01 Jan 1999

Pi-Web Join in a Web Warehouse

Sanjay Kumar Madria
Missouri University of Science and Technology, madrias@mst.edu

Wee Keong Ng

Ee-Peng Lim

Sourav S. Bhowmick

Recommended Citation

S. K. Madria et al., "Pi-Web Join in a Web Warehouse," *Proceedings of the 6th International Conference on Database Systems for Advanced Applications, 1999*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1999.
The definitive version is available at https://doi.org/10.1109/DASFAA.1999.765759

Π-Web Join in A Web Warehouse*

Sourav S Bhowmick Sanjay K Madria Wee-Keong Ng Ee-Peng Lim
Centre for Advanced Information Systems
Nanyang Technological University
Singapore 639798
{p517026, askumar, awkng, aseplim}@ntu.edu.sg

Abstract

With the enormous amount of data stored in the World Wide Web, it is increasingly important to design and develop powerful web warehousing tools. The key objective of our web warehousing project, called WHOWEDA *(*W*are*h*ouse of* W*eb* D*ata), is to design and implement a web warehouse that materializes and manages useful information from the Web. In this paper, we introduce the concept of* Π-web join *in the context of* WHOWEDA. Π-*web join operator is a* web information manipulation operator *to combine* relevant *web information residing in two* web tables. *Informally, it is the combination of* web join *and* web project *operators which filter out irrelevant information from a joined web table. In this paper, we show how to construct the* Π-*joined web table and its schema. We also highlight the benefits of the* Π-*web join operator.*

1. Introduction

Currently, web information may be discovered primarily by two mechanisms; browsers and search engines. This form of information access on the Web has a few shortcomings [7]. To resolve these limitations, we introduced Web Information Coupling System (WICS) [7], a database system for managing and manipulating coupled information extracted from the Web. WICS is one of the component of our web warehouse, called WHOWEDA (*W*are*h*ouse *o*f *We*b *Da*ta) [1]. In WICS, we materialize web information as *web tuples* and store them in *web tables*. We equip WICS with the basic capability to manipulate web tables and correlate additional, useful, related web information residing in the web tables [14]. Note that a web table is a collection of directed graphs (i.e., web tuples).

We have introduced the web join operator in [8, 14] as a web information manipulation operator in WICS. The web join operator couples related information from two web tables by concatenating a web tuple of one table with a web tuple of other table whenever there exists instances of *joinable node variables* (identical nodes).

1.1. Motivation

Example 1 Assume a web site at `http://www.panacea.org/` which stores disease and drug related information. Suppose we have the following two web tables in our warehouse constructed by coupling related information from the web site at `http://www.panacea.org/`:

1. Web table Diseases stores a list of diseases and their symptoms, treatments and evaluation details. Figures 1 and 3 depict the *web schema* of Diseases and a partial view of the web table respectively [1]. Note that the web schema is also called the query graph and is explicitly specified by the user in order to couple information from the Web.
2. Web table Drugs stores a list of drugs for various diseases and their side effects and uses. Figures 2 and 4 describes the web schema of Drugs and a partial view of the web table respectively.

Suppose a user wants to extract information related to symptoms of various diseases and side effects of drugs used for these diseases using WICS. Clearly, these information are already stored in tables Diseases and Drugs. The web join operator enabes us to relate the information from the

*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.

[1] Note that in all figures in this paper, the boxes and directed lines correspond to *nodes* (Web documents) and *links* (hyperlinks) respectively. Observe that some of the nodes and links have keywords imposed on them. These keywords express the content of the web document or the label of the hyperlink between the web documents. The dashed arrows signifies the existence of *unbound* node and/or link variables

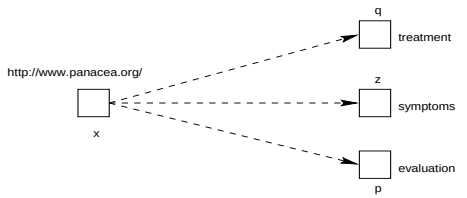

Figure 1. Query graph (web schema) of "Diseases".

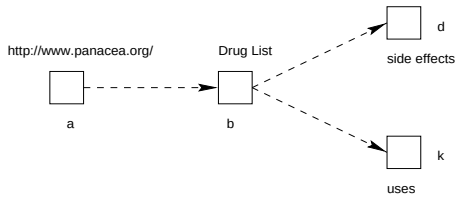

Figure 2. Query graph (Web schema) of "Drugs".

