A Methodology to Assist with a Distributed Information Discovery Process for Autonomous Databases

A. Zisman

Imperial College Research Report No. DOC 97/03

Department of Computing
Imperial College
180 Queen's Gate, London SW7 2BZ - UK
Email: az@doc.ic.ac.uk

March, 1997

Abstract

In this technical report we discuss a methodology and a support tool to assist the coordinator of a federation with the construction and evolution of hierarchical information structures. The definition of the terms composing the hierarchical information structures is based on the interests of the users and the applications, and the information that each database system shares with the other components. Therefore, the different group names refer to the types of databases participating in the federation. The other levels are related to the entity names, attribute names, class names, object names, and instances of the databases. The methodology consists of constructing a hierarchical information structure by incremental addition of the participating database systems. The support tool assists with the automation of some steps during construction and evolution of the structures.

Keywords

Heterogeneous databases, autonomous databases, global schema, interoperability, federation, information discovery, evolution, hierarchical information structure.

1 Introduction

The development of database management systems and the existence of large number of databases in an organisation require the sharing and exchange of data between various database systems. Therefore, it is necessary to allow simultaneous manipulation and access of different databases (local and remote access), preserving their autonomy.

An important aspect of interoperability among a large number of database systems is *information discovery*: the location and identification of information which is related, similar, identical or relevant to the requested data. Many existing approaches assume that the database systems know about the contents of the other participating databases. However, when dealing with a large number of database systems this assumption is not reliable.

Other approaches use centralised structures, like *repositories* and *dictionaries*, with information related to the available data and their location. Examples are found in Remote-Exchange approach [1], federated architecture [2], and Mariposa [3]. However, in an environment with a large number of database systems a centralised structure generates potential bottlenecks, is prone to failures, violates the autonomy of the database systems, and does not guarantee privacy and confidentiality of the shared data. In an effort to avoid centralised structures and to help educate the users about the information space, other approaches were proposed [4, 5, 6]. However, these approaches do not specify how to perform sharing and data access after location.

We have proposed an alternative to the problem of building an integrated global schema when dealing with a large number of databases [7, 8, 9]. The approach permits distributed information discovery when interoperating with a large number of database systems. Our aim is to allow "naive" users to access and manipulate local and remote data. The idea is to perform the complete execution of a query: data request, database location and data access, in a distributed way, avoiding the use of integrated schemas, centralised structures and broadcast to all the databases in the system. The approach attempts to preserve the autonomy of the databases and supports evolution of the system in terms of adding and removing databases. The idea is to limit the search to a group of databases that have data related to the requested information. A database can contain either the requested data or information about another database that possibly holds the required data. For a given query, the approach guarantees that the group of accessed databases are able to share and exchange data with the requester. Thus, it is not necessary to execute negotiations between the requester database and the one that contains the data. The process is recursive, executed in parallel and avoids cycles, i.e., avoids access to a database more than once for the same request.

We proposed an architecture to assist in the distributed discovery process [7, 8]. In the architecture the databases are arranged into *federations*. A federation is a set of databases willing to share data with each other. It is formed based on the shared data of each database and the databases that are allowed to access this shared data. Therefore, a database can participate in different federations. Inside a federation the databases are organised into *groups*. The idea of groups is to make the universe of search smaller, facilitating the information discovery process. A group of databases is formed based on the type of data shared by these components (context classification). To assist with the information discovery process we propose to use a subject-based hierarchical information structure. This structure is composed of the names of the groups of a federation and

specialised terms, forming a natural hierarchy of names. Associated with these terms are references to the concerned databases. Each database contains a hierarchical information structure for each federation in which it participates.

The definition of the terms composing a hierarchical information structure requires human assistance. In order to reduce human participation when constructing and evolving the hierarchical information structures, we propose a methodology and a support tool. For a federation, the initial hierarchical information structure of the participating database systems is constructed by incremental addition of the database systems, in any order. As a result of the incremental addition of the participating database systems, the support tool presents to the coordinator of the related federation a version of a possible hierarchical information structure to be initially used in the federation. The coordinator of the federation may make changes to this first version, generating a final configuration of the hierarchical information structure. Execution of changes is assisted by the support tool. The hierarchical information structures evolve as a consequence of system modification (addition and removal of databases) and as needed to adjust the structure to the requirements of the users and applications. This is done based on statistical information related to the use of the different branches of the hierarchy, and on recommendations from users when they identify the absence of terms matching their request.

2 Creating the Hierarchical Information Structure

We suggest the definition of the different groups in a federation, together with the related specialised terms, based on: (a) the part of the schema of a database that is available to the federation; (b) the values of the data (instances) related to the available schema; (c) the existing applications of each database system; (d) the 'new' possible applications and remote queries that may be performed; and (e) the different interests of the users. Therefore, the database systems are organised and classified in a federation based on the requirements of the users, and not only on the data that each database system shares with the other components in the system.

