Int J Digit Libr (1997) 1: 122–131

INTERNATIONAL JOURNAL ON
Digital Libraries

Thesaurus federations: loosely integrated thesauri for document retrieval in networks based on Internet technologies

Ralf Kramer, Ralf Nikolai, Corinna Habeck

Forschungszentrum Informatik (FZI), Haid-und-Neu-Str. 10–14, D-76131 Karlsruhe, Germany, {kramer, nikolai, habeck}@fzi.de

Received: 15 October 1996 / Accepted: 14 January 1997

Abstract. As a result of the distribution of interrelated information over several different information systems, the interconnection of information systems has increased in recent years. However, a purely technical interconnection is insufficient for users who need to find their way to information they are looking for. Thesauri are a proven means to identify documents, e.g., books of interest in a library. For different domains, different thesauri are available, which can be used in information systems as well, e.g., for the indexing and retrieval of data objects. Thus, the interconnection of information systems raises the need to integrate related thesauri. Furthermore, recent advances in open interoperability technologies (World Wide Web, CORBA, and Java) offer the potential for completely new technical solutions for employing thesauri.

This paper presents an approach for integrating multiple thesaurus databases. It concentrates on the integration of *distributed and heterogeneous* thesaurus databases and the integration of *multilingual and monolingual* thesauri. The software architecture takes advantage of the most advanced Internet and CORBA technology currently available in public domain and in commercial implementations.

Key words: Thesaurus federations – Internet thesauri – Integrating thesaurus databases – Heterogeneous resource indexing – Distributed and multilingual thesauri

1 Introduction

1.1 Motivation and general approach

Information, even when it is interrelated, is likely to be distributed among different systems. Due to this fact, interconnection among such systems has increased in recent years. However, a purely technical interconnection is insufficient for users who need to find their way to information they are looking for.

Digital libraries are important examples of distributed multimedia information systems. Information stored in digital libraries includes text documents, images, audio and video sequences, and spatial and geographic information. Thesauri provide the possibility of indexing each of these heterogeneous information resources in a uniform way and therefore enable effective information retrieval.

Thesauri are a proven means to identify documents, e.g., books in a library. The purpose of thesauri in information retrieval is to provide a uniform and consistent vocabulary for indexing documents in information systems and to supply users with a certain vocabulary for the retrieval of documents in such systems. For different domains, different thesauri are available. Even within a single domain, different thesauri which are specialized in dealing with specific areas may be used.

Since each special field has a terminology of its own, distinct thesauri are useful for individual understanding in each of these fields. On the other hand, in order to exchange information among different fields, it is advantageous to use a common thesaurus, which is likely to be more general. Hence, in general, it makes sense to use multiple thesauri simultaneously. These thesauri can either be independent of each other or more or less closely interrelated, such as so-called macro/micro thesauri.

In the past, multiple thesauri have been integrated manually into a single "super-thesaurus". In contrast to this approach, the technical interoperability of distributed information systems rather suggests a loose integration, a federation, of thesauri that allows them to retain their autonomy. Furthermore, recent advances in open interoperability technologies such as the World Wide Web, the OMG's Common Object Request Broker Architecture (CORBA), and the programming language Java offer the potential for completely new technical solutions for employing thesauri.

Whereas digital libraries call – among other things – for new approaches to classification and cataloguing

(see, e.g., [5, p. 24]), the use of a thesaurus federation offers new ways of classification. Thesaurus providers keep their autonomy, e.g., they can be paid per thesaurus use. The indexing as well as the retrieval process of thesaurus terms benefit from having access to different, general and specialized thesauri which are part of the thesaurus federation. The most appropriate thesauri can be selected and used. Improvements related to one thesaurus or the content integration of several thesauri are immediately available to the users. Updates at the client site are not necessary.

Building upon the initial concept presented in [11], in this paper, we present an approach to integrating multiple thesaurus databases. We concentrate on the integration of distributed and heterogeneous thesaurus databases and the integration of multilingual and monolingual thesauri. Our software architecture takes advantage of the most recent Internet software such as the World Wide Web, CORBA, and Java.

1.2 Outline

This paper is organized as follows. Section 2 reviews basic concepts for thesauri, technical cornerstones, and software architectures that are important for our work and the remainder of this paper. Section 3 discusses related work. Section 4 presents the software architecture of our system. Section 5 focuses on how multiple thesauri can be used for monolingual and multilingual indexing and retrieval. Section 6 concludes the paper with the major findings and an outlook on future work.

2 Basics

2.1 Thesauri

2.1.1 General introduction

A *thesaurus* is a set of terms selected from a natural language representing the vocabulary of a certain field; it is used for the indexing, saving, and retrieval of data objects (e.g., books) [25].

Indexing is the process of assigning thesaurus terms to data objects. *Retrieval* is the process of locating data objects with the help of thesaurus terms. Thesaurus terms can be classified by *descriptors* and *non-descriptors*. Descriptors are terms used directly for the indexing and retrieval of objects. Non-descriptors are not used for indexing; they are supplementary entry points for the user, providing a wider range of terms for the retrieval of objects. Information about descriptors, e.g., an explanation of the meaning of a term, can be saved in scope notes.

The terms in a thesaurus are related to each other through diverse relationships [2].

1. The *equivalence relationship* relates synonyms or quasi-synonyms to descriptors. Quasi-synonyms are terms that do not have exactly the same meaning as the descriptor to which they are related.
2. The *hierarchical relationship* relates descriptors to their super- and subordinate terms. Superordinate terms can be generalizations or subsumptions of descriptors and subordinate terms can be specializations or parts of descriptors.
3. The *associative relationship* relates descriptors to each other, e.g., if they are narrower terms for the same broader term or if they are opposites.

