in this problem (Hammer et al 1997, Nestorov 1998, Bergamaschi et al 1999). In

To extract the schema from a non-structured source is, on the other hand, a much more delicate task which is similar to the analysis of a document wrote in natural language. The notion of schema for such a source is not always relevant. Some of the techniques quoted in stage 1 can be also used here. The majority of the sources accessible from the Web belong to this category. This situation justifies that an important number of works (Cohen 2000, Craven et al 1998, Freitag 1998, Soderland 1999) is interested in their analysis. But one can think that the generalization of XML is going to change quickly this situation. As mentioned in (Doan et al 2000), it is important to differentiate structure extraction and semantics extraction. XML tags and annotations techniques can greatly facilitates semantics extraction. For example in (Staab et al 2001), methods and tools for annotating Web pages with metadata are presented.

Once a schema has been found, it is stored in the metadata ware house for further uses.

Stage 5: Selecting sources

Now the schemas of sources are known, it is possible to determine more exactly the potential contribution of every source to the request. Sources considered not enough relevant are no longer considered. To make this selection one can try to measure the similarities which exist between the schema of the request and the schema of the source (Song et al 1996, Cohen 1998). Estimation of these similarities requires to take into account semantic equivalencies supplied by the domain ontology. One could also imagine that the user allocates weights to the concepts of his request so as to measure more directly the contribution of every source: only the sources for which the contribution exceeds a certain threshold would be considered.

When several sources can supply the same results, arises also the problem to choose among these alternative sources the one that will offer the best plan of execution. Certain works, such as (Knoblock et al 1998), were interested in such a problem. It raises difficult questions: equivalence of two sources in terms of information capacity; availability and power of a source.

Stage 6: Generating the wrappers and rewriting the user request

One knows now the sources and the data which one can obtain from each of them. The request of each of these sources is going to be posed through wrappers. A wrapper accepts a request and sends back results coming from the corresponding source. Every wrapper produces naturally its results in an unique format to facilitate following integration. For reason of standardization and efficiency, we suggest that each wrapper transforms its results into an XML form. This problem drew the attention of several researchers (see for example Liu et al 2000). In our approach, the domain ontology allows to impose a unified semantics for the XML tags, avoiding so semantic heterogeneity during integration.

To generate a wrapper we must know its entry i.e. a component of the user request. The user request must be rewritten for each wrapper. Numerous works were interested in the rewriting of request and effective solutions exist (see for example Papakonstantinou et Vassalos 1999). Let us note the interest of a formalism based on the Description Logics to express request and schemas: in this context there exist effective rewriting algorithms and powerful environments (Calvanese et al 1998, Lattes et Rousset 1998).

The generation of a wrapper can be a very heavy task. So, several works were dedicated to the automation of this task (Ashish et Knoblock 1997, Kushmerick et al 1997, Grieser et al 2000). The automation is especially possible for structured and semi-structured sources. It is much more difficult for non-structured sources. In certain cases one can have interest to download the source and to transform it under an XML form by using the techniques previously quoted. For our approach, we suggest a data ware house to be able to store the results of these transformations and to reuse them later for other requests.

As mentioned by (Ashish et Knoblock 1997), when building wrappers, it is important to separate the tasks that are specific to a particular Web source such as structuring the source, and tasks which are repetitive for any source such as generating a parser from the structure of a page. (Grieser et al 2000) argue also that a given wrapper can be applied to different documents.

Stage 7: Questioning the sources

This stage consists in activating each of the wrappers and in getting back results. Since we need a real time processing, it is necessary to foresee decisions to set if a source is not available or presents too big delays for answering. Interactions with the user can help in these decisions.

Stage 8: Integrating the results

It is a question now of integrating XML documents already unified in their semantics. There are two levels of integration. The first concerns the integration of objects of similar type (having a similar root for the XML document) resulting from the interrogation of the sources. Objects in double must be discovered and eliminated after having resolved the problems of synonymic values in data (by using the domain ontology). Second level concerns composed objects resulting from some joins. Such objects can be redundant because already obtained directly from another source. Even these objects must be discovered and eliminated.

Several works address the problem of cleaning the data and merging duplicates and can be proved also very useful in our approach. Methods to detect duplicates items and to derive cleaning rules are proposed in (Hernandez et Stolfo 1998). AJAX (Galhardas et al 2000) is a tool that supports clustering and merging duplicates. A clustering algorithm to reduce inconsistencies is presented in (Lujan-Mora, Palomar 2001).