In the proposed methodology, for each federation, the terms composing a hierarchical information structure are defined in an interactive way with the coordinator of the related federation and the DBAs of the participating database systems. Initially, for each federation in which a database system participates, its DBA is responsible for defining three different types of information related to the respective part of the local schema being shared. These different types of information form three parts, as follows:

1. *database-is*: this part is composed of terms describing the type of the database system. For instance, *hospital*, when it is a hospital database system; *Ophthalmology*, when it is a database system related to the Ophthalmologic department; *Child Care*, when it is a database system related to the Child Care department. In a first attempt, the terms composing the *database-is* part are used to define different groups of database systems, forming the first level of a hierarchical information structure. It is possible to have more than one term in the *database-is* description of a database system. This situation occurs when the database system is related to different types of information. For instance, it is possible to have the terms *Radiology* and *Imaging*

composing the *database-is* part of a database system related to the Radiology and Imaging department.

2. *database-has*: this part contains terms related to the information that a database system is sharing with the other components of a related federation. The terms are specified by the DBA of the database system and are based on the part of the schema and instances being shared, e.g. entity names, attribute names, class names, object names, and data values. An example is described below. Not all entity, attribute, class, and object names composing the schema being shared are used to describe the actual information that the database contains. However, specification of suitable terms is a human task and the DBA is responsible for identifying these terms.

3. *database-wants*: this part contains terms describing the type of information that the users are interested in accessing. These terms are specified by the DBA of the associated database system, based on existing applications of the local database and on 'new' applications that the related federation supports. The terms in this part are used by the coordinator of a federation during construction, evolution and refinement of a hierarchical information structure. The coordinator uses these terms to check the feasibility of the information in the hierarchical information structure.

In order to illustrate the definition of the terms composing the *database-is*, *database-has* and *database-wants* parts, suppose db_{CHI} a relational database system of a Child Care department. Consider the part of the local schema of db_{CHI} being shared, with the relations and attribute names as presented in figure 1.

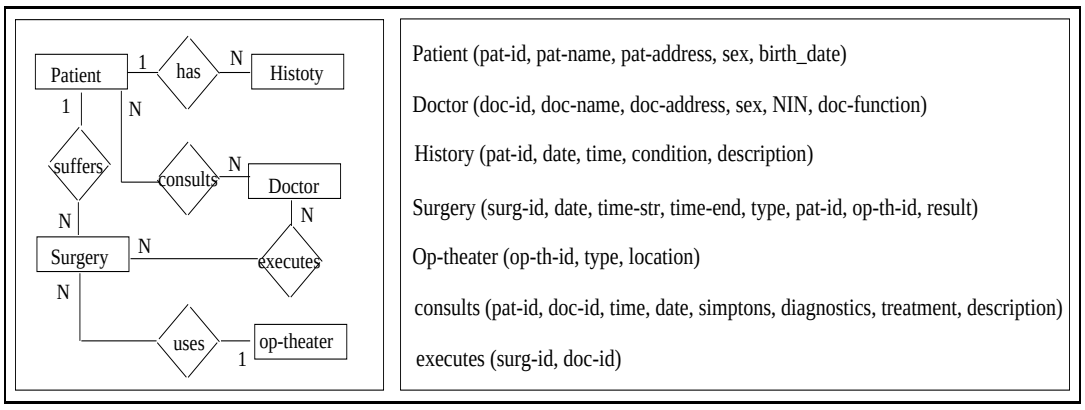

Figure 1: Example of the part of the schema being shared related to db_{CHI}

In this case, *database-is* part has the term Child-Care denoting the type of the database. On the other hand, the DBA of db_{CHI} specifies the terms *clinic, history* and *surgery* to compose the *database-has* part. The identified terms denote that db_{CHI} is sharing data related to clinical and historical aspects of the patients and surgeries performed in the department.

Let us assume that the users of db_{CHI} want to compose a federation with other database systems in order to be able to access data related to children in the areas of: Oncology, Ophthalmology, Pulmonology, Cardiology, Accidents, Orthopaedics, Transplants,

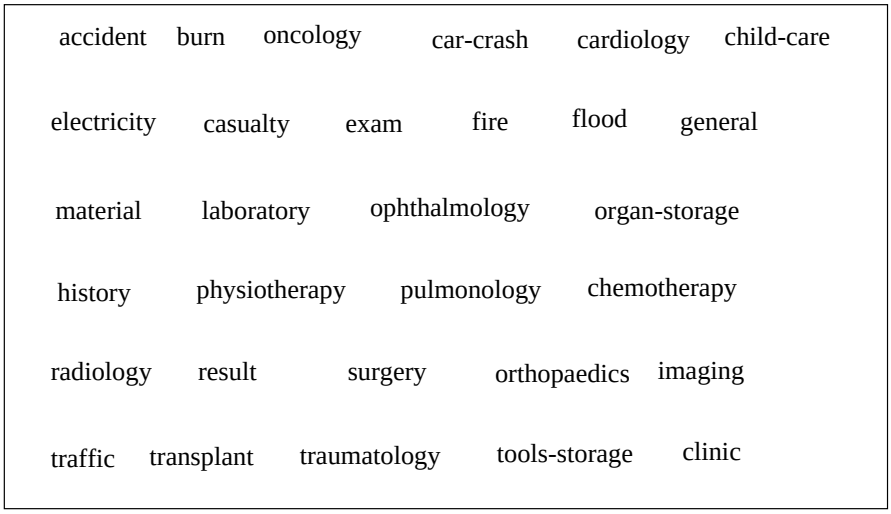

Figure 2: Example of the structure with the standard terms

and Exams. In addition, the users also want to perform existing applications related to clinical and historical aspects of both patients, and surgeries. Therefore, a possible list of terms composing the *database-wants* part is: *oncology, ophthalmology, pulmonology, cardiology, exam, results, orthopaedics, accident, clinic, surgery, transplant, organ-storage.*

