Resource and Knowledge Discovery in Global Information Systems: A Scalable Multiple Layered Database Approach[*]

Jiawei Han[†] Osmar R. Zaïane Yongjian Fu
Database Systems Laboratory
School of Computing Science
Simon Fraser University
Burnaby, B.C., Canada V5A 1S6
{han, zaiane, yongjian}@cs.sfu.ca

Abstract

With huge amounts of information connected to the global information network (Internet), efficient and effective discovery of resource and knowledge from the "global information base" has become an imminent research issue, especially with the advent of the Information Super-Highway. In this article, a multiple layered database (MLDB) *approach is proposed to handle the resource and knowledge discovery in global information base. A multiple layered database is a database formed by generalization and transformation of the information, layer-by-layer, starting from the original information base (treated as layer-0, the primitive layer). Information retrieval, data mining, and data analysis techniques can be used to extract and transform information from a lower layer database to a higher one. Layer-1 and higher layers of an MLDB can be modeled by an extended-relational or object-oriented model, constructed automatically, and updated incrementally. Information at all the layers except the primitive one can be stored, managed and retrieved by the available database technology; resources can be found by controlled search through different layers of the database; and knowledge discovery can be performed efficiently in such a multiple layered database.*

1 Introduction

With the rapid expansion of information base and user community in the Internet, efficient and effective discovery and use of the resources in the global information network has become an important issue in the research into global information systems.

Although researches and developments of database systems have been flourishing for many years, a global information system stores much huger amount of information in a much more complicated and unstructured manner than any currently available database systems. Thus, the effective organization,

[*] This paper is a compressed version of a technical report (CMPT94-10) accessible via FTP at ftp://ftp.fas.sfu.ca/pub/cs/techreports/1994/CMPT94-10.ps.Z

[†] Research partially supported by the Natural Sciences and Engineering Research Council of Canada under the grant OGP0037230 and by the Networks of Centres of Excellence Program under the grant IRIS-HMI5.

discovery and use of the rich resources in the global information network poses great challenges to the database and information system researchers.

The first major challenge of a global information system is the **diversity of information** in the global information base. The current information network stores hundreds of tera-bytes of information including documents, softwares, images, sounds, commercial data, library catalog, user directory data, weather, geography, other scientific data, and many other types of information. Since users have the full freedom to link whatever information they believe useful to the global information network, the global information base is huge, heterogeneous, in multimedia form, mostly unstructured, dynamic, incomplete or even inconsistent, which creates tremendous difficulty in systematic management and retrieval than the structured, well-organized data in most commercial database systems.

The second challenge is the **diversity of user community**. The Internet currently connects over 2 million workstations [4], and the user community is still expanding rapidly. Users may have quite different backgrounds, interests, and purposes of usage. Also, most users may not have a good knowledge about the structure of the information system, may not be aware of the heavy cost of a particular search (e.g., a click may bring megabytes of data over half of the globe), and may easily get lost by groping in the "darkness" of the network, or be bored by taking many hops and waiting impatiently for a piece of information.

The third challenge is the **volume of information** to be searched and transmitted. The huge amount of unstructured data makes it unrealistic for any database systems to store and manage and for any queries to find *all* or even *most* of the answers by searching through the global network. The click-triggered massive data transmission over the network is not only costly and unbearable even for the broad bandwidth of the communication network, but also too wasteful or undesirable to many users. Search effectiveness (e.g., hit ratio) and performance (e.g., response time) will be bottlenecks for the successful applications of the global information system.

There have been many interesting studies on information indexing and searching in the global information base with many global information system servers developed, including Archie [11], Veronica, WAIS [18], etc. Although these tools provide indexing and document delivery services, they aim at a very specific service like FTP or gopher [24]. Attempts have also been made to discover resources in the World Wide Web [3, 27]. Spider-based indexing techniques, like the WWW Worm [23], RBSE database [9], Lycos [22] and others, create a substantial value to the web users but generate an increasing Internet backbone traffic. They not only flood the network and overload the servers but also lose the structure and the context of the documents gathered. These wandering software agents on the World Wide Web have already created controversies [21]. Other indexing solutions, like ALIWEB [20] or Harvest [5], behave well on the network but still struggle with the difficulty to isolate information with relevant context. Essence [17, 5], which uses a "semantic" indexing, is one of the most comprehensive indexing systems known up to now. However, it still cannot solve most of the problems posed for systematic discovery of resources and knowledge in the global information base.

In this article, a different approach, called a **Multiple Layered DataBase (MLDB)** approach is proposed to facilitate information discovery in global information systems. A **multiple layered database (MLDB)** is a database composed of several layers of information, with the lowest layer (i.e., *layer-0*) corresponding to the primitive information stored in the global information base and the higher ones (i.e., *layer-1* and above) storing summarized information extracted from the lower layers. Every layer i ($i \in [1..n]$) stores, in a conventional database, general information extracted from layer $i-1$. The extraction of general information from lower layers to higher layers (ie. generalization) will be explained in section 2.

The proposal of the multiple layered database architecture is based on the previous studies on *multiple layered databases* [26, 15] and *data mining* [25, 13] and the following observations.

