

11-1998

Web Warehousing System: Design and issues

Wee-Keong NG

Ee Peng LIM


Singapore Management University, eplim@smu.edu.sg

Sourav S. BHOWMICK

Sanjay Kumar MADRIA

DOI: https://doi.org/10.1007/978-3-540-49121-7_8

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

 Part of the [Databases and Information Systems Commons](#), and the [Numerical Analysis and Scientific Computing Commons](#)

Citation

NG, Wee-Keong; LIM, Ee Peng; BHOWMICK, Sourav S.; and MADRIA, Sanjay Kumar. Web Warehousing System: Design and issues. (1998). *International Workshop on Data Warehousing and Data Mining (DWDM'98), held in conjunction with International Conference on Conceptual Modeling (ER'98)*. Research Collection School Of Information Systems.

Available at: https://ink.library.smu.edu.sg/sis_research/973

This Conference Proceeding Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

Web Warehousing: Design and Issues*

S. S. BHOWMICK S. K. MADRIA W. -K. NG E.P. LIM
{sourav, askumar, wkn, aseplim}@cais.ntu.edu.sg

Center for Advanced Information Systems, School of Applied Science,
Nanyang Technological University, Singapore 639798, SINGAPORE

Abstract

The World Wide Web is a distributed global information resource. It contains a large amount of information that have been placed on the web independently by different organizations and thus, related information may appear across different web sites. To manage and access heterogeneous information on WWW, we have started a project of building a web warehouse, called WHOWEDA (*Warehouse of Web Data*). Currently, our work on building a web warehousing system has focused on building a data model and designing a web algebra. In this paper, we discuss design and research issues in a web warehousing system. The issues include are designing algebraic operators for web information access and manipulation, web data visualization and web knowledge discovery. These issues will not only overcome the limitations of available search engines but also provide powerful and friendly query mechanisms for retrieving useful information and knowledge discovery from a web warehouse.

1 Introduction

Most users obtain WWW information using a combination of search engines and browsers. However, these two types of retrieval mechanisms do not necessarily address all of a user's information needs and have the following shortcomings:

- Web browsers fully exploit hyperlinks among web pages, however, search engines have so far made little progress in exploiting link information. Not only do most search engines fail to support queries on the web utilizing link information, they also fail to return link information as part of a query's result.
- The search is limited to string matching. Numeric comparisons, as in conventional databases, cannot be done.
- Queries are evaluated on the index data rather than on the original and up-to-date data.

* This work was supported in part by the Nanyang Technological University, Ministry of Education (Singapore) under Academic Research Fund #4-12034-5060, #4-12034-3012, #4-12034-6022. Any opinions, findings, and recommendations in this paper are those of the authors and do not reflect the views of the funding agencies.

- The accuracy of results is low as there is an almost unavoidable repetition of information and existence of non-relevant data in the results.
- From the query's result returned by the search engines, a user may wish to couple a set of related Web documents together for reference. Presently, he may only do so manually by visiting and downloading these documents as files on the user's hard disk. However, this method is tedious, and it does not allow a user to retain the *coupling framework*.
- The set of downloaded documents can be refreshed (or updated) only by repeating the above procedure.
- If a user has successfully coupled a set of Web documents together, he may wish to know if there are other Web documents satisfying the same coupling framework. Presently, the only way is to request the same or other search engines for further Web documents and probe these documents manually.
- Over a period of time, there will be a number of coupled collections of Web documents created by the user. As each of these collections exists simply as a set of files on the user's system, there is no convenient way to manage and infer further useful information from them.