In order to allow automatic execution of some steps when building a hierarchical information structure, we suggest standardisation, as much as possible, of the terms used to describe the *database-is*, *database-has* and *database-wants* parts. Similar to the concept of ontologies [10, 11], the idea is to have a structure with standard terms and associated synonyms, for each federation, as presented in figure 2. This structure is dynamic and evolves with the definition of the different parts. Initially, each federation will have an empty structure. Therefore, when a DBA of a database system is defining the terms to compose the *database-is*, *database-has* and *database-wants* parts, s/he consults the structure to verify if a certain term, or a similar one, has been used before. When the respective term, or a synonym is not found in the structure the term is added to the structure in order to be used in a similar situation in the future.

The process related to the creation of a hierarchical information structure is described below. A complete example is presented in the end of this section. In order to facilitate the explanation of the process consider *hospital-A* a general hospital composed of relational multidatabases, where each department has a different database system; and *hospital-B* a Child Care hospital composed of a centralised relational database system. Let us assume that db_{HB} is the database system related to *hospital-B*, db_{CHI} is the database system related to the Child Care department of *hospital-A*, and db_{ONC} is the database system related to the Oncology department of *hospital-A*. Suppose db_{HB}, db_{CHI} and db_{ONC} participating in a federation sharing and exchanging data related to children, with their *database-is*, *database-has* and *database-wants* parts, as follows:

1. db_{HB}

   database-is: *child-care*

   database-has: *accident, exam, clinic, history, surgery, transplant*

database-wants: *general child-care, transplant, organ-storage, surgery, clinic*

2. db_{CHI}

   database-is: *child-care*

   database-has: *clinic, history, surgery*

   database-wants: *general child-care, oncology, ophthalmology, pulmonology, cardiology, exam, results, orthopaedics, accident, clinic, surgery, transplant, organ-storage*

3. db_{ONC}

   database-is: *oncology*

   database-has: *clinic, history, chemotherapy, surgery*

   database-wants: *exam, results, clinic, surgery*

During the construction of a hierarchical information structure, for each incrementally added database system, new groups and specialised terms are eventually defined. In the first step, the coordinator of the federation informs the support tool about a database system participating in the federation. For example, consider database system db_{HB}. The term composing the *database-is* part of db_{HB} is used as a group name in the hierarchical information structure. The terms describing the *database-has* part of db_{HB} are used as specialised terms of the created group, having db_{HB} associated with them. Figure 3 presents the terms composing the hierarchical information structure after the addition of db_{HB}.

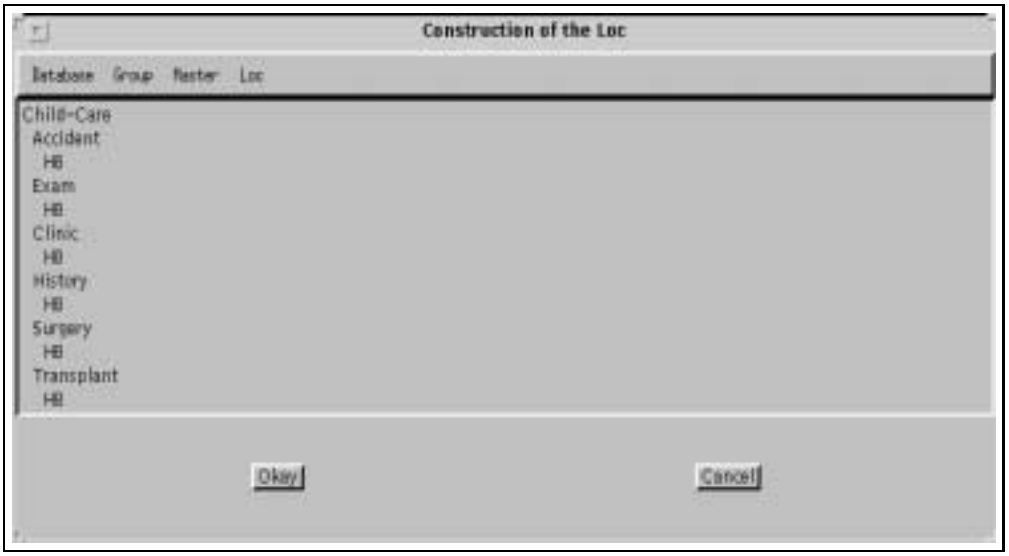

Figure 3: The hierarchical information structure after the addition of db_{HB}

In the next step, the coordinator notifies the support tool of another database system participating in the federation. Suppose db_{CHI} is this database system. The support tool compares the terms in the *database-is* part of db_{CHI} with the already existing group names, i.e. the group created by the addition of db_{HB}. If a term in the *database-is* part of db_{CHI}

is equal to one of the existing group names, db_{CHI} is added to this group. Otherwise, new groups are formed and the terms in the *database-has* part are used as specialisations. In the situation in which the database system is added to an existing group in the hierarchy, the terms in its *database-has* part are compared with the specialised terms in this group. Therefore, the database system is associated with some existing specialised terms or new specialisations are created in the group. For the addition of db_{CHI}, this database is added to the already existing Child-Care group, as presented in figure 4.