With the development of data analysis, transformation and generalization techniques, it is possible to generalize and transform the diverse, primitive information in the network into reasonably structured, classified, descriptive and higher-level information. Such information can be stored into a massive, distributed but structured database which serves as the layer-1 database in the MLDB. By transforming an unstructured global information base into a relatively structured "global database", most of the database technologies developed before can be applied to manage and retrieve information at this layer.

However, the layer-1 database is usually still too large and too widely distributed for efficient browsing, retrieval, and information discovery. Further generalization should be performed on this layer at each node to form higher layer(s) which can be then merged with the corresponding layered database of other nodes at some backbone site in the network. The merged database can be replicated and propagated to other remote sites for further integration [6]. This integrated, higher-layer database may serve a diverse user community as a high-level, global information base for resource discovery, information browsing, statistical studies, etc.

The multiple layered database architecture transforms a huge, unstructured, global information base into progressively smaller, better structured, and less remote databases to which the well-developed database technology and the emerging data mining techniques may apply. By doing so, the power and advantages of current database systems can be naturally extended to global information systems, which may represent a promising direction.

The remaining of the paper is organized as follows. In Section 2, a model for global MLDB is introduced. Methods for construction and maintenance of different layers of the global MLDB are also proposed. Resource and knowledge discovery using the global MLDB is investigated in Section 3. A discussion of the benefits of the MLDB model and the related issues and a preliminary implementation are presented in Section 4. Finally, the study is summarized in Section 5.

2 A Multiple Layered Database Model for Global Information Systems

Although it is difficult to construct a data model for the primitive (i.e., layer-0) global information base, advanced data models can be applied in the construction of better structured, higher-layered databases. The construction of higher-layer models can be performed step-by-step, incrementally updatable, evolving from simple ones to sophsiticated, heterogeneous ones for advanced applications.

To facilitate our discussion, we assume that the nonprimitive layered database (i.e., layer-1 and above) is constructed based on an extended-relational model with capabilities to store and handle complex data types, including set- or list- valued data, structured data, hypertext, multimedia data, etc.

Definition 2.1 *A global* multiple layered database (MLDB) *consists of 3 major components:* ⟨S, H, D⟩, *defined as follows.*

1. S: a database schema*, which contains the meta-information about the layered database structures;*

2. H: a set of concept hierarchies*; and*

3. D: a set of (generalized) database relations at the nonprimitive layers of the MLDB and files in the primitive global information base. □

The first component, a database schema, outlines the overall database structure of the global MLDB. It stores general information such as structures, types, ranges, and data statistics about the relations at different layers, their relationships, and their associated attributes. Moreover, it describes which higher-layer relation is generalized from which lower-layer relation(s) (i.e., a route map) and how the generalization is performed (i.e., generalization paths). Therefore, it presents a route map for data and meta-date (i.e., schema) browsing and for assistance of resource discovery.

The second component, **a set of concept hierarchies**, provides a set of predefined concept hierarchies which assist the system to generalize lower layer information to high layer ones and map queries to appropriate concept layers for processing.

The third component consists of the whole **global information base** at the primitive information level (i.e., **layer-0**) and the **generalized database relations** at the nonprimitive layers.

Because of the diversity of information stored in the global information base, it is difficult to create relational database structures for the primitive layer information base. However, it is possible to create relational structures to store reasonably structured information generalized from primitive layer information. For example, based on the accessing patterns and accessing frequency of the global information base, layer-1 can be organized into dozens of database relations, such as *document, person, organization, software, map, library_catalog, commercial data, geographic_data, scientific_data, game*, etc. The relationships among these relations can also be constructed either explicitly by creating relationship relations as in an entity-relationship model, such as *person-organization*, or implicitly (and more desirably) by adding the linkages in the tuples of each (entity) relation during the formation of layer-1, such as *adding URL*[1] *pointers pointing to the corresponding authors ("persons") in the tuples of the relation "document" when possible.* Notice that an incremental updating of the schema, such as adding new attributes at layer-1, may imply incremental updating and propagating of the lower layer information to higher ones in the multiple-layered database, which may also require incremental updates of the layer building softwares.

A philosophy behind the construction of MLDB is **information abstraction**, which assumes that most users may not like to read the details of large pieces of information (such as complete documents) but may like to scan the general description of the information. Usually, the higher level of abstraction, the better structure the information may have. Thus, the sacrifice of the detailed level of information may lead to a better structured information base for manipulation and retrieval.

2.1 Construction of layer-1: From global information base to structured database

The goal for the construction of layer-1 database is to transform and/or generalize the unstructured data of the primitive layer at each site into relatively structured data, manageable and retrievable by the database technology. Three steps are necessary for the realization of this goal: (1) **standardization of the layer-1 schema**, (2) **development of a set of softwares which automatically perform the layer-1 construction**, and (3) **layer construction and database maintenance at each site.**

To simplify our discussion, we assume in this paper that the layer-1 database contains only two relations, **document** and **person**. Other relations can be constructed and generalized similarly.

Example 2.1 Let the database schema of layer-1 contain two relations, **document** and **person**, as follows (with the attribute type specification omitted).