To overcome the limitations explained above and provide the user with a powerful and friendly query mechanism for accessing information on the web, the critical problem is to find the effective ways to build web data models of the information of interest, and to provide a mechanism to manipulate these information to garner additional useful information. The key objective of our web warehousing project, called WHOWEDA (*Warehouse of Web Data*), at the Centre for Advanced Information Systems in Nanyang Technological University, Singapore is to design and implement a web warehouse that materializes and manages useful information from the Web [22]. To meet the warehousing objective, we materialize coupled web information in the form of *web tuples* and store them in *web tables*. We define a set of web operators with web semantics to equip the warehouse with the basic capabilities to manipulate web tables and couple additional, useful, related web information residing in the web tables [9, 22]. These operators include web algebraic operations such as web select, web join, web union, web intersection, and so on [22]. In this paper, we present an overview of a web warehouse and highlight new research directions in some of the important areas of building a web warehousing system.

1.1 Related Work

There has been considerable work in data model and query languages for the World Wide Web [13, 15, 16, 20]. For example, Mendelzon, Mihaila and Milo [20] proposed a WebSQL query language based on a formal calculus for querying the WWW. The result of a WebSQL query is a set of web tuples which are flattened immediately to linear tuples. Konopnicki and Shmueli [15] proposed a high level querying system called the W3QS for the WWW whereby users may specify content and structure queries on the WWW and maintain the results of queries as database views of the WWW. In W3QL, queries are always made to the WWW. Fiebig, Weiss and Moerkotte extended relational algebra to the

World Wide Web by augmenting the algebra with new domains (data types) [13], and functions that apply to the domains. The extended model is known as RAW (Relational Algebra for the Web). Inspired by concepts in declarative logic, Lakshmanan, Sadri and Subramanian designed WebLog [16] to be a language for querying and restructuring web information. Other proposals, namely Lorel [2] and UnQL [11], aim at querying heterogeneous and semistructured information. These languages adopt a lightweight data model (based on labeled graphs) to represent data, and concentrate on the development of powerful query languages for these structures.

2 Web Information Coupling Model

In Web Information Coupling System (WICS) [9], we materialize web information as web tuples stored in web tables. Each web table is associated with a web schema. We equip the WICS with the basic capability to manipulate web tables and correlate additional useful and related web information residing in the web tables [22]. We proposed a Web Information Coupling Model (WICM) which describe web objects, web schema and a web algebra for retrieving information from the web and manipulating these information to derive additional information.

2.1 Web Objects

It consists of a hierarchy of web objects. The fundamental objects are *Nodes* and *Links*. Nodes correspond to HTML or plain text documents and links correspond to hyper-links interconnecting the documents in the World Wide Web. We define a Node type and a Link type to refer to these two sets of distinct objects. These objects consist of a set of attributes as shown below:

```
Node = [url, title, format, size, date, text]
Link = [source-url, target-url, label, link-type]
```

For the Node object type, the attributes are the URL of a Node instance and its title, document format, size (in bytes), date of last modification, and textual contents. For the Link type, the attributes are the URL of the source document containing the hyperlink, the URL of the target document, the anchor or label of the link, and the type of the link. Hyperlinks in the WWW may be characterized into three types: *interior*, *local*, and *global* [20].

The next higher level of abstraction is a **web tuple**. A web tuple is a set of connected, directed graphs each consisting of a set of **nodes** and **links** which are instances of Node and Link respectively. A collection of web tuples is called a **web table**. If the table is materialized, we associate a **name** with the table. There is a *schema* (see next section) associated with every web table. A **web database** consists of a set of web schemas and a set of web tables.

2.2 Web Schema

A web schema contains meta-information that binds a set of web tuples in a web table. Web tables are materialized results of web queries. In WICS, a user expresses a web query by describing a *query graph*.

When the query graph is evaluated, a set of web tuples each *satisfying* the query graph is harnessed from the WWW. By collecting the tuples as a table, the query graph may be used as the table's schema to bind the tuples. Hence, the web schema of a table is the query graph that is used to derive the table. Formally, a web schema is an ordered 4-tuple $M = \langle X_n, X_\ell, C, P \rangle$ where X_n is a set of node variables, X_ℓ is a set of link variables, C is a set of connectivities (in Disjunctive Normal Form), and P is a set of predicates (in Disjunctive Normal Form).