Figure 4: The hierarchical information structure after the addition of db_{HB} and db_{CHI}

After adding and classifying two or more database systems in the hierarchical information structure, the support tool performs automatic 'cross referencing' among the existing terms. The idea is to compare each existing group name with the specialised terms of the other groups. Consider a group with name t_l. Consider another group with name t_p, which has a specialised term t_l. In this case, the name t_p is added in the group named t_l as a specialised term. In the situation that the term t_l in the group named t_p has associated specialised terms, these terms are added to the group named t_l as specialisations of the term t_p. A possible extension of the support tool is to allow automatic 'cross references' between terms that are synonyms and homonyms.

In order to illustrate the automatic 'cross referencing', consider the addition of database system $db_{A\&C}$ related to the Accident and Casualty department of *hospital-A*. Assume that the *database-is* part of $db_{A\&C}$ contains the terms *Accident* and *Casualty*. Figure 5 presents the execution of 'cross referencing' after the addition of $db_{A\&C}$ to the hierarchical information structure shown in figure 4. Note the term Child-Care in the Accident group, due to the fact that the group Child-Care has Accident as a specialised term.

The building process continues while there are database systems to be added. In the next step, another database system is added and, eventually, new groups and new specialised terms are specified. Whenever a new database system is added, 'cross references' are automatic created. Figure 6 presents the hierarchical information structure after the addition of db_{ONC}. For simplicity, in this example we are not considering the addition of $db_{A\&C}$, as presented in figure 5.

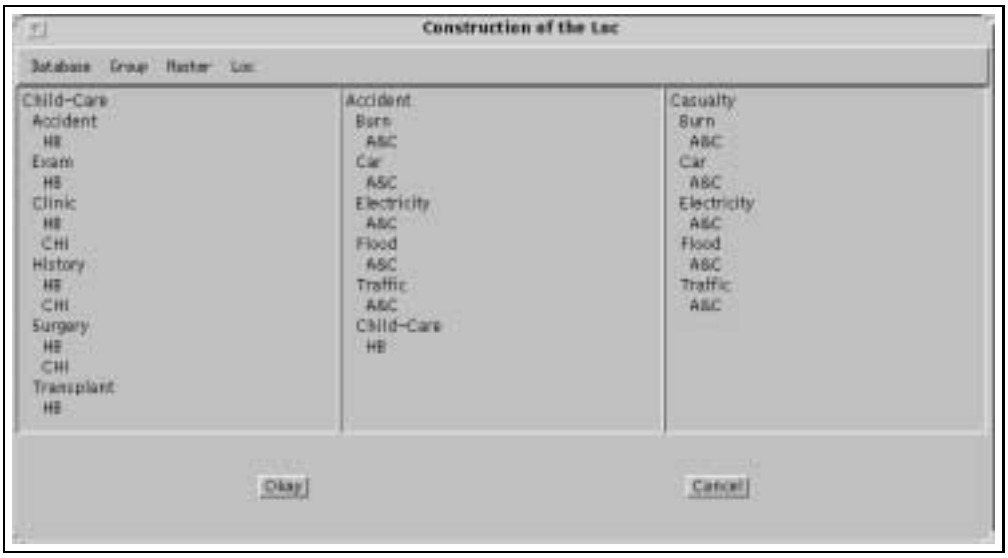

Figure 5: Example of 'cross references' after the addition of $db_{A\&C}$

When all the database systems have been included, the coordinator can ask the support tool to perform another type of cross referencing, called 'terms cross referencing'. In this case, the specialised terms of each group are compared to the specialised terms of the other groups. In the event that a term in one group is equal to a term in another group, a new group is created having this term as the name of the group. The original names of the groups with the coincident terms are defined as specialised terms in the created group. The 'terms cross references' can be created at any time during the construction process. An example is presented in figure 7, related to the 'terms cross references' between Child-Care and Oncology groups. As a result, three new groups are created, named: Clinics, History and Surgery.

After the construction of the first version of the hierarchical information structure to be used in a federation, the coordinator of the related federation analyses this version and performs any necessary modifications. Therefore, some groups may be combined, and other groups eliminated or created. Another possibility is to have unnecessary specialised terms associated with a group. This situation occurs when all the specialised terms in the group are related to the same set of database systems, not providing any distinction. The different levels of specialisations of a group depend on the number and variety of database systems associated with the group. Finally, the coordinator specifies the master of each group and the creation process is concluded.