2.1.2 Multilingual thesauri

A *multilingual thesaurus* is a regular thesaurus, with an equivalent term for every descriptor (possibly for non-descriptors as well) in each of the languages covered [3].

In multilingual thesauri each language can have the same status or one language can have a dominant status as the main language. In the first case, descriptors exist in every language. In the second case, every thesaurus term in a secondary language has to be related to a descriptor in the main language.

For multilingual thesauri there is one more equivalence relationship between the terms of different languages, the inter-language equivalence. To translate terms from one language to the indexing language, the inter-language relationship connects descriptors to each other. Non-descriptors are not connected, since the purpose of a multilingual thesaurus is not to translate, but to provide a controlled vocabulary for indexing and retrieval. To reach a descriptor in a certain language using a non-descriptor from another language, both intra- and inter-language equivalences are used. Intra-language equivalences include synonyms and related terms. Inter-language equivalences include the mappings of descriptors in different languages to each other.

The advantage of multilingual thesauri is that users do not have to be familiar with a particular language. They can use several languages for retrieval from an information system, even though terms of just one language were used for indexing.

2.2 Technical cornerstones

We assume that the reader is familiar with the basics of the World Wide Web (WWW). Here, we introduce CORBA, Java, the integration of the two, and Internet-based federation architectures.

2.2.1 Common Object Request Broker Architecture (CORBA)

The *Common Object Request Broker Architecture* (CORBA) is a standard for open distributed systems that is defined by the Object Management Group (OMG) [17]. It defines ways for objects and clients to interact within a distributed environment [9, 16, 19].

The main features of the CORBA specification are a core object model, localization transparency (clients and

server need not be aware of their respective locations, e.g., at different hosts), and programming language independence. These are realized by providing a specific *Interface Definition Language* (IDL) as well as a Dynamic Invocation Interface (DII) for objects. CORBA's *Internet Inter ORB Protocol* (IIOP) enables object request brokers from different vendors to communicate with each other.

The OMG also specifies CORBA *services* and CORBA *facilities*. CORBA services are essential services concerning security, transaction handling, naming, etc. These services should be provided by any CORBA-compliant system. CORBA facilities are services that are not mandatory. Desktop management and time operations are examples of such facilities.

2.2.2 Java

To implement WWW access to applications, different techniques can be used. Since we address the problem of accessing databases, the conventional approach would be to use the Common Gateway Interface (CGI). We opt for a more advanced solution based on Java that allows us to overcome the limitations of the stateless Hypertext Transfer Protocol (HTTP) which has to be used in conjunction with CGI.

Java [8] is an object-oriented programming language that was especially designed to enable the development of secure and architecture-neutral programs for heterogeneous networks. Java programs called applets, can be embedded in Hypertext Mark-up Language (HTML) pages and executed by Java-enabled browsers. WWW servers transfer the Java code via the HTTP to the browser, which in turn interprets and executes it. Databases can be accessed from these applets via any kind of protocol; therefore, the stateless HTTP can be bypassed. As a result, client site interactivity is enhanced dramatically and safe and efficient communication between WWW server and client is supported.

2.2.3 Integrating CORBA and Java

The combination of Java, with its strength at the client site, and CORBA, powerful at the server site, seems very promising. In order to combine the features of both technologies, several vendors currently integrate CORBA and Java. An example of this integration is the VisiBroker for Java [23], an object request broker that connects CORBA with the WWW. The VisiBroker for Java is based on the CORBA implementation VisiBroker for C++ [22] and written in Java. Currently, VisiBroker objects can be implemented in Java or C++. For a more detailed overview, see [24].

2.3 Internet-based federation architectures

In a federation architecture, autonomous components are loosely coupled to build an integrated system. Federation architectures extend the idea of distributed database systems [1, 18]. Federated database management systems (FDBMS) have been proposed to make information from different database management systems (DBMS) available under a common frame, in which each of the participating DBMSs still retains its autonomy. A global schema does not exist: each local database maintains its local schema independently. To allow information exchange between the local databases, they support common export and import schemas. Further advantages of federation architectures are flexibility and ease of extension.

Figure 1 presents an example of a federation architecture that combines the technical elements described in

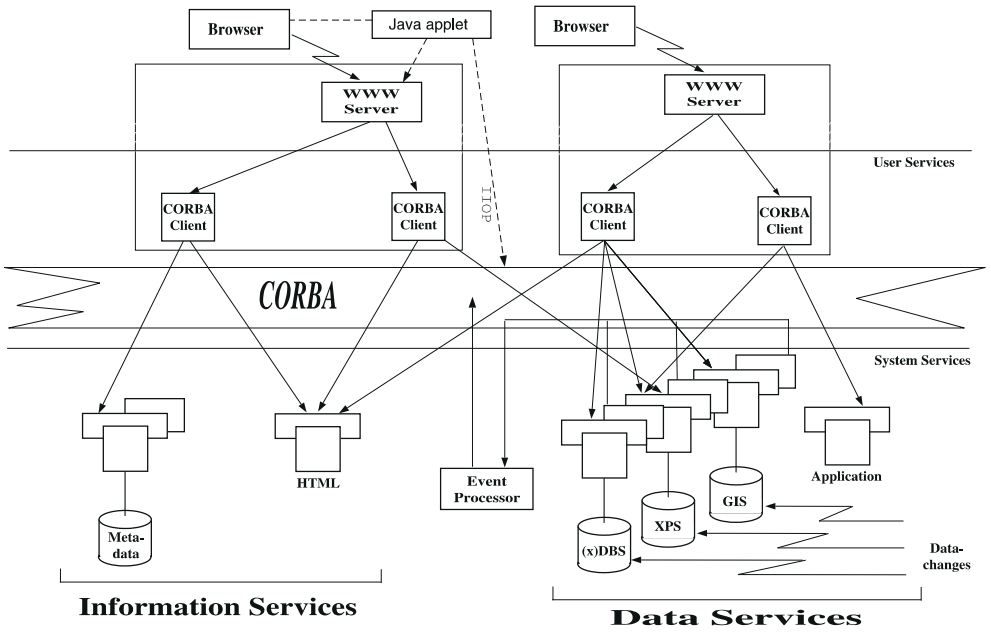

Fig. 1. A federation architecture incorporating the World Wide Web, CORBA, and Java

Sect. 2.2. As an extension to FDBMSs, not only databases ((x)DBS), but also geographic information systems (GIS) and expert systems (XPS) take part in the federation. CORBA is used as the integrating middleware layer.