Observe that some of the nodes and links in the figures have keywords imposed on them. To express these conditions, we introduced *node* and *link variables* in the query graph. Thus, in Figure ?? node d represents those web documents which contains the words 'side effects' in the text or title. In other words, variables denote arbitrary instances of `Node` or `Link`. There are two special variables: a node variable denoted by the symbol '#' and a link variable denoted by the symbol '-'. These two variables differ from the other variables in that they are never *bound* (these variables are not defined by the predicates of the schema).

Structural properties of web tuples are expressed by a set of *connectivities*. Formally, a connectivity k is an expression of the form: $x\langle\rho\rangle y$ where $x \in X_n, y \in X_n$, and ρ is a regular expression over X_ℓ . (The angle brackets around ρ are used for delimitation purposes only.) Thus, $x\langle\rho\rangle y$ describes a path or a set of possible paths between two nodes x and y .

The last schema component is the set of predicates P . Predicates provide a means to impose additional conditions on web information to be retrieved. Let p be a predicate. If x, y are node or link variables then the following are possible forms of predicates: $p(x) \equiv [x.\text{attribute CONTAINS "A"}]$ or $p(x) \equiv [x.\text{attribute EQUALS "A"}]$ and $p(x, y) \equiv [x.\text{attribute} = y.\text{attribute}]$. where `attribute` refers to an attribute of `Node`, `Link` or `link.type`, A is a regular expression over the ASCII character set, x and y are *arguments* of p .

3 Web Algebra

The web algebra provides a formal foundation for data representation and manipulation for the web warehouse. It supports structured and topological query with sets of keywords specified on multiple nodes and on hyperlinks among the nodes. The user query is a graph-like structure and it is used to match the portions of WWW satisfying the conditions. The query result is a set of graphs called web tuples. We then define a set of web operators with web semantics to manipulate web tuples stored in a web table. The basic algebraic operators include global and local web coupling, web select, web join, web intersection, web union, etc. These operators are implemented as a part of our web query language. Briefly, these operators are discussed below. More details can be found in [5, 9, 22].

3.1 Information Access Operator

Global Web Coupling Global coupling enables a user to retrieve a set of collections of inter-related documents satisfying a web schema or coupling framework,

regardless of the locations of the documents in the Web. To initiate global coupling, the user specifies the coupling framework in the form of a *query graph*. The coupling is performed by the WIC system and is transparent to the user. Formally, the global web coupling operator Γ takes in a query (expressed as a schema M) and extracts a set of web tuples from the WWW satisfying the schema. Let W_g be the resultant table, then $W_g = \Gamma(M)$. Each web tuple matches a portion of the WWW satisfying the conditions described in the schema. These related set of web tuples are coupled together and stored in a web table. Each web tuple in the web table is structurally identical to the schema of the table. The formal details appear in [9].

3.2 Information Manipulation Operators

Web Select The **select** operation on a web table extract web tuples from a web table satisfying certain conditions. However, since the schema of web tables is more complex than that of relational tables, selection conditions have to be expressed as predicates on node and link variables, as well as connectivities of web tuples. The **web select** operation augments the schema of web tables by incorporating new conditions into the schema. Thus, it is different from its relational counterpart.

Let W be a web table with schema $M = \langle X_n, X_\ell, C, P \rangle$. Selection condition(s) on that table is denoted by another schema $M_s = \langle X_{s,n}, X_{s,\ell}, C_s, P_s \rangle$ where C_s contains the selection criteria on connectivities, and P_s contains predicates on node and link variables in $X_{s,n}$ and $X_{s,\ell}$ respectively. Formally, we define web select as follows: $W_s = \sigma_{M_s}(W)$ where σ is the select operator.