Example

We now present a more complete example to illustrate the process of creating a hierarchical information structure. Let us assume the situation described above where database systems of *hospital-B* and of some departments of *hospital-A* want to co-operate by sharing and exchanging data related to children. The DBA of each participating database system defines the different parts related to each database system. The standard terms used to assist with the choice of terms composing the *database-is*, *database-has* and *database-*

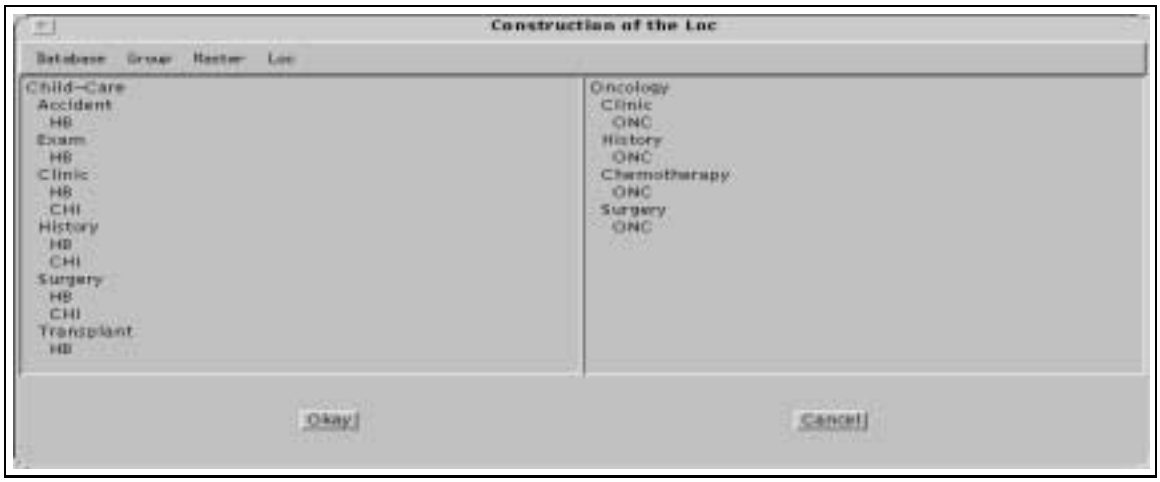

Figure 6: The hierarchical information structure after the addition of db_{HB}, db_{CHI} and db_{ONC}

wants parts is shown in figure 2. Assume db_{HB}, db_{CHI} and db_{ONC} the database systems related to *hospital-B*, and Child Care department and Oncology department of *hospital-A*, respectively, as defined before. The other participating database systems are defined as follows:

1. $db_{A\&C}$

   database-is: *accident, casualty*

   database-has: *accident-types: burn, car, electricity, flood, traffic*

   database-wants: *exam, results, surgery, clinic*

2. $db_{C\&P}$

   database-is: *cardiology, pulmonology*

   database-has: *clinic, history, surgery*

   database-wants: *exam, results, surgery, oncology, general child-care, organ-storage, cardiology, pulmonology, transplant*

3. db_{LAB}

   database-is: *laboratory*

   database-has: *exam, results, material*

   database-wants: *exam, results, material*

4. db_{OPH}

   database-is: *ophthalmology*

   database-has: *clinic, history, surgery*

   database-wants: *exam, results, transplant, general child-care, surgery, oncology, organ-storage*

Figure 7: The hierarchical information structure after the creation of 'terms cross references'

5. db_{OTP}

   database-is: *orthopaedics, traumatology, physiotherapy*

   database-has: *clinic, surgery, tools-storage*

   database-wants: *surgery, tools-storage, clinic, accidents*

6. $db_{R\&I}$

   database-is: *radiology, imaging*

   database-has: *exam, results, material*

   database-wants: *exam, results, material*

7. db_{TRA}

   database-is: *transplant*

   database-has: *organs-storage, surgery*

   database-wants: *organs-storage, surgery, history*

After defining the terms related to the different parts associated with each database system, the coordinator of the federation starts to add the participating components. The incremental addition of the database systems generates the following group names, with the respective database systems: *Accident* ($db_{A\&C}$, db_{HB}), *Casualty* ($db_{A\&C}$, db_{HB}), *Cardiology* ($db_{C\&P}$), *Child-Care* (db_{CHI}, db_{HB}), *Imaging* ($db_{R\&I}$), *Laboratory* (db_{LAB}), *Oncology* (db_{ONC}), *Ophthalmology* (db_{OPH}), *Orthopaedics* (db_{OTP}), *Physiotherapy* (db_{OTP}), *Pulmonology* ($db_{C\&P}$), *Radiology* ($db_{R\&I}$), *Transplant* (db_{TRA}, db_{HB}), and *Traumatology* (db_{OTP}). Figure 8 presents the hierarchical information structure generated by the tool, with the group names and their specialised terms.