Horizontally, we distinguish between user-level and system-level services. System-level services provide basic functionality, such as accessing databases and preparing HTML pages. User-level services basically combine several system-level services into higher-level services that are made available to the user. Java applets communicate with their server site based counterparts using the IIOP.

Vertically, we distinguish between information and data services. Information services are based on so-called metadata, i.e., data about data. These might incorporate one or more thesauri. They help the user to find relevant data sources, e.g., data services and reports. Data services access different kinds of data. Both information services and data services deliver results that are used as input for the generation of HTML pages.

3 Related work

We classify the work related to this paper into the areas of database integration and of thesaurus integration. Since the effort required to integrate databases via a global schema [18] is unacceptably high in a highly dynamic environment like the WWW, we concentrate on mediation as database integration technique.

3.1 Mediation techniques

Integration is the combination of multiple, possibly heterogeneous resources [27]. *Interoperation* is the ability of multiple resources and multiple applications to interact.

A model to provide interoperability of different database systems has to deal with conflicts arising from schematic and semantic heterogeneity [7]. *Schematic heterogeneity* includes naming conflicts and structural conflicts. *Semantic heterogeneity* deals with the actual data or data representation.

The mediation approach to providing interoperability between data sources and data receivers fits into the class of multidatabase language systems [1, 15]. Wiederhold [26] and Goh, Madnick, and Siegel [7] describe a way to deal with the conflicts of semantic and schematic heterogeneity by supplying needed information.

The following is some of the vocabulary used in the next sections. *Mediation* is the activity of interpreting data by applying knowledge of resources, search strategies, and user requirements. This means the knowledge of the schematic and semantic context of the data sources and the context of the required data in the application have to be available somewhere, to transform data to information. In this case, data is a set of letters and numbers, and information is the interpretation of data applying some kind of knowledge, e.g., the context knowledge of a date. This context information can be described in the form of concepts. *Concepts* are terms that define an abstract object or aggregation of objects, including its relationships [27]. A basis for formalizing context knowledge is an ontology. An *ontology* is a vocabulary for context definition. It is a set of terms and their relationships that are used in a domain, denoting concepts and objects [27]. The context definition can include the abstraction, the aggregation, and the format of data.

Wiederhold describes a mediator as an integrating concept [26]. Several technologies can be combined to find and transform data and to make information available on information highways. The author characterizes some of the methods and technologies a mediator should provide, i.e., knowledge of access paths, filtering of information, creation of views or objects of relational databases, etc. To describe the concepts and structure of the knowledge representation, diverse ontologies are used. Definitions of ontologies have to exist for every domain. The relationships between those concepts can be expressed by conversion functions. In addition to the mediation model, Wiederhold also defines a domain-knowledge-base algebra, providing a set of operations to handle ontologies.

The *Context Interchange Model* [7] provides data exchange between a data source and a data receiver by keeping the context specific to both in an export and an import context.

With mediation, the effort to add new data sources and receivers lies in the definition of concepts and relationships for that data source or receiver. When changes to data sources or receiver requirements occur, the ontology for that source or receiver needs to be changed. Mediation is the basic technique that we use in our approach. However, in addition to the development of thesaurus-specific ontologies, content-based integration of thesauri is necessary, which cannot be achieved by mediation.

3.2 Thesaurus integration

Approaches to thesaurus integration can be distinguished according to their technical integration and their content integration.

3.2.1 Technical integration

The technical integration of thesauri can be classified into autonomy-preserving and non-autonomy-preserving integration. The autonomy-preserving approach enables the user to access underlying thesauri through a global thesaurus. Local thesauri will not be modified. On the other hand, a non-autonomy-preserving integration builds a super-thesaurus using existing thesauri. The underlying thesauri do not keep their identity.

Stern and Rischette have developed a super-thesaurus from existing, multilingual thesauri in the field of agriculture, including AGROVOC and EUROVOC [21]. In this approach, the underlying thesauri (microthesauri) are integrated into a macrothesaurus. Object-oriented

software techniques and tools were used for the design of the super-thesaurus.

A non-autonomy-preserving construction of a super-thesaurus is described by Snidermann and Bicknell [20].

The main advantage of preserving the autonomy of underlying thesauri is that enhancement of the system is easier to realize.

3.2.2 Content-based integration

The content-based integration defines how terms from different thesauri are integrated. One possibility is that retrieved terms from one thesaurus are used as entry points to other thesauri. In this case, the relations between terms from different thesauri are generated through string matching. Another possibility is that new relationships between terms from different thesauri are defined. These relations may include equivalence relations, associative relations, and hierarchical relations.

The second possibility is realized in the approach of Snidermann and Bicknell [20]. This approach describes the partly dynamic generation of equivalence relations through string matching and ranking rules. Expert knowledge is needed to select the most similar term and to establish the hierarchical relationships of the integrated thesaurus.