Web Project The **web project** operation on a web table extract portions of a web tuple satisfying certain conditions. However, since the schema of web tables is more complex than that of relational tables, *projection conditions* have to be expressed as node and link variables and/or connectivities between the node variables. The **web project** operation reduces the number of node and link variables in the original schema and the constraints over these variables. For more details, refer to [5].

Given a web table W with schema $M = \langle X_n, X_\ell, C, P \rangle$, a web projection on W computes a new web table W' with schema $M_p = \langle X_{n_p}, X_{\ell_p}, C_p, P_p \rangle$. The components of M_p depends on the project conditions. Formally, we define web project as follows: $W' = \pi_{\langle project_condition(s) \rangle}(W)$ where π is the symbol for project operation.

A user may explicitly specify any one of the conditions or any combination of the three conditions discussed below to initiate a web project operation.

- **Set of node variables:** To project a set of node variables from the web table.
- **Start-node variable and end-node variable:** To project all the instances of node variables between two node variables.
- **Node variable and depth of links:** To restrict the set of nodes to be projected within a limited number of links starting from the specified node variable.

Web Cartesian Product A web cartesian product, denoted by \times , is a binary operation that combines two web tables by concatenating a web tuple of one web table with a web tuple of other. If W_i and W_j are web tables with n and m web tuples respectively, the resulting web table W created by web cartesian product consists of $n \times m$ web tuples.

Local Web Coupling Given two web tables, local coupling is initiated explicitly by specifying a pair(s) of web documents (*coupling node variables*) and a set of keyword(s) to relate them. The result of local web coupling is a web table consisting of a set of collections of inter-related Web documents from the two input tables. To elaborate further, let W_i and W_j be two web tables with schemas $M_i = \langle X_{i,n}, X_{i,\ell}, C_i, P_i \rangle$ and $M_j = \langle X_{j,n}, X_{j,\ell}, C_j, P_j \rangle$ respectively. Let w_i and w_j be two web tuples from W_i and W_j , and $n_c(w_i)$ and $n_c(w_j)$ are instances of node variables n_{c_i} and n_{c_j} respectively. Suppose documents at <http://www.virtualdisease.com/cancer/index.html> (represented by node $n_c(w_i)$) and <http://www.virtualdrug.com/cancerdrugs/index.html> (represented by node $n_c(w_j)$) contain information related to cancer and appears in w_i and w_j respectively. Tuples w_i and w_j are *coupling-compatible locally* on $n_c(w_i)$ and $n_c(w_j)$ since they both contain similar information (information related to cancer).

We express local web coupling between two web tables as follows:

$$W = W_i \otimes_{(\langle \text{node_pair} \rangle, \langle \text{keyword(s)} \rangle)} W_j$$

where W_i and W_j are the two web tables participating in the coupling operation and W is the coupled web table satisfying a schema $M = \langle X_n, X_\ell, C, P \rangle$. In this case, $\langle \text{node_pair} \rangle$ specifies a pair of coupling node variables in the web table W_i and W_j , and $\langle \text{keyword(s)} \rangle$ specifies a list of keyword(s) on which the similarity between the coupling node variable pair is evaluated.

Web Join The web join operator combines two web tables by *concatenating* a web tuple of one table with a web tuple of other table whenever there exists *joinable nodes*. Let W_i and W_j be two web tables with schemas $M_i = \langle X_{i,n}, X_{i,\ell}, C_{i,p}, P_i \rangle$ and $M_j = \langle X_{j,n}, X_{j,\ell}, C_j, P_j \rangle$ respectively. Then W_i and W_j are *joinable* if and only if there exist at least one node variable in M_i and in M_j which refers to identical (having the same URL) node or web document.