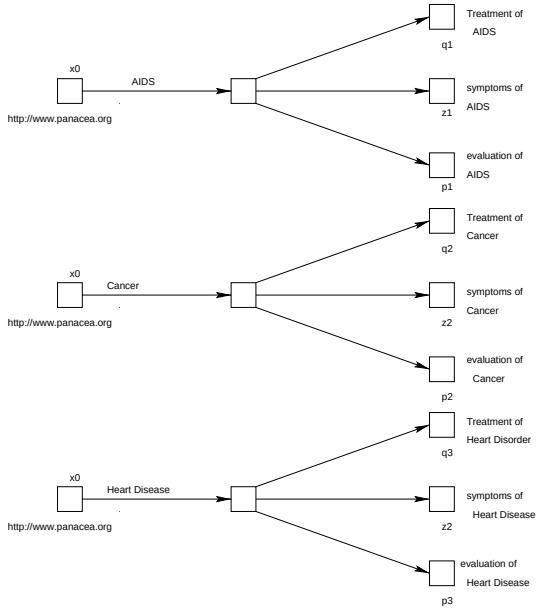

Figure 3. Partial view of web table "Diseases".

two web tables. In particular, web join concatenates web tuples in the web tables based on the joinable node variables (a and x in Figures 1 and 2 since they have identical URL). The joined schema and a portion of the joined web table are shown in Figures 5 and 6 respectively. The black boxes in Figures 5 and 6 refer to the joinable node variable and instances of it over which concatenation of the web schemas (coupling frameworks) and web tuples is performed respectively. Each web tuple in the joined web table contains information related to the symptoms, treatment and evaluation procedures of a particular disease, and a drug with their uses and side effects.

Although it seems to be advantageous to couple related information from two web tables using web join operator, in many cases the web join operation may also couple irrelevant information. For example, consider the web schemas in Figures 1 and 2. The node variables related to the treatment and evaluation strategies of diseases (node variables q and p), and the list of drugs and their uses (node variables b and k) are irrelevant for the user in the above example. This is because he is only interested in information related to symptoms of various types of diseases and side effects of various drugs. These irrelevant information are present in the joined schema (Figure 5) as web join operation does not provide a mechanism to eliminate these irrelevant node variables. Ideally, the structure of the user's query should be similar to the web schema (query graph) in Figure 8. ∎

In this paper, we introduce the concept of Π-*web join* to resolve the above limitation. Π-web join operation is an important extension of the web join operation that filters out irrelevant information from the joined web table. The basic idea of Π-web join is as follows: Given two web tables, Π-web join is initiated explicitly by the user by specifying a set of web documents (node variables) to be eliminated from the result of web join operation. The result of Π-web join is a web table consisting of a set of collections of interrelated relevant Web documents from the two input tables. Note that in many cases, the size of this web table will be significantly smaller than the size of the web table created by web join operation. Thus, the Π-web join operator may be used to isolate data of interest in a joined web table, allowing subsequent queries to run over a smaller, perhaps more structured web data. Note that Π-web join eliminates irrelevant information, thus it reduces the browsing time. Due to the reduction in the size of the web tables, the communication cost of *distributed* web join process over the geographically separated web tables is minimized.

2 Related Work

There has been considerable work in data model and query languages for the World Wide Web [11, 12, 13]. For example, Mendelzon, Mihaila and Milo [13] proposed a WebSQL query language based on a formal calculus for querying the WWW. The result of WebSQL query is a set of web tuples which are flattened immediately to linear tuples. Konopnicki and Shmueli [11] 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.

Lakshmanan, Sadri and Subramanian designed WebLog [12] to be a language for querying and restructuring web information. Other proposals, namely Lorel [2] and UnQL [9], aim at querying heterogeneous and semistructured information. These languages adopt a lightweight data model to represent data, based on labeled graphs, and concentrate on the development of powerful query languages for these structures. The WebOQL system [3] supports a general class of data restructuring operations in the context of the Web. It synthesizes ideas from query languages for the Web, for semistructured data and for website restructuring. The data model proposed in WebOQL is based on ordered trees where a web is a graph of trees. This model enables us to navigate, query and restructure graphs of trees. In this system, the *concatenate* operator allows us to juxtapose two trees which can be viewed as the manipulation of trees. However, none of the above systems addresses the issues similar to Π-web join in WICS.

