Resource and Knowledge Discovery in Global Information Systems: A Preliminary Design and Experiment

Osmar R. Zaïane and Jiawei Han*

School of Computing Science
Simon Fraser University
Burnaby, B.C., Canada V5A 1S6
{zaiane, han}@cs.sfu.ca

Abstract

Efficient and effective discovery of resource and knowledge from the Internet has become an imminent research issue, especially with the advent of the Information Super-Highway. A multiple layered database (MLDB) *approach is proposed to handle the resource and knowledge discovery in global information base. A preliminary experiment shows the advantages of such an approach. 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. Resources can be found by controlled search through different layers of the database, and knowledge discovery can be performed efficiently in such a layered database.*

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.

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, Veronica, WAIS, etc. Although these tools provide indexing and document delivery services, they aim at a very specific service like FTP or gopher. Attempts have also been made to discover resources in the World Wide Web (Schwartz *et al.* 1992). Spider-based indexing techniques, like the WWW Worm (McBryan 1994), RBSE database (Eichmann 1994), Lycos 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. Other indexing solutions, like ALIWEB (Koster 1994) or Harvest (Bowman *et al.* 1994), behave well on the network but still struggle with the difficulty to isolate information with relevant context. Essence (Bowman *et al.* 1994), which uses a "semantic" indexing, is one of the most comprehensive indexing systems currently known. 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. An MLDB is a database composed of several layers of information, with the lowest layer corresponding to the primitive information stored in the global information base and the higher ones 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 $i-1$. This extraction of information is called *generalization.*

The proposal of the multiple layered database architecture is based on the previous studies on *multiple layered databases* (Han, Fu, & Ng 1994) and *data mining* (Piatetsky-Shapiro & Frawley 1991; Han, Cai, & Cercone 1993) and the following observation: 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 remainder of the paper is organized as follows: in Section 2, a model for global MLDB is introduced, and methods for construction and maintenance of the layers of the global MLDB are also proposed; resource and knowledge discovery using the global MLDB is investigated in Section 3; a preliminary experiment is presented in Section 4; finally, the study is summarized in Section 5.

*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 of Canada under the grant IRIS-HMI5.

Generalization: Formation of higher layers

Layer-1 is a detailed abstraction 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 Construction of an MLDB on top of the layer-1 global database.

The two layer-1 relations presented in Example 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, category_ description, language, keywords, 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 (Han, Cai, & Cercone 1993). 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. □

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,an attribute-oriented generalization method has been proposed for the construction of multiple layered databases (Han, Fu, & Ng 1994). 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, reducing the size of the database.

Generalization on nonnumerical values should rely on the concept hierarchies which represent background knowledge that directs generalization. Using a concept hierarchy, primitive data can be expressed in terms of generalized concepts in a higher layer.

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

Generalization on numerical attributes can be performed automatically by inspecting data distribution. In many cases, it may not require any predefined concept hierarchies. For example, the size of document can be clustered into several groups according to a relatively uniform data distribution criteria or using some statistical clustering analysis tools.

Concept hierarchies allow 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. Relation generalization aims to summarize relations by merging identical tuples in a relation and incrementing counts.

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.

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

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, constructed and updated incrementally, evolving from simple ones to sophisticated, 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 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) and how the generalization is performed.

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 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 pointers pointing to the corresponding authors ("persons") in the tuples of the relation "document" when possible.*

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 the 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.

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) creation 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.

Example 1 Let the database schema of layer-1 contain two relations, document and person, as follows.

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 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. □

Computing Science	contains:	Theory, Database Systems, Programming Languages, ...
Theory	contains:	Parallel Computing, Complexity, Computational Geometry, ...
Parallel Computing	contains:	Processors Organization, Interconnection Networks, PRAM, ...
Interconnection Networks	contains:	Gossiping, Broadcasting, ...
Gossiping	alias:	Gossip Problem, Telephone Problem, Rumor, ...
Computational Geometry	contains:	Geometry Searching, Convex Hull, Geometry of Rectangles, Visibility, ...
...		

Figure 1: Specification of hierarchies and aliases extracted from the experimental concept hierarchy.

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 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.

Resource and Knowledge Discovery

As the first step towards a comprehensive multiple layered database model for resource and knowledge mining in global information systems, our model 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. The novelty of this framework is that it allows the discovery of both resources and implicit knowledge in the Internet.