These algorithms are generally very time consuming. A good compromise must be found between quality of results and acceptable response time.

Stage 9 : Evaluating the quality of results

System seeks the user who can indicate the aptness of results with regard to his request and the problems of incoherence or ambiguity which can remain. System readjusts possibly its information bases according to these indications. The evaluation of the results quality is a problem which is also studied in domains like Data Quality and Data Cleaning.

4. Conclusion

In this chapter we suggested a request oriented approach for the interoperability of heterogeneous sources in a dynamic context. This approach does not require the integration of the schemas of the sources. The user request is rewritten according to the capacities of each from the selected sources. Results are expressed under a semi-structured form before integration. Process is driven thanks to different information and knowledge bases : domain ontology and linguistic

ontology for helping the formulation of the request and for resolving the semantic heterogeneity, meta data warehouse to store schemas of sources, data warehouse to load and transform unstructured sources.

This approach is decomposed into nine stages of which we have argued feasibility in the light of the most recent search results. Some of the mentioned problems have already received suitable solutions. Others remain very open. The most difficult problem is undubitably the extraction of the schema from a new source. In the current state of the art, a complete automation of this extraction is not possible for non-structured sources. Very numerous works are interested in this problem and one can hope for interesting results in the close future. Another difficult problem is the automatic generation of wrappers.

This approach does not question classic mediation with preliminary integration of the schemas from the involved sources. One has advantage to incorporate into the metadata warehouse partially integrated schemas which are considered relevant for the domain and which are associated with stable sources. Such schemas can be exploited to search more effectively some composed objects of a request or to give the users a more realistic view on the data. Naturally, the processes for generating the wrappers and for rewriting the request have to take into account the use of these schemas.

5. Bibliography

Ashish N. and C. Knoblock (1997); Semi-automatic wrapper generation for Internet information sources; Proc. of the Int. Conf. On Cooperative Information Systems

Bergamaschi S. (1997); Extraction of Informations from Highly Heterogeneous Sources of Textual Data; Proc. of the First International Workshop on Cooperative Information Agents (CIA97) (Ed. Springer) (pp. 42-63)

Bergamaschi S., Castano S. and Vincini M. (1999); Semantic integration of semi-structured and structured data sources; SIGMOD Record, Vol. 28, No. 1 (pp. 54-59)

Berners-Lee T., Hendler J. and Lassila O. (May 2001); The Semantic Web; Scientific American

Calvanese D., De Giacomo G., Lenzerini M., Nardi D. and R. Rosati (1998); Description logic framework for information integration; Proc. of the 6th Int. Conf. On the Principles of Knowledge Representation and Reasoning (KR98), (pp. 2-13)

Carchiolo V., Longheu A. and Malgeri M. (2000); Extracting Logical Schema from the Web; Proc. of the PRICAI Workshop on Text and Web Mining (pp. 64-71)

Craven M., Di Pasquo D., Freitag D., McCallum A., Mitchell T., Nigam K. and S. Slattery (1998); Learning to extract symbolic knowledge from the world wide web; Proc. of the Fifteenth National Conference on Artificial Intelligence

Cohen W.W. (1998); Integration of Heterogeneous Databases without Common Domains using Queries Based on Textual Similarity; Proc. of the ACM SIGMOD Int. Conf. on Management of Data

Cohen W.W. (2000); Automatically Extracting Features for Concept Learning from the Web; Proc. of the 17th International Conference On Machine Learning, (ed. Morgan Kaufmann) (pp. 159-166)

Doan A., Domingos P., and Levy A. (2000); Learning Source Descriptions for Data Integration; Proc. of the WebDB2000 Conference, Dallas

Freitag D. (1998); Information extraction from html: Application of a general learning approach; Proc of the Fifteenth Conference on Artificial Intelligence (pp. 517-523)

Galhardas H., Florescu D., Shasha D. and Simon E. (2000); AJAX: An extensible data cleaning tool; Proc. of the 2000 ACM SIGMOD Conference, Dallas

Garcia-Molina H., Papakonstantinou Y., Quass D., Rajaraman A., Sagiv Y., Ullman J., Vassalos V. and Widom J. (1997); The Tsimmis approach to mediation : Data models and languages; Journal of Intelligent Information Systems (JIIS), Vol. 8, No. 2 (pp. 117-132)