3. Preliminaries

3.1. WHOWEDA - Warehouse Of WEb DAta

With the enormous amount of data stored in the World Wide Web, it is increasingly important to develop powerful web warehousing and web data mining tools for querying and analysis of such data and to generate interesting knowledge from it. The key objective of our web warehousing project, called WHOWEDA (*W*are*h*ouse *o*f *We*b *Da*ta), is to design and implement a web warehouse that materializes and manages useful information from the Web [14].

WHOWEDA is a data repository of useful, relevant web information, available for querying and analysis. As relevant information becomes available in the WWW, these information are coupled from various sources, translated into a common web data model (*Web Information Coupling Model*), and integrated with existing data in WHOWEDA. In the next section, we briefly describe WICS.

3.2. Web Information Coupling System

The primary components of WICS is a web data model called *Web Information Coupling Model* (WICM) and an algebra called *Web Information Coupling Algebra* (WICA) for retrieving information from the Web and manipulating these information to derive additional useful information.

3.2.1 Web Information Coupling Model (WICM)

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 such as: `Node = [url, title, format, size, date, text]` and `Link = [source-url, target-url, label, link-type]`. Note that hyperlinks in the WWW may be characterized into three types: *interior*, *local*, and *global* [13].

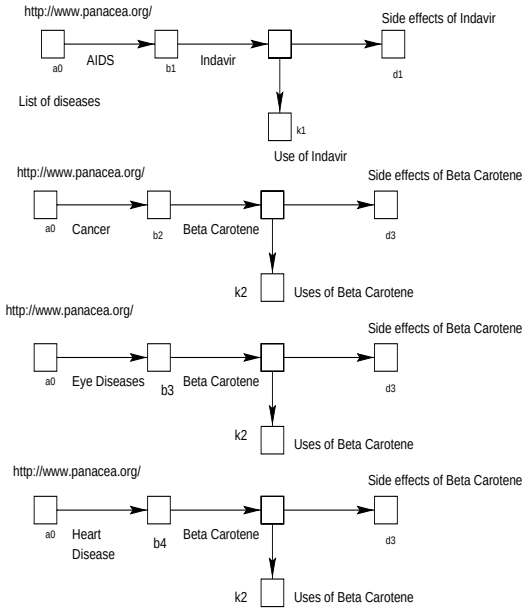

Figure 4. Partial view of web table "Drugs".

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 tables.

Web Schema

A web schema contains meta-information that binds the web tuples in a web table. Web tables are materialized results of web queries. In WICS, a user expresses a web query using a query graph. A query graph is a directed, connected graph consisting of nodes and links.

When the query graph in Figure 1 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 $S = \langle X_n, X_\ell, C, P \rangle$ 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 2, node d represents those web documents which contain 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 assigned any predicates in 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.

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: $p(x) \equiv [x.\texttt{attribute CONTAINS "}A\texttt{"}]$ or $p(x) \equiv [x.\texttt{attribute EQUALS "}A\texttt{"}]$ and $p(x, y) \equiv [x.\texttt{attribute} = y.\texttt{attribute}]$ 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.

Consider the query graphs (Figures 1 and 2) in Example 1. The web schemas of these query graphs are given below:

Example 2 *Produce a list of diseases and their symptoms, treatment and evaluation procedures starting from the web site at* `http://www.panacea.org/`.

We express the schema of the above query by $S_i = \langle X_{n_i}, X_{\ell_i}, C_i, P_i \rangle$ where $X_{n_i} = \{x, z, q, p\}, X_{\ell_i} = \{\texttt{-}\}$, and $C_i \equiv k_{i_1} \wedge k_{i_2} \wedge k_{i_3}$ such that $k_{i_1} = x\langle\texttt{-}^+\rangle z, k_{i_2} = x\langle\texttt{-}^+\rangle p, k_{i_3} = x\langle\texttt{-}^+\rangle q$ and $P_i \equiv p_{i_1} \wedge p_{i_2} \wedge p_{i_3} \wedge p_{i_4}$ such that $p_{i_1}(x) \equiv [x.\texttt{url EQUALS "http://www.panacea.org/"}], p_{i_2}(p) \equiv [p.\texttt{title CONTAINS "evaluation"}], p_{i_3}(z) \equiv [z.\texttt{title CONTAINS "symptoms"}], p_{i_4}(q) \equiv [q.\texttt{title CONTAINS "treatment"}]$. Note that `-` denotes one unbound link and `-`$^+$ denotes one or more unbound links. ∎

Example 3 *Produce a list of drugs for various diseases, their side effects and uses starting from the web site at* `http://www.panacea.org/`.