1. **document**(*file_addr, authors, title, publication, publication_date, abstract, language, table_of_contents, category_description, key_words, index, URL_links, multimedia_attached, num_pages, form, first_page, size_doc, time_stamp, access_frequency, ...*).

2. **person**(*last_name, first_name, home_page_addr, position, picture_attached, phone, e-mail, office_address, education, research_interests, publications, size_of_home_page, time_stamp, access_frequency, ...*).

Take the *document* relation as an example. Each tuple in the relation is an abstraction of one *document* at layer-0 in the global information base. The first attribute, *file_addr*, registers its file name and

[1] Uniform Resource Locator. Reference is available by anonymous FTP from ftp.w3.org as /pub/www/doc/url-spec.txt

its "URL" network address. There are several attributes which register the information directly associated with the file, such as *size_doc* (size of the document file), *time_stamp* (the last updating time), etc. There are also attributes related to the formatting information. For example, the attribute *form* may indicate the format of a file: .ps, .dvi, .tex, .troff, .html, text, compressed, uuencoded, etc. One special attribute, *access_frequency*, registers how frequently the entry is being accessed. Other attributes register the major semantic information related to the document, such as *authors, title, publication, publication_date, abstract, language, table_of_contents, category_description, key_words, index, URL_links, multimedia_attached, num_pages, first_page*, etc. □

Since the layer-1 construction is a major effort, a set of softwares should be developed to automate the construction process. (Notice that some existing global information index construction softwares, like the Harvest Gatherer [5], have contributed to such practice and could be further developed to meet our needs by digging more semantic information out of documents). Unless certain information network construction standard is enforced, a software may not always be able to determine certain properties of a file, e.g., a file category (document vs. library catalog), which may need some minimal human assistance before and during the process.

The layer-1 construction softwares, after being developed and tested, should be released to the information system manager in a regional or local network, which acts as a "*local software robot*" for automated layer-1 construction. Customization may need to be performed on some softwares, such as handling multilingual information, etc., before they can be successfully applied to their local information bases to generate consistent layer-1 local databases. Softwares for upgrading database structures and information transformers should also be released timely to local information system managers to keep their local layer-1 database upgraded and consistent with others.

2.2 Generalization: Formation of higher layers in MLDB

Layer-1 is a detailed abstraction (or descriptor) of the layer-0 information. It should be substantially smaller than the primitive layer global information base but still rich enough to preserve most of the interesting pieces of general information for a diverse community of users to browse and query. Layer-1 is the lowest layer of information manageable by database systems. However, it is usually still too large and too widely distributed for efficient storage, management and search in the global network. Further compression and generalization can be performed to generate higher layered databases.

Example 2.2 Construction of an MLDB on top of the layer-1 global database.

The two layer-1 relations presented in Example 2.1 can be further generalized into layer-2 database which may contain two relations, doc_brief and person_brief, with the following schema,

1. doc_brief(*file_addr, authors, title, publication, publication_date, abstract, language, category_description, key_words, major_index, URL_links, num_pages, form, size_doc, access_frequency*).

2. person_brief (*last_name, first_name, publications, affiliation, e-mail, research_interests, size_home_page, access_frequency*).

The layer-2 relations are generated after studying the access frequency of the different fields in the layer-1 relations. The least popular fields are dropped while the remaining ones are inherited by the layer-2 relations. Long text data or structured-valued data fields are generalized by summarization techniques.

Further generalization can be performed on layer-2 relations in several directions. One possible direction is to partition the *doc_brief* file into different files according to different classification schemes, such as category description (e.g., *cs_document*), access frequency (e.g., *hot_list_document*), countries, publications, etc., or their combinations. Choice of partitions can be determined by studying the referencing statistics. Another direction is to further generalize some attributes in the relation and merge identical tuples to obtain a "summary" relation (e.g., *doc_summary*) with data distribution statistics associated [13]. The third direction is to join two or more relations. For example, *doc_author_brief* can be produced by generalization on the join of *document* and *person*. Moreover, different schemes can be combined to produce even higher layered databases.

A few layer-3 relations formed by the above approaches are presented below.

1. cs_doc(*file_addr, authors, title, publication, publication_date, abstract, language, category_description, key_words, URL_links, num_pages, form, size_doc*).

2. doc_summary(*affiliation, field, publication_year, count, first_author_list, file_addr_list*).

3. doc_author_brief(*file_addr, authors, affiliation, title, publication, pub_date, category_description, key_words, num_pages, URL_links, form, size_doc*).

4. person_summary (*affiliation, research_interest, year, num_publications, count*).

In general, the overall global MLDB structure should be constructed based on the study of frequent accessing patterns. It is also plausible to construct higher layered databases for a special interested community of users (e.g., ACM/SIGMOD, IEEE/CS) on top of a common layer of the global database.

This customized local higher layer acts as cache which may drastically reduce the overall network traffic [7, 2]. □

Clearly, successful generalization becomes a key to the construction of higher layered databases. Following our previous studies on attribute-oriented induction for knowledge discovery in relational databases [13, 14], an attribute-oriented generalization method has been proposed for the construction of multiple layered databases [15]. According to this method, data in a lower layer relation are generalized, attribute by attribute, into appropriate higher layer concepts. Different lower level concepts may be generalized into the same concepts at a higher level and be merged together, which reduces the size of the database.