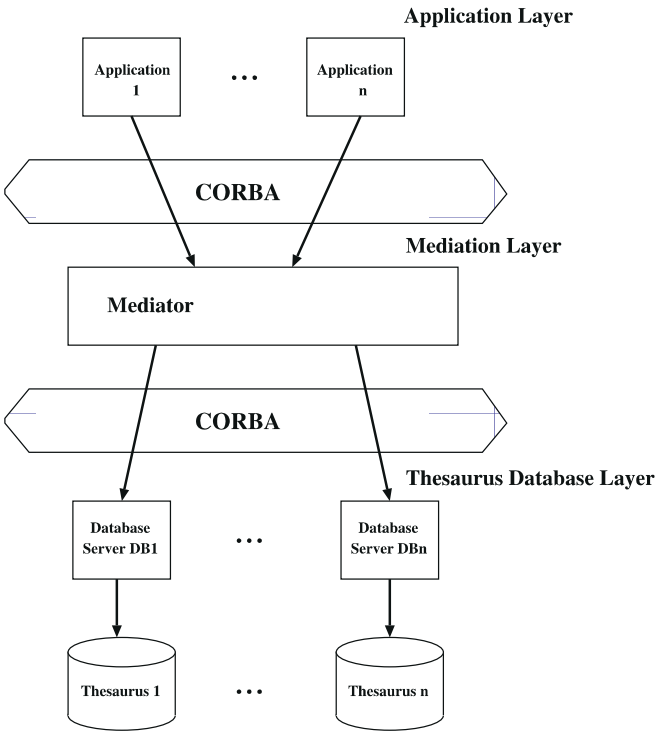

Fig. 2. System architecture

4 Software architecture

4.1 Overview

Figure 2 outlines the overall software architecture of our system. The architecture comprises three different layers:

1. The bottom layer of the system, the thesaurus database layer, consists of the thesaurus databases as well as the database servers, which provide access to the databases even when they are distributed.
2. The main functionality of the system is located in the mediation layer. The mediator receives requests from applications, translates the requests according to the databases that have to be accessed, and queries the affected databases.
3. The top layer is the application layer, which is the user interface to the system. Different applications can access the mediator and provide its functionality.

CORBA is the middleware that technically connects the three layers and, hence, allows all three to be distributed.

The following subsections describe the different layers. The mediation layer contains the main functionality of the system; therefore, this layer is described in greater detail.

4.2 Thesaurus database layer

Databases that represent the same data are most likely to be heterogeneous when generated by different providers. Our approach assumes that the thesaurus databases are modeled in the relational data model. Still, differences can occur, for example in the naming of attributes or in the database schema. Furthermore, the thesaurus databases can be located on different nodes in the network. Therefore, for each data source that provides data to the system, an interface that can be accessed remotely is needed. In the presented system, these interfaces are provided by CORBA objects. The services are called database wrappers. A wrapper is a module that extends or changes the functionality of a system. In our case, the database wrapper enables us to access a remote database.

4.3 Mediation layer

The mediation layer comprises several components, shown in Fig. 3. A repository stores the contexts of different thesauri and applications. This information is described using an ontology. A control component controls the sequence of steps and invokes other components. A translation module translates queries by applying the knowledge from the repository. A term mapper invokes queries to the term mapping database which stores inter-thesaurus relations.

When the control component needs to access thesaurus databases, it sends a query to the translation module, which translates the query according to the information stored in the repository. The control component then invokes remote services to send requests to the thesaurus databases and receives the results. The results

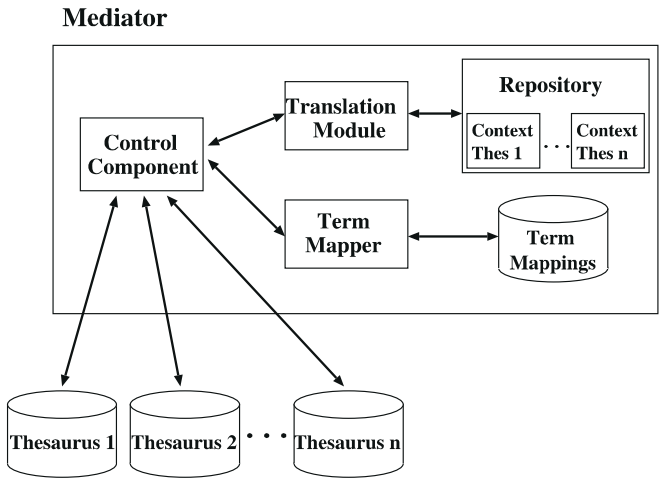

Fig. 3. Architecture of the mediation layer

are sent to the term mapper, which searches for inter-thesaurus relations between the non-indexing and the indexing thesauri. The control component then queries the indexing thesauri with the results from the term mapper.

The mediator needs to have an IDL interface, i.e., it needs to work as a CORBA server, because applications on remote network nodes might want to use it. It includes the client for the database wrapper.

4.3.1 Components of the mediator

The *repository* is a knowledge base for the mediator. It provides the information necessary to access different databases and to hand the data back to an application. This information includes the differences in the thesaurus databases and the differences in the data representations of applications. The context information in the repository is specified in terms of an ontology, which has been defined especially for this application domain. Through the ontology, the system can be easily enhanced, which is especially useful in a highly dynamic environment like the WWW. Further information about the thesaurus databases that is stored in the repository includes the name, the location, the primary key, and the thesaurus languages.

The *translation module* translates the queries of the control component into queries to the local thesaurus databases. If conflicts occur, the context information and functions supply the translation module with the necessary knowledge to transform data.