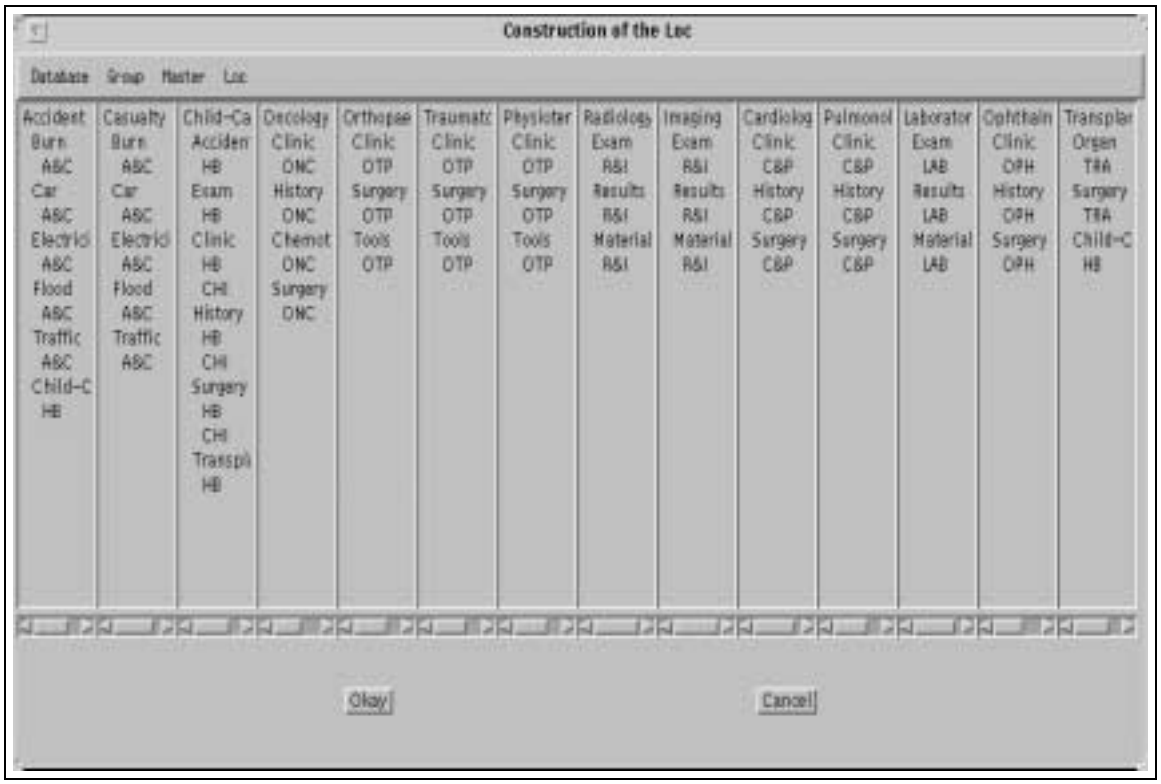

Figure 8: A version of the hierarchical information structure being constructed

In figure 9 another version of the structure is presented after the execution of 'terms cross referencing' and the removal of some unnecessary groups. Note the creation of the group *Surgery* with specialised terms such as *Child-Care, Oncology, Orthopaedics, Traumatology, Physiotherapy, Cardiology, Pulmonology, Ophthalmology,* and *Transplant.* The same occurs with *History* and *Exam* groups.

Based on this version, the coordinator of the federation performs some modifications. For instance, the *Imaging, Laboratory* and *Radiology* groups are removed, due to the existence of *Exam* group, where *Imaging, Laboratory* and *Radiology* became specialisations of this group. The final version of the hierarchical information structure is presented in figure 10. The coordinator decides to remove the specialised terms in the *Accident* and *Casualty* groups. However, groups like *Oncology, Orthopaedics, Traumatology, Physiotherapy, Cardiology, Pulmonology* and *Ophthalmology* maintain their specialised terms, even having the same set of database systems related to each of these terms. The purpose of maintaining the specialised terms in these groups is to allow a better notion of the information composing the associated databases. Therefore, these specialisations are not crucial and are optional. The hierarchical information structure contains some redundant paths such as *Surgery, Ophthalmology* and *Ophthalmology, Surgery*, giving the users more flexibility when composing the requests.