We examine in detail the generalization techniques for the construction of higher layered databases.

2.2.1 Concept generalization

Nonnumeric data (such as keywords, index, etc.) is the most popularly encountered type of data in the global information base. Generalization on nonnumerical values should rely on the concept hierarchies which represent necessary background knowledge that directs generalization. Using a concept hierarchy, primitive data can be expressed in terms of generalized concepts in a higher layer.

Concept hierarchies should be provided explicitly by domain experts or stored implicitly in the database. For the global MLDB, a set of relatively stable and standard concept hierarchies should be provided as a common reference by all the local databases in their formation of higher layered databases and in their browsing and retrieval of information using different levels of concepts.

The concept hierarchies for keywords and indices can be obtained by referencing a standard concept hierarchy catalog which specifies the partial order of the terms frequently used in the global information base.

A portion of the concept hierarchy for *keywords* is illustrated in Fig. 1, and the specification of such a hierarchy and alias is in Fig. 2. Notice that a **contains**-list specifies a concept and its immediate

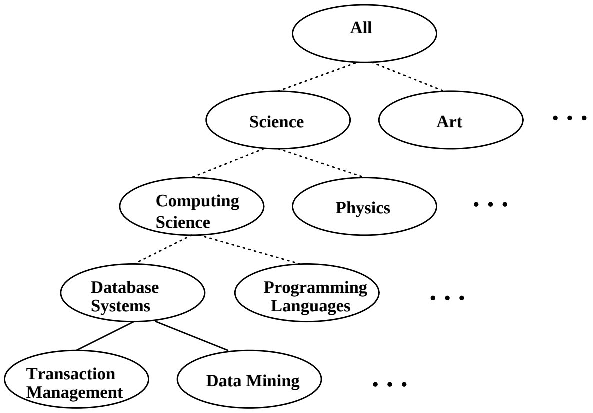

Figure 1: A possible concept hierarchy for *keywords*

All	**contains**:	Science, Art, . . .
Science	**contains**:	Computing Science, Physics, Mathematics, . . .
Computing Science	**contains**:	database systems, Programming Languages, . . .
Computing Science	**alias**:	Information Science, Computer Science, Computer technologies, . . .
Database Systems	**contains**:	Data mining, transaction management, query processing, . . .
Database Systems	**alias**:	Database technologies, Data management, . . .
Data mining	**alias**:	Knowledge discovery, data dredging, data archaeology, . . .
Transaction management	**contains**:	concurrency control, recovery, . . .
. . .		

Figure 2: Specification of hierarchies and aliases.

subconcepts; and an **alias**-list specifies a list of synonyms (aliases) of a concept, which avoids the use of complex lattices in the "hierarchy" specification. The introduction of alias-lists allows flexible queries and helps dealing with documents using different terminologies and languages. Also, the dashed lines between concepts in Fig. 1 represent the possibility to have other layers of concepts in between.

Generalization on numerical attributes can be performed in a more automatic way by the examination of data distribution characteristics [1, 12, 8]. In many cases, it may not require any predefined concept hierarchies. For example, the size of document can be clustered into several groups, such as {*below 10Kb, 10Kb-100Kb, 100Kb-1Mb, 1Mb-10Mb, over 10Mb*}, according to a relatively uniform data distribution criteria or using some statistical clustering analysis tools. Appropriate names can be assigned to the generalized numerical ranges, such as {*tiny-size, small-size, middle-size, large-size, huge-size*} by users or experts to convey more semantic meaning.

The availability of concept hierarchies allows us two kinds of generalization, *data generalization* and *relation generalization*. The data generalization aims to summarize tuples by eliminating unnecessary fields in higher layers which often involves merging generalized data within a set-valued data item. The summarization can also be done by compressing data like multimedia data, long text data, structured-valued data, etc. Whereas relation generalization aims to summarize relations by merging identical tuples in a relation and incrementing counts.

2.2.2 An MLDB construction algorithm

Based on the previous discussion, the construction of an MLDB can be summarized into the following algorithm, which is similar to attribute-oriented generalization in knowledge discovery in databases [13].

Algorithm 2.1 Construction of an MLDB.

Input: A global information base, a set of concept hierarchies, and a set of frequently referenced attributes and frequently used query patterns.

Output: A global multiple layered database (MLDB).

Method. A global MLDB is constructed in the following steps.

1. Determine the multiple layers of the database based on the frequently referenced attributes and frequently used query patterns.
2. Starting with the global information base (layer-0), generalize the relation step-by-step (using the given concept hierarchies and generalized schema) to form multiple layered relations (according to the layers determined in Step 1).
3. Merge identical tuples in each generalized relation and update the *count* of the generalized tuple.
4. Construct a new schema by recording the definitions of all the generalized relations, their relationships and the generalization paths. □

Rationale of Algorithm 2.1.