Consider the following predicates of the node variables c and z where $c \in X_{i,n}$ and $z \in X_{j,n}$:

$$\begin{aligned} p_{i_a}(c) &\equiv [c.\text{url EQUALS "http://www.singapore.com/area/"}, \\ p_{j_a}(z) &\equiv [z.\text{url EQUALS "http://www.singapore.com/area/"}] \end{aligned}$$

Since the node variables c and z of M_i and M_j respectively refers to the same web document at URL 'http://www.singapore.com/area/', the web tables W_i and W_j are joinable. The joinable nodes are c and z .

Formally, we define $W = W_i \bowtie W_j$ is a set of web tuples satisfying schema $M = \langle X_n, X_\ell, C, P \rangle$ where X_n is the set of node variables appearing in P , X_ℓ is the set of link variables appearing in P , C and P are obtained from M_i and M_j . We discussed web join operator in detail in [10].

Schema Tightness In a web warehouse, a user expresses a web query by describing a query graph or coupling framework. The query graph is used as the schema of the web table to bind the web tuples. In reality, it is unrealistic to assume from the (naive) user complete knowledge of the structure of the query graph. The user may express some incomplete graph structure based on the partial knowledge. Thus, such query graphs, if used as schema of the web table, may contain unbound nodes and links. Furthermore, a web schema serves two important purposes: First, it enables users to understand the structure of the web table and form meaningful queries over it. Second, a query processor relies on the schema to devise efficient plans for computing query results. Both the tasks become significantly harder when the schema contains unbound nodes and links.

To address these challenges, we design and implement a web operator called *schema tightness operator*. The schema tightness operator takes as input a web table containing unbound nodes and links and web schema and *tighten* the web schema of the web table by imposing constraints on the unbound nodes and links in the web table.

4 Web Data Visualization

A query graph returns a set of web tuples which are stored in a web table. However, a user may wish to view these web tuples in different framework. Here we present some data visualization operators to add flexibility in viewing query results coupled from the WWW and to generate some additional useful information.

4.1 Web Ranking Operators

Presently, in the WICM we have web operators to manipulate information from the WWW globally and locally. A crucial problem that a WIC system faces is extracting *hot tuples* for a user query (the most relevant web tuples). This problem is challenging because hot tuples might not be displayed at the beginning of the web table. We are developing two web ranking operators; *global ranking* and *local ranking* operators to rank web tuples generated by global and local web coupling respectively. The ranking operators are based on the following factors:

- Number of occurrence of a keyword in a document.
- Location of occurrence of a keyword in a document, i.e., whether the keyword occurs in the title, text or anchor of a document.
- Overlap between all pairs of web documents, i.e., if A , B and C are documents in three web tuples considered relevant (in that order) to a user query, then we believe that the “interestingness” of B is lower than C if B overlaps with A significantly, while C is a distinct result. Thus, our ranking function will rank web tuple containing C before the web tuple containing B .

4.2 Data Visualization Operators

Presently, in the WICM we have a set of web operators to manipulate information from the WWW. Web information is materialized and displayed in the form of web tuples stored in a web table. This approach of displaying information to the user has few shortcomings:

- It does not provide us with the ability to see the overall structure of the information captured in the web table. It is not possible for a user to visualize how one web tuple in a web table is related to another.
- The set of web tuples in a web table may contain duplicate web documents. There is no mechanism to provide a *coalesced view* of the set of web tuples. A coalesced view allows a user to browse lesser number of directed connected graphs when locating information.
- It does not allow a user to group web tuples based on *related information content*, or *similar (or identical)* web sites. A user has to manually probe each web tuple to find these information. However, these information cannot be grouped together in our web table.
- The set of web tuples are materialize in web table. There is no other representation of these web tables. For example, the collective view of these web tuples can be stored as a set of directed graphs having lesser number of nodes or links as compared to the original web table.