Grieser G., Jantke K.P., Lange S. and Bernd Thomas B. (2000); A Unifying Approach to HTML Wrapper Representation and Learning ; Proc. of the 3rd Int. Conference DS 2000, Kyoto, Japan (pp. 50-64)

Grishman R. (1997); Information extraction: Techniques and challenges; Information Extraction (ed. M.T. Pazienza), Springer-Verlag, LNCS (pp. 10-25)

Hammer J., Garcia-Molina H., Cho J., Aranha R. and A. Crespo (1997); Extracting Semi-structured Information from the Web; Proc. of the Workshop on Management of Semi-structured Data (PODS/SIGMOD), Tucson, Arizona

Haas J.M., Miller R.J., Niswonger B., Roth M.T., Schwarz P.M. and Wimmers E.L. (1999); Transforming heterogeneous data with database middleware : Beyond integration; IEEE Data Engineering Bulletin, 22(1) (pp. 31-36)

Hernandez M. and Stolfo S. ((1998); Real world data is dirty : Cleaning and the Merge/Purge problem; Journal of Data Mining and Knowledge Discovery, 2(1) (pp. 9-37)

Hull R.(1997); Managing semantic heterogeneity in databases: A theoretical perspective; Proc. of the Symposium on Principles of Database Systems (PODS), Tucson, Arizona (pp. 51-61)

Kedad Z. and E. Métais (1999); Dealing with Semantic Heterogeneity During Data Integration; Proc of the International Entity Relationship Conference (pp. 325-339)

Knoblock C.A., Minton S., Ambite J.L., Ashish N., Muslea I., Philpot A.G. and Tejada S. (1998); Modeling web sources for information integration; Proc. of the Fifteenth National Conference on Artificial Intelligence (ed. AAAI Press / The MIT Press), (pp. 211-218)

Kushmerick N., WeId D., and R. Doorenbos (1997) : Wrapper Induction for Information Extraction ; Proc. of the Fifteenth International Joint Conference on Artificial Intelligence

Lattes V. and Rousset M.C. (1998) ; The use of Carin language and algorithms for information integration : The Picsel project ; Proc. of the ECAI'98 Workshop on Intelligent Information Integration, Brighton

Liu L., Pu C., and W. Han (2000) ; XWRAP: An XML-enabled wrapper construction system for web information sources ; Proc. of the International Conf. on Data Engineering (pp. 611-621)

Lopes S., Petit J.M. and F. Toumani (2001); Discovering interesting inclusion dependencies : Application to logical database tuning; Information Systems, to appear

Lujan-Mora S. and Palomar M. (2001); Reducing Inconsistency in Integrating Data from Different Sources, Proc. of the IDEAS 2001 Conference, Grenoble, France (pp. 209-218)

Papakonstantinou Y. and V. Vassalos (1999) ; Query rewriting for semi-structured data ; Proc. of the ACM SIGMOD Int. Conf. on Management of Data (pp. 455-466)

Nestorov S., Abiteboul S., and Motwani R. (1998); Extracting schema from semi-structured data ; Proc ACM SIGMOD Int. Conference on Management of Data, Seattle (pp. 295-306)

Saltor F. and Rodriguez E. (1997); On Intelligent Access to Heterogeneous Information; Proc. of the 4th KRDB Workshop, Athens, Greece (pp. 15-1, 15-7)

Soderland S. (1999); Learning information extraction rules for semi-structured and free text ; Machine Learning, Vol. 34, No.1-3 (pp. 233-272)

Song W.W., Johannesson. P. and Bubenko J.A. (1996); Semantic similarity relations and computation in schema integration; Data and Knowledge Engineering, Vol. 19 (pp. 65-97)

Staab S., Maedche A. and S. Handschuh S. (2001) ; An annotation framework for the semantic web; Proc. of the First Workshop on Multimedia Annotation, Tokyo, Japan

Turmo J., Catal N. and Rodriguez H. (1998); TURBIO: A System for Extracting Information from Restricted Domain Texts; Proc. of IEA/AIE'98, LNAI 1415 (pp 708-721)

Wiederhold G. (1992); Mediators in the architecture of future information systems; IEEE Computer, Vol. 25, No.3 (pp.38-49)