Step 1 indicates that the layers of an MLDB should be determined based on the frequently referenced attributes and frequently used query patterns. This is reasonable since to ensure the elegance and efficiency of an MLDB, only a small number of layers should be constructed, which should provide maximum benefits to the frequently accessed query patterns. Obviously, the frequently referenced attributes should be preserved in higher layers, and the frequently referenced concept levels should be considered as the candidate concept levels in the construction of higher layers. Steps 2 and 3 are performed in a way similar to the attribute-oriented induction, studied previously [13, 15]. Step 4 constructs a new schema which records a route map and the generalization paths for information browsing and knowledge discovery. □

Example 2.3 An portion of relation *doc_brief* is presented in Table 1.

file_addr	authors	title	publication	pub_date	key_words	···
http://fas.sfu.ca/9/cs/research /projects/HMI-5/documents /papers/han/coop94.ps.gz	J. Han Y. Fu R. Ng	Cooperative Query Answering Using Multiple Layered Databases	Proc. 2nd Int'l Conf. Cooperative Info. Systems	May 1994	data mining, multiple layered database, ···	···
···	···	···	···	···	···	···
ftp://ftp.cs.colorado.edu /pub/cs/techreports /schwartz/FTP.Caching-PS	P.B.Danzig R.S.Hall M.F.Schwartz	A Case for Caching File Objects Inside Internetworks	Proc. SIGCOMM	Sept. 1993	caching, ftp, ···	···
···	···	···	···	···	···	···
http://sobolev.mit.edu/people /jphill/publications/shap.dvi	J.R.Phillips H.S.J. Zant	Influence of induced magnetic fields on Shapiro steps in Josephson junction arrays	Physical Review B 47	1994	magnetic fields, Josephson array, Shapiro step, ···	···
···	···	···	···	···	···	···

Table 1: A portion of *doc_brief* extracted from *document* at layer-1.

By extraction of only the documents related to *computing science*, a layer-3 relation *cs_doc* can be easily obtained. Also, performing attributed-oriented induction on *doc_brief* leads to another layer-3 relation *doc_summary*, a portion of which is shown in Table 2.

affiliation	field	pub_year	count	first_author_list	file_addr_list	⋯
Simon Fraser Univ.	Database Systems	1994	15	Han, Kameda, Luk, ⋯	⋯	⋯
Univ. of Colorado	Global Network Systems	1993	10	Danzig, Hall, ⋯	⋯	⋯
MIT	Electromagnetic Field	1993	53	Bernstein, Phillips, ⋯	⋯	⋯
⋯	⋯	⋯	⋯	⋯	⋯	⋯

Table 2: A portion of *doc_summary* extracted from *doc_brief* at layer-2.

Notice that backward pointers can be stored in certain entries, such as *first_author_list* and *file_addr_list*, in the *doc_summary* table, and a click on a first author or a file_address will lead the presentation of the detailed corresponding entries stored in layer-2 or layer-1. □

2.3 Replication and distribution of the global MLDB

A global MLDB is constructed by extracting extra-layers from an existing (layer-0) global information base using generalization and transformation techniques. A higher layer database is usually much smaller than the lower layered ones. However, since the layer-1 database is resulted from direct, detailed information extraction from the huge global information base, its size is still huge. It is unrealistic to have this layer replicated and distributed to other servers.

A possible implementation is to store each local layer-1 database at each local network server site, but replicate the higher layered databases, such as layer-2 and above, and propagate them to remote backbone and/or ordinary network servers. Load can be further partitioned between backbone and ordinary servers. For example, one may store a complete layer-2 database at the backbone site but only the relatively frequently referenced portions of layer-2 and/or higher layers at the corresponding sites. Also, specifically projected layers (e.g., medical database) can be stored at the closely relevant sites (e.g., hospitals and medical schools).

By doing so, most information browsing and brief query answering can be handled by searching within the local network. Only detailed requests will be forwarded to the backbone servers or further to the remote sites which store the information. Only when the full document is explicitly requested by a user (with the awareness of its size), will the full layer-0 document be sent across the network to the user site. This will substantially reduce the amount of data to be transmitted across the network and improve the response time.

Moreover, some higher layered database could be defined by users for easy reference. For example, a user may define a new database at a high layer as "*all the documents related to heterogeneous databases published in major conferences or journals since 1990*". An information manager cannot construct a new database for every user's definition. Most such definitions will be treated like views, i.e., no physical databases will be created, and queries on such views will be answered by the query modification technique [10, 19]. Only if such a view is shared and frequently referenced, may it be worthwhile to create a new database for it.

Notice that the replication of higher-layered databases at network servers may take nontrivial disk space. However, the rapid progress of computer hardware technology has reduced the cost of disk space dramatically in the last decade. It could be more beneficial to trade disk space, as long as it is reasonable, for reduced network traffic and prompt query answering.

2.4 Incremental updating of the global MLDB

The global information base is dynamic, with information added, removed and updated constantly at different sites. It is very costly to reconstruct the whole MLDB database. Incremental updating could be the only reasonable approach to make the information updated and consistent in the global MLDB.

In response to the updates to the original information base, the corresponding layer-1 and higher layers should be updated incrementally. Incremental update can be performed on every update or at every night at the local site and propagate the updates to higher layers.