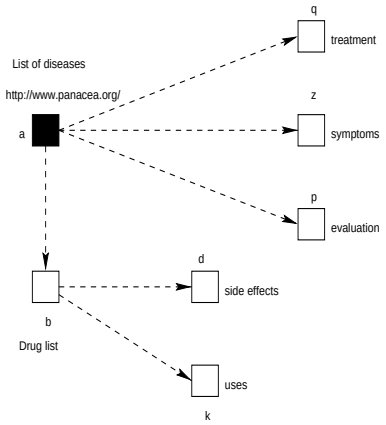

Figure 5. Web schema of joined web table.

We express the schema of the above query by $S_j = \langle X_{n_j}, X_{\ell_j}, C_j, P_j \rangle$ where $X_{n_j} = \{a, b, k, d\}$, $X_{\ell_j} = \{\texttt{-}\}$, $C_j \equiv k_{j_1} \wedge k_{j_2} \wedge k_{j_3}$ such that $k_{j_1} = a\langle\texttt{-}\rangle b$, $k_{j_2} = b\langle\texttt{-}^+\rangle d$, $k_{j_3} = b\langle\texttt{-}^+\rangle k$ and $P_j \equiv p_{j_1} \wedge p_{j_2} \wedge p_{j_3} \wedge p_{j_4} \wedge p_{j_5}$ such that $p_{j_1}(a) \equiv [a.\texttt{url EQUALS "http://www.panacea.org/"}]$, $p_{j_2}(b) \equiv [b.\texttt{title CONTAINS "Drug List"}]$, $p_{j_3}(k) \equiv [k.\texttt{title CONTAINS "uses"}]$, $p_{j_4}(d) \equiv [d.\texttt{title CONTAINS "side effects"}]$. ∎

3.2.2 Web Information Coupling Algebra (WICA)

The Web Information Coupling Algebra (WICA) provides a formal foundation for data representation and manipulation for the web warehouse. The basic algebraic operators include global and local web coupling, web select, web project, web join, web intersection, web union, etc [5]. In this section, we briefly describe the web project and web join operator which we need explain the concept of Π-web join. Then, we introduce the concept of *web bag*. Web bag is a web table with duplicate web tuples and is a by-product of web project operation. A complete description of web project and web bag is given in [4].

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 exist joinable nodes (identical nodes or web documents). Let W_i and W_j be two web tables with schemas $S_i = \langle X_{n_i}, X_{\ell_i}, C_i, P_i \rangle$ and $S_j = \langle X_{n_j}, X_{\ell_j}, C_j, P_j \rangle$ respectively. Then W_i and W_j are *joinable* if and only if there exist at least one node variable (joinable node variable) in S_i and in S_j which refers to an identical (having the same content) node or web document.

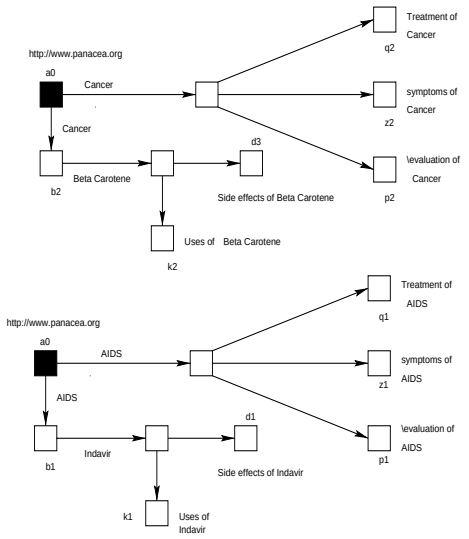

Figure 6. Partial view of the joined web table.