Resource discovery in the global MLDB

Most search engines available on the Internet are keyword-driven, and their answers are a list of URL anchors. WebMiner, our MLDB system, apprehends and solves the resource discovery issues by (1) presenting a list of pointers to documents, and (2) allowing the user to 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 to find a list of addresses of objects corresponding to the criteria specified in the query. By clicking at an entry in the list, the user either accesses the detailed descriptors of the document stored in layer-1 or directly fetches the layer-0 document.

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 vague, preliminary query, and works interactively with the user to focus the search and deepen the layer. Such a search takes an advantage of the concept hierarchies and information layers. The approach allows users to either interactively add more constraints, such as "located in British Columbia", "published since 1994", etc. 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, either the detailed information or the documents can be downloaded.

Knowledge discovery in the global MLDB

A major weakness of the current global information system services is their difficulty in supporting effective *information browsing* operations. The global MLDB's architecture allows us to submit queries about the meta-data. In a global MLDB, a high layered database stores the summary data and statistics of the global information base; information browsing can be easily performed by searching through this high layer.

Requesting and looking over meta-data itself is one kind of information browsing which may lead to resource discovery. Perhaps, the major purpose of information browsing, however, is to visualize the information about the global information base and the artifacts it includes. This does not necessarily mean finding physical pointers on the Internet, but it may indicate finding high level implicit information about the global information base, which is, in other words, mining the Internet.

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 as rules.

WebQL(Han, Zaïane, & Fu 1994) has been defined for resource and knowledge discovery using a syntax similar to the relational language SQL. Four new operators, coverage, covered_by, synonym and approximation, have their correspondent language primitives in WebQL, respectively covers, covered_by, like and close_to. These operators allow us to take full advantage of the concept hierarchies for key-oriented searches in the MLDB. A search key can be at a more general concept level than those at the current level or be a synonym of the key used in the relation, and still be used effectively in a query.

The top-level WebQL query syntax is presented in Table 1. At the position for the keyword select in SQL, an alternative keyword list can be used when the search is to browse the summaries at a high layer; describe can be used when the search is to discover and de-

```
{ select | list | describe } { attributes_name_list | * }
from relation_list
[ related-to name_list ]
[ in location_list ]
where where_clause
```

Table 1: The top level syntax of WebQL.

scribe the general characteristics of the data; whereas select remains to be a keyword, indicating to find more detailed information. Two optional phrases, "related-to *name_list*" and "in *location_list*", are introduced in WebQL for quickly locating the related subject fields and/or geographical regions. They are semantically equivalent to some phrases in the where-clause, but their inclusion not only makes the query more readable but also helps system locate the corresponding higher layer relation if there exists one. The where-clause is similar to that in SQL except that several new operators may be used.

The Experiment

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 (Bowman *et al.* 1994), have contributed to such practice and could be further developed to meet our needs). The layer-1 construction softwares 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.

Our experiment is to demonstrate the strength of our model for information discovery. We assume the availability of layer-1 construction softwares and thus constructed layer-1 manually. Our experiment is based on Marc Vanheyningen's Unified Computer Science Technical Reports Index (UCSTRI)(VanHeyningen 1994) and is confined to computer science documents only. UCSTRI 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. We used the master index as primitive data to create our MLDB by selecting 1224 entries from four arbitrarily chosen FTP sites (University of California Berkeley, Indiana University, INRIA France and Simon Fraser University). Since an important number of documents did not have keywords attached to them, we manually deduced them or used the title and, if available, the abstract to do so. The aim of using Vanheyningen's master index as primitive data for our experiment is to compare the query results with what the conventional search engines available on the Internet can provide. The first layer of our MLDB was built based on the information provided by the four FTP sites we chose. The layer-1 of our simplified MLDB contains just one relation:

document(*file_addr, authors, affiliation, title, publication, publication_date, abstract, keywords, URL_links, num_pages, form, size_doc, time_stamp, local_ID*).

The relation with 1224 tuples constitutes our mini database on top of which we constructed a concept hierarchy for keywords. Part of the Concept hierarchy is illustrated in Fig. 1. This hierarchy was used to deduce general topics for generalization of layer-1 tuples. The generalized layer-2 relation looks as follows:

doc_summary(*affiliation, field, publication_year, count, first_authors_list, file_addr_list*).

The field field contains a high level concept which embraces all lower concepts under it. The field count (#) is a counter for the documents that correspond to affiliation, field and pub_year.