We only examine the incremental database update at insertion and update. Similar techniques can be easily extended to deletions. When a new file is connected to the network, a new tuple t is obtained by the layer-1 construction algorithm. The new tuple is inserted into a layer-1 relation R_1. Then t should be generalized to t' according to the route map and be inserted into its corresponding higher layer. Such an insertion will be propagated to higher layers accordingly. However, if the generalized tuple t' is equivalent to an existing tuple in this layer, it needs only to increment the count of the existing tuple, and further propagations to higher layers will be confined to count increment as well. When a tuple in a relation is updated, one can check whether the change may affect any of its high layers. If not, do nothing. Otherwise, the algorithm will be similar to the deletion of an old tuple followed by the insertion of a new one.

This simple incremental updating algorithm makes the global MLDB very scalable. No matter how many sites are generalized, when a new site joins, its layer-1 is constructed locally and propagated to the higher layers as described.

3 Resource and Knowledge Discovery in the Global MLDB

The model presented here is preliminary, serving as the first step towards a comprehensive multiple layered database model for resource and knowledge mining in global information systems. As the first step, it presents a simple, but clear and scalable way to organize the global information base, which makes the growing Internet more usable. The layer construction at the current conceptual level may need some human intervention with reasonable efforts. After refinement and tool development, the model will be constructed based on the agreement of most users, and fully automated processes are expected. The novelty of this framework is that it allows the discovery of both resources and implicit knowledge in the Internet, which will be examined in this section.

3.1 Resource discovery in the global MLDB

Most search engines currently available on the Internet are keyword-driven, and the answers presented are a list of URL anchors related to the keywords. The global MLDB allows us to apprehend and solve the resource discovery issues in two different ways: (1) presenting a list of pointers to documents corresponding to the request, (2) allowing the user to progressively and interactively browse detailed information leading to a targeted set of documents.

The resource discovery led by direct addressing uses the relations in a high layer, and possibly those in the lower layers as well, when necessary, to find a list of URL addresses of the documents or objects corresponding to the criteria specified in the query. By clicking at an entry in the list, the user has the choice to either first access the detailed descriptors of the document stored in the layer-1 or directly fetch the layer-0 document.

On the other hand, the resource discovery led by progressively detailed information browsing suits the users who do not have a clear mind on what are the exact resources that they need. The system first presents the top-layer high-level view with selected statistics to a relatively vague, or preliminary query, and works interactively with the user to progressively focus the search and deepen the layer. Such a search adopts a top-down approach, takes advantages of the available concept hierarchies and multiple information layers, and allows users to either interactively adding more constraints, such as "located in British Columbia", "published since 1994", etc. to refine the query, or to focus at a subset of high-level answers by selecting appropriate tuples in the answer list to go down to a lower layer for more detailed information. Finally, by clicking the entries in the last selected list, the detailed information can be down-loaded from layer-1 or the right documents can be fetched from the layer-0 information base.

3.2 Information browsing and knowledge discovery in the global MLDB

A major weakness of the current global information system services is their difficulty at supporting efficient and effective *information browsing* operations. The global MLDB is a meta-data based relational database and this architecture allows us to submit queries about the meta-data. In a global MLDB, the higher layer database stores the abstract or summary data and statistics of the global information base, and information browsing can be easily performed by searching through the higher layered databases.

Requesting and looking over meta-data (the higher layered database or a query-relevant portion of it) itself is one kind of information browsing. Notice that such information browsing may lead to resource discovery. However, another application of information browsing, which could be of its major purpose, is to visualize the information about the global information base and the artifacts it includes. This does not necessarily mean to find physical pointers on the Internet or the documents themselves but it may indicate to find interesting high level information about the global information base.

A glance at Table 2 shows how higher layers contain implicit data (i.e., counts) about the artifacts on-line. Note that these tables can also be expressed in rules. The global MLDB allows queries such as "*based on the document information on the Internet, list the universities in Europe which are productive in 1990s on database-related research*".

More detailed examples using NetQL query language, handling synonymy and concept closeness for knowledge discovery on the Internet, can be found in [16].

4 Discussion and Implementation

The multiple layered database architecture provides the following advantages for information discovery in global information systems.

1. **Application of database technology**: The MLDB architecture transforms an unstructured global information base into a structured, global database, which makes the database technology (not just storage management and indexing techniques) applicable to resource management, information retrieval, and knowledge discovery in the global information network.

2. **High-level, declarative interfaces and views**: The architecture provides a high-level, declarative query interface on which various kinds of graphics user-interfaces can be constructed for browsing, retrieval, and discovery of resource and knowledge. Moreover, multiple views can be defined by different users or user communities, cross-resource linkages can be constructed at different layers, and resource search can be initiated flexibly.

3. **Performance enhancement**: The layered architecture makes most searches confined to local or less remote sites on relatively small and structured databases, which will reduce the network bandwidth consumption, substantially enhance the search efficiency, and lead to relatively precise locating of resources and quick response of user's requests.