Consider the following predicates of the node variables x and a in Examples 2 and 3 where $a \in X_{n_j}$ and $x \in X_{n_i}$: $p_{j_1}(a) \equiv [a.\texttt{url EQUALS "http://www.panacea.org/"}], p_{i_1}(x) \equiv [x.\texttt{url EQUALS "http://www.panacea.org/"}]$. Since the node variables a and x of S_i and S_j respectively refers to the same web document at URL '`http://www.panacea.org/`', the web tables W_i (Diseases) and W_j (Drugs) are joinable [2]. The joinable nodes are a and x. The joined web tuples are materialized in a separate web table. As one of the joinable nodes with identical URLs is superfluous in the resulting web table, we keep only one joinable node variable (i.e, a or x).

Formally, we define $W_{wj} = W_i \bowtie W_j$ is a set of web tuples satisfying schema $S_{wj} = \langle X_{n_{wj}}, X_{\ell_{wj}}, C_{wj}, P_{wj} \rangle$ where $X_{n_{wj}}$ is the set of node variables appearing in P_{wj}, $X_{\ell_{wj}}$ is the set of link variables appearing in P_{wj}, C_{wj} and P_{wj} are obtained from S_i and S_j. For further details on web join, the reader is referred to [8, 14].

Example 4 Consider the Diseases and Drugs tables as described in Example 1. These two web tables are joinable over the node variables x and a since the instances of x and a are identical (identical URL `http://www.panacea.org/`). Performing a web join on these web tables creates a joined web table (Figure 6). The schema of this joined web table is shown in Figure 5. Formally, let the joined schema be $S_{wj} = \langle X_{n_{wj}}, X_{\ell_{wj}}, C_{wj}, P_{wj} \rangle$ where $X_{n_{wj}} = \{a, z, q, p, b, k, d\}$, $X_{\ell_{wj}} = \{\text{-}\}$, $C_{wj} \equiv k_1 \wedge k_2 \wedge k_3 \wedge k_4 \wedge k_5 \wedge k_6$ such that $k_1 = a\langle\text{-}\rangle b$, $k_2 = b\langle\text{-}^+\rangle d$, $k_3 = b\langle\text{-}^+\rangle k$, $k_4 = a\langle\text{-}^+\rangle p$, $k_5 = a\langle\text{-}^+\rangle z$, $k_6 = a\langle\text{-}^+\rangle q$, and $P_{wj} \equiv p_1 \wedge p_2 \wedge p_3 \wedge p_4 \wedge p_5 \wedge p_6 \wedge p_7$ such that $p_1(a) \equiv [a.\texttt{url EQUALS "http://www.panacea.org/"}]$, $p_2(b) \equiv [b.\texttt{title CONTAINS "Drug List"}]$, $p_3(k) \equiv [k.\texttt{title CONTAINS "uses"}]$, $p_4(d) \equiv [d.\texttt{title CONTAINS "side effects"}]$, $p_5(p) \equiv [p.\texttt{title CONTAINS "eval uation"}], p_6(z) \equiv [z.\texttt{title CONTAINS "symptoms"}], p_7(q) \equiv [q.\texttt{title CONTAINS "treatment"}]$. ∎

[2] We assume that the last modification dates of these pages are identical. That is, web documents with same URLs are identical in contents.

Web Project

The web project operation on a web table extracts portions of a web tuple satisfying certain *project conditions*. These conditions are expressed as node and link variables and/or connectivities between the node variables. The web project is used to isolate data of interest in a web table, allowing subsequent web queries to execute over smaller web table, perhaps having more complete web schema.

Given a web table W with schema $S = \langle X_n, X_\ell, C, P \rangle$, a web project on W computes a new web table W' or a web bag W_b with schema $S_p = \langle X_{n_p}, X_{\ell_p}, C_p, P_p \rangle$. The components of S_p depends on the project condition(s). Note that, unlike relational project, the web project operation does not remove duplicate web tuples automatically. The projected collection of web tuples may contain identical web tuples. In this case, it is called a web bag. Formally, we define web project as $W_b = \pi_{\langle project\_condition(s) \rangle}(W)$ where π is the web project operator. The duplicate elimination process is then initiated explicitly by the user and is performed by the following operation: $W' =$ **Distinct(**W_b**)** where W_b is a web bag and W' is the projected web table with distinct web tuples. Note that if a web bag is not created after a web project operation then $W' = \pi_{\langle project\_condition(s) \rangle}(W)$. Note that in web project, we specify the node variables to be eliminated in the project conditions, as opposed to relational project, where we specify the attributes to be projected from a relation.