To resolve the above difficulties, we have introduced to following data visualization operators:

- **Web Nest:** This operator allows one to visualize relationships between web tuples in a web table. It provides an overview of the information space harnessed from the WWW and shows how these information are topically related.
- **Web Unnest:** This operator returns the original web table from a set of directed connected graphs created by the web nest operation.
- **Web Coalesce:** This operator coalesces duplicate nodes to reduce the size of the number of directed connected graphs in a web table. A coalesced web table is a set of condensed graphs and allow a user to browse lesser number of graphs (web tuples).
- **Web Expand:** This operator expands a coalesced web table to recover the original set of web tuples.
- **Web Pack:** This operator groups a web tuples based on related (or similar) information content, and similar (or identical) web sites.
- **Web Unpack:** This operator returns the original set of web tuples from the packed web table.
- **Web Sort:** This operator sorts web tuples in *ascending* or *descending* order based on the total number of nodes or link types (local, global or interior) in a web tuple.

These web operators take as input a set of web tuples of a web table and provide a different view of the tuples as output. This gives users the flexibility to view documents in perspectives that are more meaningful. These operators provide different storage representation of web tuples which will help in optimizing the query, storage and maintenance cost. These different representations may also lead to inconsistency problems with respect to the original web table. Currently, we are investigating these problems.

5 Web Data Mining

The resulting growth in on-line information combined with the almost unstructuredness of web data necessitates the development of powerful yet computationally efficient web mining tools. Web mining can be defined as the discovery and analysis of useful information from WWW data. Web mining involves three types of data; data on the WWW, web documents structure and the data on users who browse web pages. Thus, web data mining should focus on three issues; content-based mining [17], web usage mining [23] and web structure mining. Web content mining describes the automatic search of information resources available on-line. Web usage mining includes the mining of data from the server access logs, user registration or profiles, user sessions or transactions, etc. Web structure mining involves mining web document's structures and links. A survey of some of the emerging tools and techniques for web usage mining is given in [21]. One of the important areas in WHOWEDA involves the development of tools and techniques for mining useful information from the web.

The web contains a mix of many different data types, and in a sense subsumes text data mining, database data mining, image mining, and so on. The web contains additional data types that are not available in a large scale before, including hyperlinks and massive amounts of (indirect) user usage information. Spanning across all these data types is the dimension of time, since data on the web changes over time. Finally, there is data that are generated dynamically, in response to user input and programmatic scripts. In WHOWEDA, we primarily focus on mining useful information from these different data types. For further information related to web mining in WHOWEDA refer to [7].

5.1 Web Bags and Knowledge Discovery

Most of the search engines fail to handle the following knowledge discovery goals:

- From query results returned by search engines, a user may wish to locate the most *visible* Web sites [4] or documents for reference. That is, sites or documents which can be reached by many paths (high fan-in).
- Reversing the concept of visibility, a user may wish to locate the most *luminous* Web sites [4] or documents for reference. That is, web sites or documents which have the most number of outgoing links.
- Furthermore, a user may wish to find out the most traversed path for a particular query result. This is important since it helps the user to identify the set of most popular interlinked Web documents which are traversed frequently to obtain the query result.

We have introduced the concept of web bag in [5] and used web bags for knowledge discovery. Informally, a web bag is a web table containing multiple occurrences of the identical web tuples. A web tuple is a set of inter-linked documents retrieved from the WWW that satisfy a query graph. A web bag may only be created by projecting some of the nodes from the web tuples of a web table using the web project operator. A web project operator is used to isolate data of

interest, allowing subsequent queries to run over a smaller, perhaps more structured web data. Unlike its relational counterpart, a web project operator does not eliminate identical web tuples autonomously. The projected web table may contain identical web tuples, thus forming a web bag. The duplicate elimination process is initiated explicitly by a user. Autonomous duplicate elimination may hinder the possibility of discovering useful knowledge from a web table. This is due to the fact that these knowledge may only be discovered from the web bags.