4. A global view of database contents: By preprocessing and generalizing primitive data, a global MLDB system may transform semantically heterogeneous, primitive level information into more homogeneous, high-level data at a high layer. It may provide a global view of the current contents in a database with summary statistics, which will assist users to browse database contents, pose progressively refined queries, and perform knowledge discovery in databases. Users could even be satisfied with the general or abstract data at a high layer and not bother to spend time and network bandwidth for more details.

5. Intelligent query answering and database browsing: A user may not always know the exact need when searching in the global information base. In the global MLDB system, a query is treated like an information probe, being mapped to a relatively high concept layer and answered in a hierarchical manner. This will provide with users a high-level view of the database, statistical information relevant to the answer set, and other associative and summary information at different layers.

6. Information resource management: Incremental updating can be performed on different layers using efficient algorithms, as discussed in Section 3. With the MLDB architecture, it is relatively easy to manage the global MLDB and make it consistent and up-to-date. For example, it is easy to locate weakly-consistent replicas [6] based on their property similarity at higher layers (rather than searching through the whole global information base!). Based on accessing statistics, one can also decide whether a duplicate should be removed or be preserved for resource redirection.

However, it is also important to note that cost should be paid for the construction of such a global MLDB, as presented below.

1. Extra disk spaces: Extra disk spaces are needed to store and replicate multiple layers and concept hierarchies. With the low cost of computer disks and hardwares, this seems not to be a bottleneck. Division of labor among different nodes and trade-offs between disk space and network bandwidth should also be considered in the construction of local or backbone MLDBs.

2. DBMS softwares: A subset of the functionalities of a database system, including storage management, indexing, query processing and recovery, should be considered as essential for construction and maintenance of MLDB and query processing in the global MLDB. Moreover, preliminary data mining facilities are also essential for the success of resource and knowledge discovery in the global MLDB. With the fast expansion of relational DBMS market, such preliminary database facilities (which are suggested to be specially designed, developed and distributed for the development of global information systems) should be affordable financially in the near future for most backbone and distributed servers.

3. New softwares for layer construction and query processing: Softwares should be developed for the construction and maintenance of the global MLDB, especially the extraction of different kinds of information from the global information base, and the implementation of query processing with additional relational operations introduced here. Some existing global information indexing and servicing softwares can be improved and adapted to the construction and use of MLDBs.

4. Reasonable standardization: Similar to the library catalog standardization, a classification standard for the documents in the global information base may need to be introduced and enforced to help reduce errors and enhance the quality of service in the development of the global MLDB.

A preliminary implementation is currently in process to prove the feasibility of our model. Our experiment is confined to computer science documents only. It is the best represented subset of the Internet since computer scientists are those who put most papers and technical reports on-line today. The results will be extrapolated to the whole Internet. Using a subset of the Internet also simplifies the concept hierarchies to be built. We are using Marc Vanheyningen's Unified computer science master index [28] as primitive data to create our MLDB. The master index was created by merging indexes of different FTP sites. These indexes, though not fully satisfactory to our usage, contain rich semantic information like keywords, abstract, etc. The layer-1 is built into a unique Sybase relation and the upper layers are constructed using a semi-automatically generated concept hierarchies. DBLearn, a data mining tool developed in our laboratory, will be used to query the MLDB and discover knowledge and resources. A Mosaic-like interface will also be put on-line for querying the MLDB and the algorithm described in section 2 will be implemented to update the data in different layers.

5 Conclusions

Different from the existing global information system services, a new approach, called *multiple layered database (MLDB) approach*, has been proposed and investigated for resource and knowledge discovery in global information systems. The approach is to construct progressively a global multiple layered database by generalization and transformation of lower layered data, store and manage multiple layered information by database technology, and perform resource and knowledge discovery by query transformation, query processing and data mining techniques.

The major strength of the MLDB approach is its promotion of a tight integration of database and data mining technologies with resource and knowledge discovery in global information systems. With the dynamically growing, highly unstructured, globally distributed and huge information base, the application of the mature database technology and promising data mining techniques could be an important direction to enhance the power and performance of global information systems.

Our study shows that the global MLDB can be constructed automatically and updated incrementally by integration of information retrieval, data analysis and data mining techniques, information at all of the nonprimitive layers can be managed by database technology, and resource and knowledge discovery can be performed efficiently and effectively in such a multiple layered database.

Our study presents a general framework of the MLDB approach for resource and knowledge discovery in the global information system. More studies are needed in the construction and utilization of the global multiple layered databases. We are currently developing softwares and performing experiments for automatic construction of the global MLDB on top of the global information base and for discovery of resource and knowledge in such a MLDB. Modifications and refinements to our initial design are expected, along with the progress of the research and developments. The effectiveness of the approach will be tested in the environment of the global information network, and further investigation and experimentation will be reported in the future.

References

[1] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer, and A. Swami. An interval classifier for database mining applications. In *Proc. 18th Int. Conf. Very Large Data Bases*, pages 560–573, Vancouver, Canada, August 1992.

[2] R. Alonso, D. Barbara, and H. Garcia-Molina. Data caching issues in an information retrieval system. In *ACM Transactions on Database Systems*, pages 359–384, 1990.