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

- **Set of node variables:** A set of node variables to eliminate from the web table.
- **Start-node variable and end-node variable:** To eliminate all the instances of node variables between two node variables.
- **Node variable and depth of links:** This condition restricts the set of nodes to be eliminated within a limited number of links starting from the specified node variable.

Example 5 Continuing with Example 1, the set of project conditions P_c for the joined web table is given as: $P_c = \{(b, k, q, p)$, (start-node variable a and end-node variable q), (start-node variable a and end-node variable p), (start-node variable b and end-node variable k), (start-node variable b and end-node variable $d)\}$. ∎

Web Bag

Informally, a web bag is a web table containing multiple occurrences of *identical web tuples*[4]. Recall that a web tuple is a set of inter-linked documents retrieved from the WWW which satisfies a query graph. A web bag may only be created by eliminating some of the nodes from web tuples of a web table using the web project operator. As we shall see in Section 4, a web bag may be created due to Π-web join operation. For example, consider the collection of web tuples in Figure 7. The third and fourth web tuples (web tuples with black nodes), denoted by t_3 and t_4 are identical, i.e., $t_3 = t_4$. This is because the set of URLs and the connectivities of the nodes are identical in both the web tuples. Thus, the collection of web tuples can be considered as a web bag.

4 Concept of Π-Web Join

In this section, we start by formally defining the Π-web join operator. Then, we show how to construct the Π-joined web table and its schema.

4.1 Definition

In WICS, a web join followed by a web project operation is used quite commonly to eliminate irrelevant nodes from the joined web tuples. We denote this combined operation as a Π-web join. The Π-web join operator, denoted by the symbol $\overline{\bowtie}$, is a binary operator and is used to combine web tuples from two web tables containing relevant nodes. This operation eliminates irrelevant nodes from the joined web tuples created by concatenating web tuples of one web table with a web tuples of another table whenever there exists joinable nodes.

Formally, let W_i and W_j be two web tables with schemas $S_i = \langle X_{n_i}, X_{\ell_i}, C_i, P_i\rangle$ and $S_j = \langle X_{n_j}, X_{\ell_j}, C_j, P_j\rangle$ respectively. Let N be the set of joinable node variables in W_i and W_j such that $N \in X_{n_i} \cup X_{n_j}$ and P_c be the set of project conditions to eliminate irrelevant nodes from the joined web tuples. Furthermore, let $N_p(i)$ be the set of node variables to be eliminated based on the condition $P_c(i) \in P_c \; \forall \; 0 < i \leq |P_c|$. Then, the Π-web join between the web tables W_i and W_j based on the project condition(s) P_c is expressed as : $W_i \overset{P_c}{\overline{\bowtie}} W_j$ where for each project condition $P_c(i) \in P_c$, $N_p(i) \cap N = \emptyset$. That is, the set of node variables to be eliminated from the joined web table should not include any one of the joinable node variables. Each web tuple in the Π-joined web table must retain the instances of joinable node variable(s) of one of the input schema. The justification of retaining the joinable nodes is to preserve the correlation between the resultant information captured from the two input web tables (each instance of z and d correlates to a particular disease in x or a) after Π-web join operation.

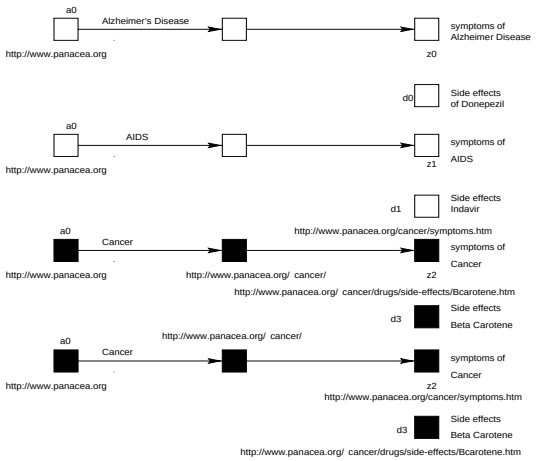

Figure 7. Π-joined web bag.

4.2 Construction

We first show how to determine if two web tables are Π-*joinable* and then we proceed to construct the Π-joined web table and its schema.