The *term mapper* realizes the content-based integration of different thesauri through the mapping of terms from one thesaurus to another. This is especially helpful when terms exist in one thesaurus but not in a second thesaurus. The correspondence between terms in different thesauri, which can be approximate, is stored in a database. The term mapper invokes queries to the term mapping database, in order to retrieve terms from the indexing thesauri. Therefore, term results of previous queries from non-indexing thesauri are passed to the term mapper by the control component.

The *control component* is the main component of the mediator. It formulates the queries and sends them to the translation module to transform them for the corresponding thesaurus databases. The attributes and relations in a query have to be formulated in the vocabulary of the ontology. After the translation module translates the query, the control component accesses the corresponding thesaurus databases and receives the results from the databases. Furthermore, the control component sends term lists from non-indexing thesauri to the term mapper, which queries the term mapping database. The results of this query are handed back to the control component and are used for retrieval in the indexing thesauri.

4.3.2 Ontologies

Ontologies are used to represent information about different thesaurus databases. For each thesaurus database, the names of the relations, attributes, and their relationships have to be known.

To integrate different thesaurus databases, a universal schema has been developed that represents general thesaurus information. The relations of the universal thesaurus schema have a maximum grade of normalization, i.e., the relations are decomposed free of loss into the smallest possible relations. None of the local thesaurus databases can be further decomposed. If local thesaurus databases are less decomposed than the universal thesaurus, i.e., information from two or more relations is combined in one single relation, then join operations that are normally required to retrieve information from relations of the local thesauri are not performed.

The ontology is applied by mapping the local schema to the universal schema similar to [26]. Matching attributes and relations are mapped to each other. For a less normalized local schema, the mapping to the universal schema is slightly more complicated.

If an attribute of the universal relation does not exist in a local schema, but can be substituted by another attribute, e.g., an identifier for synonyms is missing, then the substitute has to be specified instead, e.g., the synonym itself, except if an equivalent relation and an equivalent attribute are not available. The unavailability of an attribute or relation has to be specified by a hyphen.

4.4 Application layer

We decided to develop a WWW application, because it offers several advantages. WWW browsers are available on all major operating systems which makes WWW applications fairly easily accessible on different platforms. Furthermore, we have recently observed a convergence of open technologies that encourage

Internet-based federation architectures as outlined in Sect. 2.3. An example application is described in Sect. 5.4.

5 Indexing and retrieval with multiple thesauri

5.1 Indexing with multiple thesauri

Multiple thesauri for indexing provide the indexer with a more specialized and more precise vocabulary. Interconnection of the indexing thesauri used is desirable, to provide the indexer with the complete vocabulary he/she can use for indexing. The indexer needs knowledge of document contents in order to choose the right terms for indexing. Furthermore, he/she has to be familiar with the indexing language.

5.2 Retrieval with multiple thesauri

For the retrieval process, even more thesauri than the indexing thesauri used are advantageous, e.g., when a user descends from a different application domain and is not familiar with the terminology of the indexing thesauri. In this case, the mapping of terms from the non-indexing thesauri to the indexing thesauri is suggested. The retrieval of documents is possible even if the user is not familiar with the indexing language, since the translation of thesaurus terms is possible if multilingual thesauri are used for retrieval. The mediator needs to know the indexing thesauri of the information system.

To retrieve documents from an information system, only terms from the indexing thesauri can be used. The use of indexing and non-indexing thesauri is beneficial only if we are going to use terms from non-indexing thesauri as entrance points for indexing thesauri. These terms can be descriptors, as well as synonyms, since both terms may lead to descriptors in the indexing thesauri. The non-indexing terms can be mapped to indexing thesaurus terms or compared with indexing thesaurus terms via string matching.

In the following, thesauri are symbolized by T_i where i is a number from the interval $[1, n]$ and n is the number of thesauri the user has chosen. The indexing thesauri are symbolized by an I in front of the thesaurus symbol. The set of terms retrieved from the non-indexing thesauri is depicted by r_i, where i is the number of the thesaurus. R is the set of retrieved terms from the indexing thesauri.

The retrieval of terms from the indexing thesaurus occurs according to the following strategy. The user chooses a set of thesauri to be used for retrieval and enters a search term. All matching descriptors and synonyms are then retrieved from the selected thesauri. For these terms, all synonyms are selected as well. For each indexing thesaurus IT_i, the following steps have to be performed:

- In the term mappings we search for each descriptor from the set of terms retrieved from the non-indexing thesauri.
- For descriptors that do not have an entry in the mapping table, descriptors and synonyms are used as entry terms and compared via string matching with terms from this particular indexing thesaurus.

The set of descriptors retrieved from the indexing thesauri will then be presented to the user, who can choose a term and use it for retrieval in an information system.

Term mappings are useful when:

- the search for a term from one thesaurus in another thesaurus supplies an empty set as a result,
- the search for a T_i term supplies a term in IT_i, but the IT_i term is a homonym of the T_i term, i.e., it is written the same way but has a different meaning, or
- thesaurus T_i includes terms from application domain X, whereas the IT_i includes terms from application domain Y. In this case, vocabulary switching is desired.

Figure 4 shows the adapted representation for one indexing thesaurus. The term mappings of T_i terms to IT terms are represented by T_iIT.

5.3 Vocabulary switching

Hitherto, we did not consider the fact that multilingual and monolingual thesauri may be used and that the user may enter the search term in a certain language. For instance, the search term may be in English and some of the thesauri are monolingual in a different language. Several problems arise due to language differences between thesauri, and between the search term and the thesauri used. We use the terminology in Fig. 4 extended by an L, which stands for a unary language operator. For instance, $L(T_3)$ symbolizes the set of languages of the thesaurus T_3. For monolingual thesauri, this set includes just one element. The language of the search term $L(s)$ is a set with just one element.