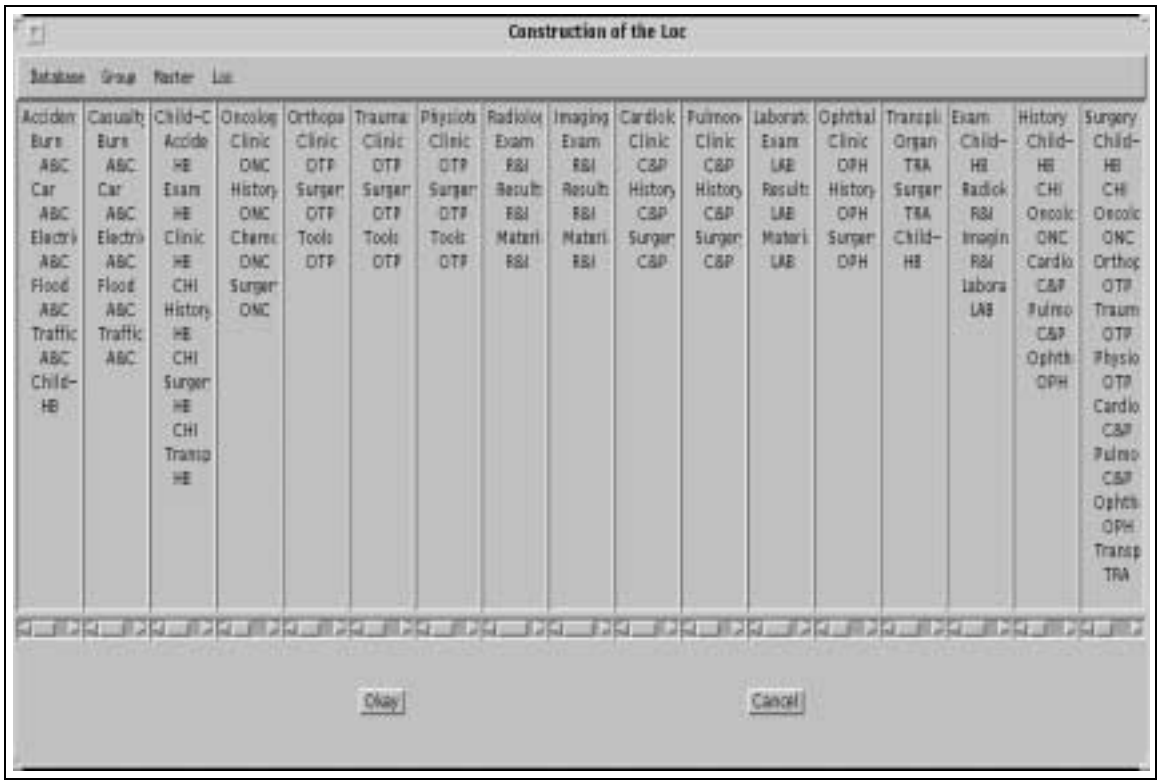

Figure 9: Another version of the hierarchical information structure being constructed

3 Evolving the Hierarchical Information Structure

The hierarchical information structure can evolve either as a reflection of system evolution, addition and removal of database systems, or as a need to modify the structure in order to adjust it to the requirements of the users and applications.

The addition of a new database system to an existing federation may cause the creation of new groups and the specification of specialised terms. Therefore, we use the process described in section 2, in which a database system is added in the same way as when constructing a hierarchical information structure. The DBA of the database system to be added defines the *database-is*, *database-has* and *database-wants* parts. The coordinator of the federation informs the support tool about this database system. The support tool tries to add the database system into existing groups and terms, or to create new groups and terms. The coordinator verifies the suggestions of the tool and performs necessary changes. After defining the configuration of the hierarchical information structure to be used in the federation, the hierarchical information structures of the other database systems participating in the related federation are updated.

In order to support evolution of the hierarchical information structure and to decide about its modifications, we suggest the use of two strategies. In the first strategy, we propose maintaining statistical information about the terms composing a hierarchical information structure. The idea consists of having a *counter* associated with each group name and term in the hierarchical information structures. A term has its counter in-