5.2 Warehouse Concept Mart (WCMart)

Due to the large amount of data on the WWW, knowledge discovery from web data becomes more and more complex. We propose the building of concept hierarchies from web documents and use them in discovering new knowledge. We call the collection of concepts a Warehouse Concept Mart (WCMart). Concept marts are built by extracting and generalizing terms from web documents to represent classification knowledge of a given class hierarchy. For unclassified words, they can be clustered based on their common properties. Once the clusters are decided, the keywords can be labeled with their corresponding clusters, and common features of the terms are summarized to form the concept description. We may associate a weight at each level of concept marts to evaluate the importance of a term with respect to the concept level in the concept hierarchy. Concept marts can be used for the following:

- *Intelligent answering of web queries:* Knowledge discovery using concept marts facilitates querying web data and intelligent query answering in web warehousing system. A user can supply the threshold for a given key word in the concept mart and the words with the threshold above the given value can be taken into account while answering the query. The query can also be answered using different levels of concept marts [14] or can provide approximate answers [19].
- *Ranking result tuples of a query:* In our model, tuples returned as a result of a web query are stored in a web table. We use warehouse concept marts for ranking these tuples so that the most relevant tuples are returned in response to a user's query. The user may rank tuples interactively by specifying various threshold values and concept levels.
- *Global information coupling across the WWW:* In our model, we introduce the concept of web information coupling. It refers to an association of related web documents. We use concept marts to define this "association" among web documents. Since the coupling is initiated by the user, he may supply threshold values for keywords in the concept mart to be used in the coupling.
- *Web mining:* Concept marts can also be used for web data mining. Web mining so far in the literature is restricted to mining web access patterns and trends by examining the web server log files [23]. An alternative is to make use of web concept marts in generating some useful knowledge. We may use association rule techniques to mine associations between words appearing in the concept mart at various levels, and the query graph. Mining knowledge

at multiple levels may help WWW users discover interesting rules which are difficult to find otherwise.

- *Characterization of web tables*: In a web warehouse, we store tuples in web tables. We need to categorize these tables so that a user's query can be directed to the appropriate table(s). Based on the classification of web tables, a user may also specify tables to be used for evaluating his query. We categorize the web tables using the concept marts. We generate concept marts from web tables and classify a set of tables by known classes or concepts.

6 Conclusions

In this paper, we have reported an overview of a Web Warehousing Project (WHOWEDA) [22] at the Nanyang Technological University, Singapore and discussed some interesting research ideas in that context. Building a warehouse that accommodates data from the WWW has required us to rethink nearly every aspect of conventional data warehousing and relational technology. This paper brings together some of the important techniques required for designing a web warehouse and generating useful knowledge. In particular, our focus is on web data model and an algebra for web information access, manipulation and visualization. We have also discussed the motivation for developing a Web Concept Mart and its application in web warehousing. Furthermore, we have outline the problem of web mining and maintenance of web information. We believe that issues presented here will serve as an interesting example for further discussion.

References

1. <http://www.cais.ntu.edu.sg:8000/~whoweda/>.
2. S. ABITEBOUL, D. QUASS, J. MCHUGH, J. WIDOM, J. WEINER. The Lorel Query Language for Semistructured Data. *Journal of Digital Libraries*, 1(1):68-88, April 1997.
3. G. AROCENA, A. MENDELZON. WebOQL: Restructuring Documents, Databases and Webs. *Proceedings of International Conference on Data Engineering*, Orlando, Florida, February 1998.
4. T. BRAY. Measuring the Web. *Proceedings of the 5th International World Wide Web Conference (WWW)*, Paris, France, 1996.
5. S. BHOWMICK, S. K. MADRIA, W.-K. NG, E.-P. LIM. Web Bags: Are They Useful in A Web Warehouse? *Proceedings of 5th International Conference of Foundation of Data Organization (FODO'98)*, Kobe, Japan, November 1998.
6. S. BHOWMICK, S. K. MADRIA, W.-K. NG, E.-P. LIM. Data Visualization in a Web Warehouse. *Proceedings of International Workshop on Data Warehousing and Data Mining (DWDM'98) (in conjunction with ER'98)*, Singapore, 1998.
7. S. BHOWMICK, S. K. MADRIA, W.-K. NG, E.-P. LIM. Web Mining in WHOWEDA: Some Issues. *PRICAI'98 Workshop on Knowledge Discovery and Data Mining*, Singapore, 1998.
8. S. BHOWMICK, S. K. MADRIA, W.-K. NG, E.-P. LIM. Bags in A Web Warehouse: Design and Analysis. *Submitted for publication*.