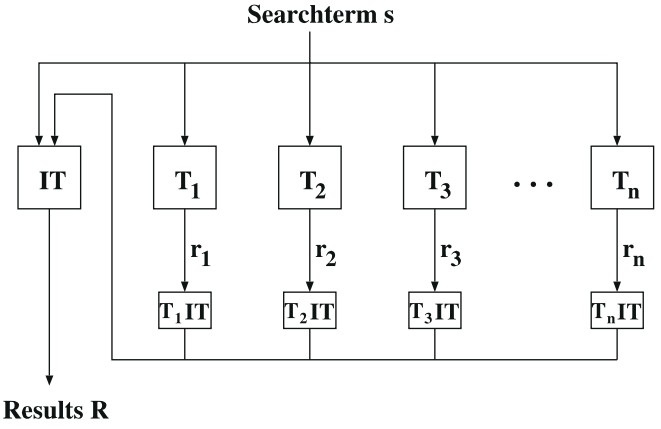

Fig. 4. Information retrieval with multiple thesauri using term mappings

1. $L(s) \notin L(T_i)$
   The language of the search term is different from the languages of a certain thesaurus. We have to translate the search term to $l \in L(T_i)$.
2. $L(IT_i) \cap L(T_i) = \emptyset$
   The indexing thesauri can be multi- or monolingual. In both cases, it may be that the language of a certain indexing thesaurus is not the same as that of the regular thesauri. In this case, the r_i have to be translated to $l \in L(IT_i)$.
3. $L(s) \notin L(IT_i)$
   To enable searching in a specific indexing thesaurus, it is necessary to translate the search term to $l \in L(IT_i)$. The user may have difficulties in understanding the results R from the indexing thesaurus, as they are not in the user language. Therefore, all terms from R could be translated into $L(s)$. As a result, many terms from R may not be translatable in the system and the user may not receive as many results as intended. Therefore, the results will be represented in $l \in L(IT_i)$.

The vocabulary switching is realized through the translation of terms using a multilingual local thesaurus. The multilingual thesaurus is identified through the thesaurus language information in the repository, which must include the desired source and target language.

5.4 Example application

Figure 5 shows the sequence of thesaurus pages generated by the application. Each page results in an invocation of the mediator. Currently, the system assumes that just one thesaurus has been used for indexing. However, our approach describes the generalization with different indexing thesauri.

The initial page lists the thesauri available in the system. To send a request, the user has to specify a search term. Since different thesaurus databases can be accessed, the user has to select a set of thesauri he/she wants to use. The resulting page lists the retrieved terms from the chosen thesauri. The user can choose a subset from these terms to retrieve terms from the indexing thesaurus. The chosen terms and their synonyms are then used as entry terms to the indexing thesaurus. The user may choose a term from this list and request the detailed representation of the chosen term. The detailed