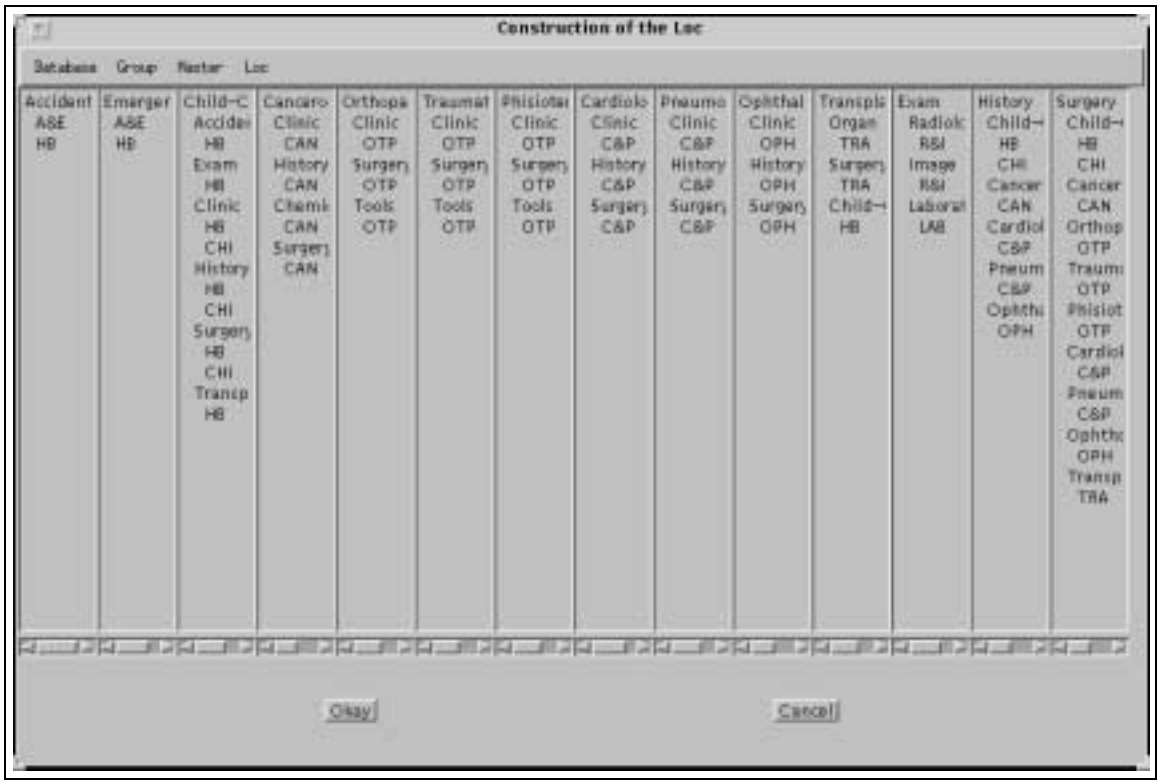

Figure 10: Final version of the hierarchical information structure being constructed

cremented whenever this term is used in a request and identified when traversing the hierarchy. The amount related to the use of a certain term in a federation is specified by the addition of the values of all the counters related to this term, in all the hierarchical information structures of the associated federation which were traversed during a discovery process. Periodically, the coordinator of the federation analyses the values of the various counters. The coordinator decides about maintaining, removing, joining, or modifying the terms in the hierarchical information structure. The decision is based on the relation between the values of the counters and the total number of query requests executed in the federation. Hence, some terms in the hierarchical information structure which are never or hardly ever used are identified and thereby removed or modified.

The second strategy is to allow the users to suggest new terms to be used in the hierarchical information structure. This is executed when a user is formulating a query request and verifies the absence of terms matching the request in the hierarchical information structure being browsed. The suggested terms can be related to any level of the structure and are based on the interest of the users. These terms are stored in a special structure named 'suggestion box'. Periodically, the coordinator of the federation consults this structure and performs modifications on the hierarchical information structure, depending on the suggestions.

4 Conclusion

The hierarchical information structure used in the system evolves as a consequence of system modifications and as a need to adjust the structure to the requirements of the users and applications. The definition of the terms composing the hierarchical information structures is important for the execution of the discovery process. It requires human assistance. Therefore, it is necessary to have a way to assist the coordinators of the federations when constructing and evolving a hierarchical information structure. In this technical report we have presented a methodology and a support tool to assist the coordinators with these tasks.

The idea is to build a hierarchical information structure based on the information being shared by each database system and on the requirements of the users and the applications. This information is described by using, as much as possible, standard terms. The support tool automatically builds a first version of the hierarchical information structure to be used in a federation, by incremental addition of the participating database systems. The coordinator of the related federation may perform any necessary changes in this first version, in order to construct the final initial version of the structure. On the other hand, modifications to a hierarchical information structure are performed based on new database systems added to the federation, statistical information related to the use of the terms composing the structure and interests of the users.

References

[1] J. Hammer and D. McLeod. An approach to resolving semantic heterogeneity in a federation of autonomous, heterogeneous database systems. *International Journal of Intelligent and Cooperative Information Systems*, 2(1):51–83, 1993.

[2] D. Heimbigner and D. McLeod. A federated architecture for information management. *ACM Transaction on Office Information Systems*, 3(3):253–278, July 1985.

[3] M. Stonebreaker, P.M. Aoki, A. Pfeffer, and A. Sah. Mariposa: A wide-area distributed database system. *VLDB Journal*, 5(1):48–63, January 1996.

[4] A. Bouguettaya, S. Milliner, and R. King. Resource location in large scale heterogeneous and autonomous databases. *Journal of Intelligent Information System*, 5(2), 1995.

[5] R. Giladi and P. Shoval. An architecture of an intelligent system for routing user requests in a network of heterogeneous databases. *Journal of Intelligent Information Systems*, 3:205–219, 1994.

[6] S. Milliner, A. Bouguettaya, and M. Papazoglou. A scalable architecture for autonomous heterogeneous database interactions. In *Proceedings of the 21st International Conference on Very Large Data Bases*, pages 515–526, Zurich, Switzerland, 1995.

[7] A. Zisman. Towards interoperability in heterogeneous database systems. Technical Report 11, Department of Computing, Imperial College of Science, Technology and Medicine, 1995.

[8] A. Zisman and J. Kramer. An architecture to support interoperability of autonomous database systems. In 2^{nd} *International Baltic Workshop on DB and IS*, pages 87–98, Estonia-Tallin, June 1996.

[9] A. Zisman and J. Kramer. An information discovery process for interoperable heterogeneous databases. In *Americas Conference on Information Systems*, Association for Information Systems, pages 617–619, Phoenix, Arizona, August 1996.

[10] K. Mahalingam and M.H. Huhns. An ontology tool for query formulation in an agent-based context. In *Second IFCIS International Conference on Cooperative Information Systems (CoopIS97)*, IEEE Computer Society, pages 170–178, Kiawah Island, South Carolina, June 1997.

[11] G. Wiederhold. Interoperation, mediation, and ontologies. In *Proceedings International Symposyum on Fifth Generation Computer Systems*, volume w3, pages 33–48, Tokyo, Japan, December 1994.