4.2.1 Checking Π-Web Joinability

Given the web schemas $S_i = \langle X_{n_i}, X_{\ell_i}, C_i, P_i\rangle$ and $S_j = \langle X_{n_j}, X_{\ell_j}, C_j, P_j\rangle$ of input web tables W_i and W_j, and a set of project condition(s) P_c, the output of this procedure is the decision whether the two web tables are Π-joinable. If they are, then we obtain a set of joinable node variables N_i and N_j where $N_i \in X_{n_i}$ and $N_j \in X_{n_j}$. Otherwise, the web tables cannot be Π-joined. We first identify whether the two web tables are joinable by inspecting the two input web tables. In case they are joinable, we identify the set of joinable node variables (node variables participating in the web join). Once the joinable node variables are determined, we check whether the given web tables are Π-joinable based on the project condition(s) P_c. If the set of node variables to be eliminated from the joined schema includes any one of the joinable node variables then the web tables W_i and W_j are not Π-joinable for the given project conditions P_c. Otherwise, it is possible to perform a Π-web join on W_i and

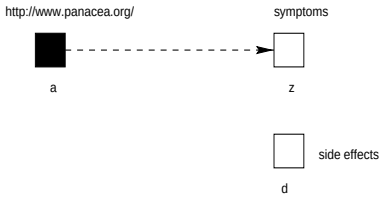

Figure 8. Web schema of Π-joined web table.

W_j.

Example 6 Consider the web schemas of Diseases and Drugs in Example 2 and 3. The Π-web joinability of these two web tables are determined as follows: We evaluate the web schemas of Diseases and Drugs to determine if these two web tables are joinable. Since we can identify joinable node variables from these web schemas, the web tables are joinable. The joinable node variables are $N_i = \{x\}$ and $N_j = \{a\}$. The node variables to be eliminated are $N_p = \{b, k, q, p\}$. Since, the set of node variables to be eliminated from the joined web schema does not include any one of the joinable node variables, the web tables are Π-joinable. ∎

4.2.2 Construction of Π-Web Joined Schema

First, we construct the joined schema of the web table W_i and W_j. The joined schema is constructed by integrating the two input web schemas and removing one of the joinable node variables from the integrated schema. The formal construction details of the joined schema is given in [8, 6]. Once the joined schema is constructed, we proceed to create the Π-joined web schema. We modify the four components of the schema based on the project conditions. We first identify the set of node and link variables to be eliminated from the schema of the joined web table W_{wj} based on the specified project conditions and delete these variables from $X_{n_{wj}}$ and $X_{\ell_{wj}}$. Then, it identifies the set of connectivities and predicate set involving the nodes and link variables to be deleted from the joined schema and remove them from C_{wj} and P_{wj}. Note that it is not necessary that all the node and link variables to be removed are bound. The schema of the Π-joined web table or web bag is then created by the resultant node and link variables and their associated connectivities and predicates. Figure 8 depicts the Π-joined schema based on project conditions in Example 1.

Example 7 Consider the project conditions in Example 1. The schema of the Π-joined web table is shown below (Figure 8). Let the Π-joined schema be $S_{pj} = \langle X_{n_{pj}}, X_{\ell_{pj}}, C_{pj}, P_{pj} \rangle$ where $X_{n_{pj}} = \{a, z, d\}$, $X_{\ell_{pj}} = \{\text{-}\}$, $C_{pj} \equiv k_1$ such that $k_1 = a\langle\text{-}^{+}\rangle z$, and $P_{pj} \equiv p_1 \wedge p_2 \wedge p_3$ such that $p_1(a) \equiv$ [a.`url EQUALS "http://www.panacea.org/"`]$p_1(d) \equiv$ [d.`title CONTAINS "side effects"`], $p_2(z) \equiv$ [z.`title CONTAINS "symptoms"`]. ∎

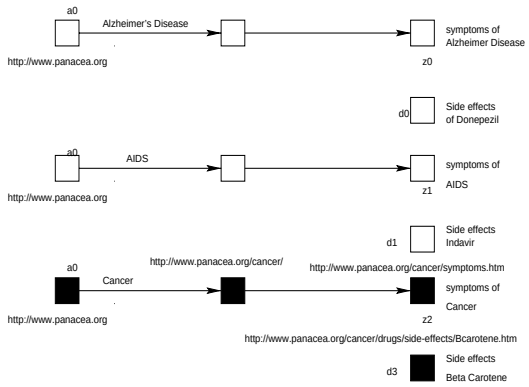

Figure 9. Π-joined web table (after duplicate elimination)..