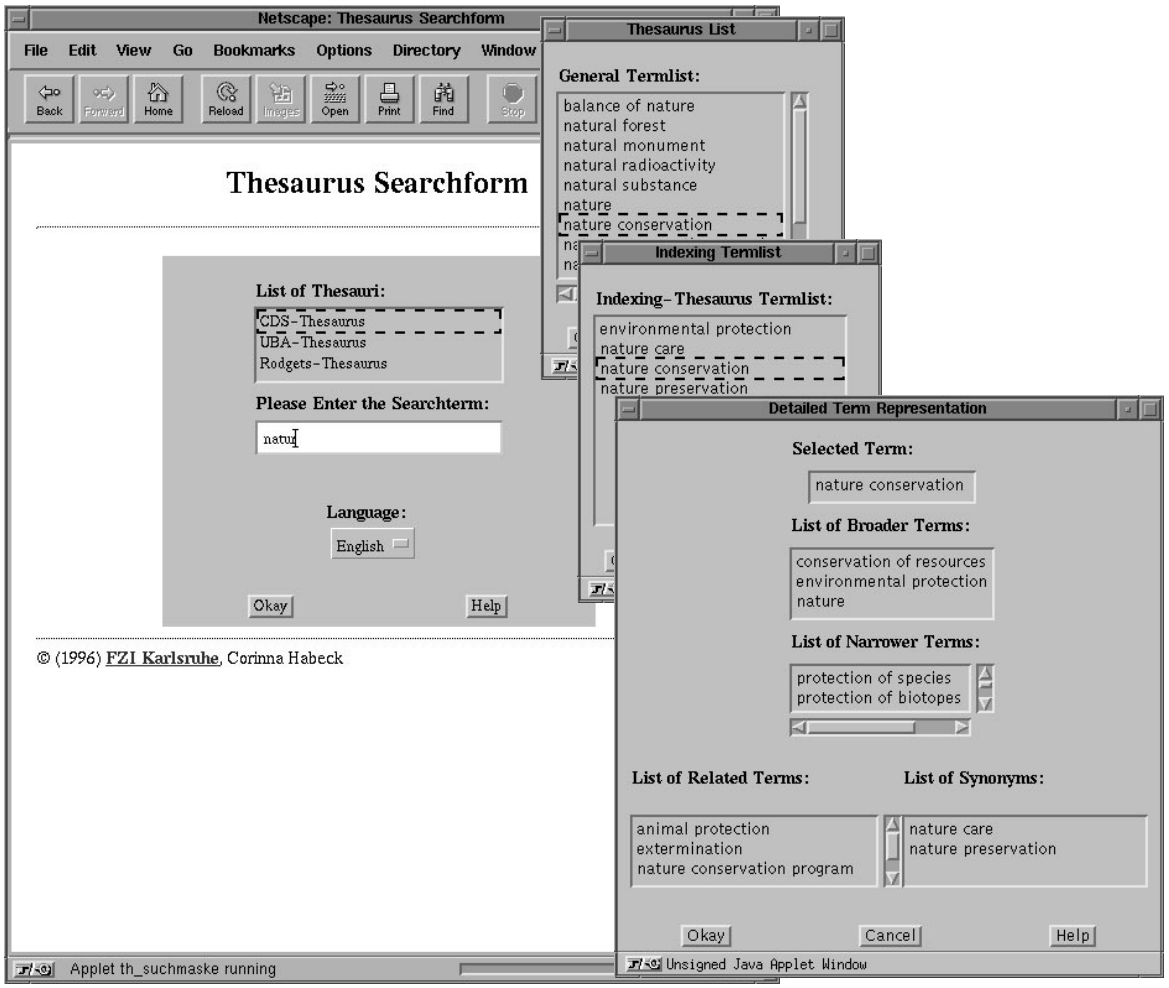

Fig. 5. Example of a sequence of thesaurus forms

representation of a term includes broader and narrower terms, as well as related terms and synonyms. Furthermore, the user is able to navigate through the broader and narrower terms in the hierarchy to find the desired term.

6 Conclusions and outlook

6.1 Conclusions

In this paper, we have presented a software architecture that integrates multiple thesauri with support for multilinguality. The use of the WWW technology as the front-end architecture for a multi-thesaurus system introduces new possibilities, i.e., worldwide access to information, but it also leads to a different dimension of difficulties that need to be solved. We have presented several solutions to these difficulties, which build a cornerstone for future work in this field.

The following items summarize the characteristics of the approach:

- The autonomy-conserving integration of multiple thesaurus databases is realized via mediation. In order to describe the schematic differences of the component thesauri, an ontology was designed. Using mediation, the extension of the multi-thesaurus system by adding another thesaurus is less costly than for global schema design, since the integration of schemas is not required.
- The support of the content-based integration of thesauri was achieved through the use of term mappings and the reuse of retrieved terms from non-indexing thesauri for querying indexing thesauri. This enables the user to switch vocabularies and supplies him/her with a broader range of resulting terms. Vocabulary switching provides the user with the ability to retrieve data objects with his/her terminology, even when the data objects are indexed in a different terminology.
- The approach allows the integration of multilingual and monolingual thesauri in the multi-thesaurus system, in order to provide a vocabulary for worldwide information systems. If possible, term results that are going to be reused as entry terms in different language thesauri are translated, in order to retrieve results from a diversity of thesauri.
- A fully operational prototype was implemented which uses one indexing thesaurus. It takes into account the distribution of databases. The distributed object architecture VisiBroker for Java, a CORBA implementation, was chosen as the underlying architecture, since this is the most innovative and promising technology of the approaches examined. Through the use of the WWW and Java, a platform-independent implementation was realized, which provides global access to the system.

The approach to thesaurus integration presented here offers several advantages when compared to previous approaches [21,20]. It takes into account the autonomy, distribution, and heterogeneity of underlying thesaurus databases. The autonomy preservation of thesaurus databases enables the user to choose certain vocabularies, rather than to use a broader range of terms in only one vocabulary. This is especially advantageous when vocabularies from different application domains have been integrated, as in some applications the user needs to know the origin, i.e., terminology of an application domain, in order to select the desired term.

We have taken advantage of the multilinguality of component thesauri to provide an approach to the development of a multilingual vocabulary interconnection for worldwide information systems. This is especially advantageous for applications available on the WWW.

6.2 Outlook

Currently, the translation of terms is achieved via the querying of multilingual thesauri. Since the purpose of multilingual thesauri is not to translate terms, the retrieved results in the translation steps will not be complete. In fact, the translation of terms from monolingual thesauri using a particular multilingual thesaurus might lead to an empty result set. The replacement of multilingual thesauri by dictionaries for translation purposes could improve the translation of terms in the system. Therefore the thesaurus federation has to be extended to a thesaurus and dictionary federation to cover a broader range of application areas.

Standardized multilingual thesauri only allow a one-to-one translation of terms, which does not always yield a satisfying translation. Different equivalence levels, like inexact (e.g., *teenager* and the German term *Jugendlicher*) or partial equivalences (e.g., *science – Wissenschaft*) lead to representation problems. Fuzzy thesauri extend conventional equivalence relations to enable the expression of approximate correspondences between terms. The user can additionally specify the degree of precision for term correspondence in which he/she is interested. Although the system presented allows the integration of fuzzy thesauri, several problems still need further investigations. One inherent problem of fuzzy thesauri is the difficulty in specifying the fuzzy values. Also, the user interface for fuzzy thesauri and – more generally – the use of fuzzy relations in a thesaurus federation has to be improved.

We plan to evaluate our prototype, which is fully operational, in a real-world scenario. One of the scenarios currently under consideration for this purpose comprises a set of environmental reports. These environmental reports are available as HTML pages both on the Internet and on an Intranet [6]. They can be retrieved using the environmental meta-information system WWW-UDK [10,12,13]. Currently, only a single fairly general environmental thesaurus which comprises some 8000 descriptors (preferred terms) is used for indexing these environmental reports. The indexers are already calling for more specific thesauri, such as one containing the terminology about the treatment of hazardous disused sites.

Another scenario is to index HTML pages by using a thesaurus federation and to store the descriptors as name value pairs directly in the HTML page. The use of a thesaurus federation that covers different domains and languages is especially advantageous in a huge international, multilingual information system like the WWW. Next-generation search engines could use this additional information to improve retrieval quality.

Acknowledgement. Prof. Dr. P.C. Lockemann's comments on an earlier version of this paper are gratefully acknowledged.