Table 2 shows a portion of doc_summary.

affiliation	field	pub_ year	#	first_ author_ list	file_ adr_ list	...
S F U	Natural Language	1993	6	Popowich, Dahl, ...	...	...
S F U	Parallel Prog.	1993	5	Liestman, ...	...	...
Indiana	Machine Learning	1994	5	Leake, Fox, ...	...	...
...	...	...	...	...	...	...

Table 2: A portion of *doc_summary*.

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 to the presentation of the detailed corresponding entries stored in layer-1. □

The same simple query submitted to UCSTRI and to WebMiner returns two different answers revealing a better hit ratio with our model. A query like:

```
select      *
from        document
related-to  Parallel Computing
where       one of keywords close_to "Gossiping"
```

would give, using UCSTRI, 50 references in the 4 targeted FTP sites, only 13 of which are indeed related to parallel computing. The same query submitted to our model, will return 21 references all related to parallel computing but with reference to gossiping or broadcasting (ie., siblings in the concept hierarchy). WebMiner not only reduces the noise by giving just documents related to the appropriate field, but also improves the hit ratio by checking synonyms and siblings in the concept hierarchies. Moreover, WebMiner will allow queries like:

```
describe    affiliation, publication_date.year
from        document
where       one of keywords like "Computational Geometry"
```

which will return the brief description of all universities or organizations that published documents about Computational Geometry with the date of publication as shown in Table 3. This query clearly does not target the documents themselves but the information about them. Note that this information is not explicitly published anywhere on the Internet but the generalization in layers of the MLDB reveals it. The question mark is due to the fact that the publication date was not available for documents served at INRIA.

affiliation	pub_year	count	count %
Simon Fraser University	1990	1	8.3%
Simon Fraser University	1991	2	16.6%
U. of Calif. Berkeley	1988	1	8.3%
U. of Calif. Berkeley	1990	3	25.0%
U. of Calif. Berkeley	1991	1	8.3%
INRIA France	?	4	33.33%

Table 3: Computational Geometry Publishcations.

For a query like:

```
describe   affiliation
from       doc_summary
where      affiliation belong_to "university"
           and field = "Machine Learning"
           and publication_year > 1990
           and count > 2
```

a simple search in the table doc_summary will produce the list of the universities which serve at least 2 documents about machine learning published after 1990 shown in Table 4. Such a query is not processable with the conventional search engines on the world wide web.

affiliation	count	count %
Indiana University	13	68.4%
Univ. of California Berkeley	6	31.6%

Table 4: 1990 Machine Learning Publishcations.

It is clear that the generalization of the MLDB allows WebMiner to mine the Internet by simply querying the meta-data summerized in different layers without accessing the artifacts themselves.

Conclusion

Different from the existing global information system services, a new approach, called *multiple layered database approach*, has been proposed and investigated for resource and knowledge discovery in global information systems. The approach is to construct a global multiple layered database by generalization and transformation of data, to store and manage multiple layered information by database technology, and to 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. The multiple layered database architecture provides a high-level, declarative query interface on which various kinds of graphical user-interfaces can be constructed. Moreover, multiple views can be defined by different users. An MLDB system may provide a global view of the current contents in a database with summary statistics. This structure allows intelligent query answering and database browsing. In addition, the layered architecture makes most searches confined to local or less remote sites on relatively small and structured databases, which will enhance the overall performance.

However, extra disk spaces are needed to store and replicate multiple layers and concept hierarchies. A cost should be paid for the development of new softwares for layer construction and query processing. Finally, a reasonable standardization may need to be introduced to enhance the quality of the services.

References

Bowman, M.; Danzig, P.; Hardy, D.; Manber, U.; and Schwartz, M. 1994. *Harvest, A scalable, Customizable Discovery and Access System.* University of Colorado: Technical Report CU-CS-732-94.

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

Han, J.; Cai, Y.; and Cercone, N. 1993. Data-driven discovery of quantitative rules in relational databases. *IEEE Trans. Knowl. & Data Eng.* 5:29–40.

Han, J.; Fu, Y.; and Ng, R. 1994. Cooperative query answering using multiple-layered databases. In *Proc. 2nd Int. Conf. Cooperative Info. Syst.*, 47–58.

Han, J.; Zaïane, O. R.; and Fu, Y. 1994. *Resource and Knowledge Discovery in Global Information Systems: A Multiple Layered Database Approach. ftp.fas.sfu.ca/pub/cs/techreports/1994/CMPT94-10.ps.Z.*

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

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

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

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

VanHeyningen, M. 1994. The unified computer sciencem technical report index: Lessons in indexing diverse resources. In *Proc. 2cd Int. Conf. on the World Wide Web.*