9. S. BHOWMICK, W.-K. NG, E.-P. LIM. Information Coupling in Web Databases. *Proceedings of the 17th International Conference on Conceptual Modelling (ER'98)*, Singapore, 1998.
10. S. S. BHOWMICK, W.-K. NG, E.-P. LIM, S. K. MADRIA. Join Processing in Web Databases. *Proceedings of the 9th International Conference on Database and Expert Systems Application (DEXA)*, Vienna, Austria, 1998.
11. P. BUNEMAN, S. DAVIDSON, G. HILLEBRAND, D. SUCIU. A query language and optimization techniques for unstructured data. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Canada, June 1996.
12. M. FERNANDEZ, D. FLORESCU, A. LEVY, D. SUCIU. A Query Language for a Web-Site Management Systems *SIGMOD Record*, 26(3), Sept, 1997.
13. T. FIEBIG, J. WEISS, G. MOERKOTTE. RAW: A Relational Algebra for the Web. *Workshop on Management of Semistructured Data (PODS/SIGMOD'97)*, Tucson, Arizona, May 16, 1997.
14. J. HAN, Y. HUANG, N. CERCONI, Y. FU. Intelligent Query Answering by Knowledge Discovery Techniques. *IEEE Transactions of Knowledge and Data Engineering.*, 8(3):373 – 390, 1996.
15. D. KONOPNICKI, O. SHMUELI. W3QS: A Query System for the World Wide Web. *Proceedings of the 21st International Conference on Very Large Data Bases*, Zurich, Switzerland, 1995.
16. L.V.S. LAKSHMANAN, F. SADRI., I.N. SUBRAMANIAN. A Declarative Language for Querying and Restructuring the Web *Proceedings of the Sixth International Workshop on Research Issues in Data Engineering*, February, 1996.
17. S. H. LIN, C. S. SHIH, M. C. CHANG CHEN ET AL. Extracting Classification Knowledge of Internet Documents with Mining Term Associations: A Semantic Approach. *Proceedings of the Sixth International Workshop on Research Issues in Data Engineering*, February, 1996.
18. M. LIU, T. GUAN, L. V. SAXTON. Structured-Based Queries over the World Wide Web. *Proceedings of the 17th International Conference on Conceptual Modeling (ER'98)*, Singapore, 1998.
19. S. K. MADRIA, M. MOHANIA, J. F. RODDICK. A Query Processing Model for Mobile Computing using Concept Hierarchies and Summary Databases. *Submitted for publication*.
20. A. O. MENDELZON, G. A. MIHAILA, T. MILO. Querying the World Wide Web. *Proceedings of the International Conference on Parallel and Distributed Information Systems (PDIS'96)*, Miami, Florida.
21. B. MOBASHER, R. COOLEY, J. SHRIVASTAVA. Web Mining: Information and Pattern Discovery on the World Wide Web. *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, November 1997.
22. W. K. NG, E.-P. LIM, C. T. HUANG, S. BHOWMICK, F. Q. QIN. Web Warehousing: An Algebra for Web Information. *Proceedings of IEEE International Conference on Advances in Digital Libraries (ADL'98)*, Santa Barbara, California, April 22–24, 1998.
23. O. R. ZAINE, M. XIN, J. HAN. Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. *Proceedings of IEEE International Conference on Advances in Digital Libraries (ADL'98)*, Santa Barbara, California, April 22–24, 1998.