4.2.3 Construction of Π-Web Joined Table

We now proceed to create the Π-joined web table. We construct the joined web table first by concatenating the input web tuples over the instances of the joinable node variables. Next, we delete some of the nodes and links from each joined web tuple based on the project condition(s) P_c using the web project operation. Then, we compare each web tuple with the other to determine the existence of duplicate tuples. If identical tuples are located, then the collection of web tuples is a web bag. For example, Figure 7 represents a web bag based on the project conditions in Example 1. We may eliminate the duplicate web tuples and store only distinct web tuples. Figure 9 depicts Π-joined web table after duplicate elimination. Otherwise, the set of tuples after web project operation results in Π-joined web table with distinct web tuples.

5 Benefits

By eliminating irrelevant information, Π-web join helps to reduce the cognitive overhead and storage cost associated with a joined web table. It also helps in minimizing the amount of data transmitted over the network in *distributed* web join processing. Recall that a web tuple is a directed graph containing nodes (web documents) and links between the nodes. However, not all nodes in a web tuple may be relevant to the user. The predicates on the link and node variables as defined in the web schema expedite the process of locating relevant nodes in a web tuple since it enables us

to speculate with reasonable accuracy the actual content of nodes and links. Although it is possible to identify relevant nodes efficiently in a web tuple containing few nodes, in general, locating relevant web documents in a web tuple containing large number of nodes may be frustrating and tedious. This situation may occur in a concatenated web tuple in a joined web table where there is a high possibility of existence of large number of nodes (since it combines nodes from two web tuples into a single joined web tuple).

As large number of unbound nodes and links may exist in a web tuple, the task of locating relevant nodes become significantly harder. Note that, it may be difficult to surmise the actual content of an unbound node or link as the are not defined by predicates in the web schema. Thus, the existence of unbound nodes and links aggravate the problem of identifying relevant nodes in a joined web tuple. Π-web join helps to eliminate these unbound node and links and reduce cognitive overhead associated with the joined web table.

The web schema beyond its use to define the structure of the web data in a web table, also serves two important purposes: First, it enables us to understand the structure of the web table and form meaningful queries over it. Second, a web query processor relies on the schema to devise efficient plans for computing query results. Without a well defined schema, both these tasks become significantly harder. Although it may be possible to manually browse a small web table, in general forming a meaningful query is difficult without a schema or some kind of structural summary of the underlying web table. Further, a lack of information about the structure of a web table can cause a query processor to resort to exhaustive searches. Π-web join may resolve these problems by removing unbound node and links to improve the structural information of the joined schema for subsequent query processing.

6 Summary

In this paper, we have introduced the concept of Π-web join in a web warehouse and discussed the construction of Π-joined web table and its schema. Our paper focus on how Π-web join may be used to resolve the shortcoming of web join operation in WHOWEDA. Currently, we have implemented the Π-web join operator and have interfaced it with other web operators. Due to space limitations, we have not reported the analysis of above mentioned benefits in detail here. Please refer to [6] for further details.

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 ICDE 98*, Orlando, Florida, February 1998.

[4] 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.

[5] S. BHOWMICK, S. K. MADRIA, W.-K. NG, E.-P. LIM. Web Warehousing: Design and Issues. *Proceedings of International Workshop on Data Warehousing and Data Mining (DWDM'98) (in conjunction with ER'98)*, Singapore, 1998.

[6] S. BHOWMICK, S. K. MADRIA, W.-K. NG, E.-P. LIM. Π-Web Join in A Web Warehouse: Design and Evaluation. *Technical Report*,CAIS-TR-98-21, Center for Advanced Information Systems, Nanyang Technological University, Singapore, 1998.

[7] S. BHOWMICK, W.-K. NG, E.-P. LIM. Information Coupling in Web Databases. *Proceedings of the 17th International Conference on Conceptual Modeling (ER'98)*, Singapore, 1998.

[8] 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.

[9] 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.

[10] M. FERNANDEZ, D. FLORESCU, A. LEVY, D. SUCIU. A Query Language and Processor for a Web-Site Management Systems. *Proceedings of the Workshop of Semi-structured Data*, Tuscon, Arizona, May 1997.

[11] 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.

[12] 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.

[13] 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.

[14] 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.