References

1. M.W. Bright, A.R. Hurson, S.H. Pakzad. A taxonomy and current issues in multidatabase systems. *IEEE Computer* 50–60, March 1992
2. DIN 1463/1987. Richtlinien für die Erstellung und Weiterentwicklung von Thesauri. Deutsche Industrienorm, 1987
3. DIN 1463/1993. Richtlinien für die Erstellung und Weiterentwicklung von mehrsprachigen Thesauri. Deutsche Industrienorm, 1993
4. Ahmed K. Elmagarmid Calton Pu. Special issue on heterogeneous databases. *ACM Computing Surveys* 22(3), September 1990
5. Edward A. Fox, Robert M. Akscyn, Richard K. Furuta, John J. Leggett. Digital libraries. *Communications of the ACM* 38(4):23–28, April 1995
6. W. Geiger, M. Reissfelder, R. Weidemann. Das WWW-basierte Altlasten-Fachinformationssystem AlfaWeb. In Lessing and Lipeck [14], pp. 211–220
7. Cheng Hian Goh, Stuart E. Madnick, Michael D. Siegel. Context interchange: overcoming the challenges of large-scale interoperable database systems in a dynamic environment. In *Proceedings of the Third International Conference on Information and Knowledge Management* (CIKM-94), Nov. 29–Dec. 2, 1994, Gaithersburg, MD, http://rombutan.mit.edu/pub/papers/cikm.ps, 1994
8. J. Gosling, B. Joy, G. Steele. *The Java Language Specification*. Addison-Wesley, 1996
9. Arne Koschel, Ralf Kramer, Dietmar Theobald, Günter v. Bültzingsloewen. Evaluation and application of CORBA implementations. In *ECOOP'96 Workshop*: *Putting Distributed Objects to Work*, *10th European Conference on Object-Oriented Programming*, Linz, Austria, July 1996
10. R. Kramer, T. Quellenberg. Global access to environmental information. In R. Denzer, D. Russel, G. Schimak, editors, *Environmental Software Systems; Proceedings of the International Symposium on Environmental Software Systems*, 1995, International Federation for Information Processing (IFIP), pp. 209–218, London, Chapman and Hall, 1996
11. Ralf Kramer, Ralf Nikolai. Accessing multilingual, heterogeneous data sources in wide area networks – requirements and approach. In Henning Christiansen, Henrik Legind Larsen, and Troels Andreasen, editors, *Flexible Query-Answering Systems*, *Proceedings of the* 1996 *workshop (FQAS*'96*)*, p. 255–264 Roskilde Universitetscenter, Roskilde, Denmark, May 1996
12. Ralf Kramer, Ralf Nikolai, Andree Keitel, Rudolf Legat, Konrad Zirm. Enhancing the environmental data catalogue UDK for the World Wide Web. In Lessing and Lipeck [14], pp. 59–68
13. Ralf Kramer, Ralf Nikolai, Arne Koschel, Claudia Rolker, Peter Lockemann, Andree Keitel, Rudolf Legat, Konrad Zirm. WWW-UDK: A Web-based environmental metainformation system. *ACM SIGMOD Record*, March 1997
14. H. Lessing, U.W. Lipeck, editors. *Informatik für den Umweltschutz*; 10. *Symposium*, *Hannover* 1996, number 10 in Umwelt-Informatik Aktuell, Marburg, Metropolis, 1996.
15. Witold Litwin, Leo Mark, Nick Roussopoulos. Interoperability of multiple autonomous databases. In *ACM Computing Surveys* [4], pp. 267–293
16. T. J. Mowbray, R. Zahavi. *The Essential CORBA*. John Wiley & Sons, New York, 1995
17. Object Management Group. The Common Object Request Broker: Architecture and Specification, Version 2.0. OMG Document, Object Management Group, Inc. (OMG), July 1995
18. Amit P. Sheth, James A. Larson. Federated database systems for managing distributed, heterogenous, and autonomous databases. In *ACM Computing Surveys* [4], pp. 183–236
19. J. Sigel. *CORBA Fundamentals and Programming*. John Wiley & Sons, New York, 1996
20. Charles A. Snidermann, Ellen J. Bicknell. Computer-assisted dynamic integration of multiple medical thesauruses. *Comput. Biol. Med.*, 22(1/2):135–145, 1992
21. A. Stern, N. Rischette. On the construction of a super thesaurus based on existing thesauri. In *Tools for Knowledge Organization and the Human Interface*, vol. 2, pp. 134–44. Indeks Verlag, Frankfurt/Main, Germany, 1990
22. Visigenic. VisiBroker for C++. http://www.visigenic.com/prod/vbcpd.html
23. Visigenic. VisiBroker for Java. http://www.visigenic.com/prod/vbjpd.html
24. A. Vogel. WWW and Java–threat or challenge for CORBA? *Middleware Spectra*, *Spectrum Reports*, *Winchester*, *UK*, May 1996. http://www.dstc.edu.au/AU/staff/andreas-vogel/papers/mws96/paper.html
25. Gernot Wersig. *Thesaurus-Leitfaden*. K.G.Saur Verlag, München, 1985
26. Gio Wiederhold. Interoperation, mediation and ontologies. In *Proceedings of the International Symposium on Fifth Generation Computer Systems (FGCS94), Workshop on Heterogenous Cooperative Knowledge-Bases* (ICOT), Tokyo, Japan, Dec. 1994, W3, pp. 33–48, http://www-db.stanford.edu/pub/gio/1994/medont.ps, 1994.
27. Gio Wiederhold. Glossary: Intelligent Integration of Information. Technical report, Intelligent Integration of Information (I3), http://www-db.stanford.edu/pub/gio/1994/vocabulary.html, 1995