[3] T. Berners-Lee, R. Cailian, A. Luotonen, H. F. Nielsen, and A. Secret. The world-wide web. *Communications of the ACM*, 37:76–82, August 1994.

[4] C. M. Bowman, P. B. Danzig, U. Manber, and M. Schwartz. Scalable internet resource discovery: Research problems and approaches. *Communication of the ACM*, 37:98–114, August 1994.

[5] M. Bowman, P. Danzig, D. Hardy, U. Manber, and M. Schwartz. *Harvest, A scalable, Customizable Discovery and Access System.* Technical Report CU-CS-732-94 Department of CS, University of Colorado, Boulder, July 1994. Available from *ftp://ftp.cs.colorado.edu/pub/cs/techreports/schwartz/Harvest.FullTR.ps.Z.*

[6] P. Danzig, K. Obraczka, D. DeLucia, and N. Alam. *Massively replicating services in autonomously managed wide-area internetworks.* Technical report, 1994. Available from *ftp://catarina.usc.edu/pub/kobraczk/ToN.ps.Z.*

[7] P. B. Danzig, R. S. Hall, and M. F. Schwartz. A case for caching file objects inside internetworks. In *Proc. SIGCOMM'93*, pages 239–248, September 1993.

[8] B. de Ville. Applying statistical knowledge to database analysis and knowledge base construction. In *Proc. 6th Conf. on Artificial Intelligence Applications*, pages 30–36, Santa Barbara, CA, 1990.

[9] D. Eichmann. The RBSE spider - balancing effective search against web load. In *Proc. 1st Int. Conf. on the World Wide Web*, pages 113–120, May 1994.

[10] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems, 2nd ed.* Bemjamin/Cummings, 1994.

[11] A. Emtage and P. Deutsh. Archie: An electronic directory service for the internet. *Proc. of the USENIX Winter Conf.*, pages 93–110, 1992.

[12] D. Fisher. Improving inference through conceptual clustering. In *Proc. 1987 AAAI Conf.*, pages 461–465, Seattle, Washington, July 1987.

[13] J. Han, Y. Cai, and N. Cercone. Data-driven discovery of quantitative rules in relational databases. *IEEE Trans. Knowledge and Data Engineering*, 5:29–40, 1993.

[14] J. Han, Y. Fu, Y. Huang, Y. Cai, and N. Cercone. DBLearn: A system prototype for knowledge discovery in relational databases. In *Proc. 1994 ACM-SIGMOD Conf. Management of Data*, page 516, Minneapolis, MN, May 1994.

[15] J. Han, Y. Fu, and R. Ng. Cooperative query answering using multiple-layered databases. In *Proc. 2nd Int. Conf. Cooperative Information Systems*, pages 47–58, Toronto, Canada, May 1994.

[16] J. Han, O. R. Zaïane, and Y. Fu. *Resource and Knowledge Discovery in Global Information Systems: A Multiple Layered Database Approache.* Technical Report CMPT94-10, November 1994. Available from *ftp://ftp.fas.sfu.ca/pub/cs/techreports/1994/CMPT94-10.ps.Z.*

[17] D. Hardy and M. F. Schwartz. Essence: A resource discovery system based on semantic file indexing. In *Proc. of the USENIX Winter Conf.*, pages 361–374, 1993.

[18] B. Kahle. An information system for corporate users: Wide area information servers. In *Thinking Machines technical report TMC-199*, April 1991.

[19] H. F. Korth and A. Silberschatz. *Database System Concepts, 2ed.* McGraw-Hill, 1991.

[20] M. Koster. ALIWEB – archie-like indexing in the web. In *Proc. 1st Int. Conf. on the World Wide Web*, pages 91–100, May 1994.

[21] M. Koster. *A Standard for Robot Exclusion.* Nexor Corp., 1994. Available from *http://web.nexor.co.uk/mak/doc/robots/norobots.html.*

[22] M. L. Mauldin. *Lycos: Hunting WWW Information.* CMU, 1994. Available from *http://lycos.cs.cmu.edu/.*

[23] O. McBryan. GENVL and WWWW: Tools for taming the web. In *Proc. 1st Int. Conf. on the World Wide Web*, pages 79–90, May 1994.

[24] M. McCahill. The internet gopher protocol: A distributed server information system. *ConneXions- The Interoperability Report*, 6:10–14, July 1992.

[25] G. Piatetsky-Shapiro and W. J. Frawley. *Knowledge Discovery in Databases.* AAAI/MIT Press, 1991.

[26] R.L. Read, D.S. Fussell, and A. Silberschatz. A multi-resolution relational data model. In *Proc. 18th Int. Conf. Very Large Data Bases*, pages 139–150, Vancouver, Canada, Aug. 1992.

[27] M. F. Schwartz, A. Emtage, B. Kahle, and B. C. Neuman. A comparison of internet resource discovery approaches. *Comput. Syst.*, 5:461–493, Fall 1992.

[28] M. VanHeyningen. The unified computer sciencem technical report index: Lessons in indexing diverse resources. In *Proc. 2cd Int. Conf. on the World Wide Web*, October 1994. Available from *http://www.cs.indiana.edu/ucstri/paper/paper.html.*