

Singapore Management University Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

11-1998

Information coupling in web databases

Sourav S. BHOWMICK


Wee-Keong NG

Ee Peng LIM

Singapore Management University, eplim@smu.edu.sg

DOI: https://doi.org/10.1007/978-3-540-49524-6_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

BHOWMICK, Sourav S.; NG, Wee-Keong; and LIM, Ee Peng. Information coupling in web databases. (1998). *Conceptual Modeling - ER '98: 17th International Conference on Conceptual Modelling, Singapore, 16-19 November 1998: Proceedings*. 1507, 92-106. Research Collection School Of Information Systems.

Available at: https://ink.library.smu.edu.sg/sis_research/971

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.

Information Coupling in Web Databases^{*}

SOURAV S. BHOWMICK WEE-KEONG NG EE-PENG LIM

Center for Advanced Information Systems, School of Applied Science,
Nanyang Technological University, Singapore 639798, SINGAPORE
{sourav,wkn,aseplim}@cais.ntu.edu.sg

Abstract

Web information coupling refers to an association of topically related web documents. This coupling is initiated explicitly by a user in a web warehouse specially designed for web information. Web information coupling provides the means to derive additional, useful information from the WWW. In this paper, we discuss and show how two web operators, i.e., *global web coupling* and *local web coupling*, are used to associate related web information from the WWW and also from multiple *web tables* in a web warehouse. This paper discusses various issues in web coupling such as coupling semantics, coupling-compability, and coupling evaluation.

1 Introduction

Given the high rate of growth of the volume of data available on the WWW, locating information of interest in such an anarchic setting becomes a more difficult task everyday. Thus, there is the recognition of the undeferring need for effective and efficient tools for information consumers, who must be able to easily locate and manipulate information in the Web. 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:

- While web browsers fully exploit hyperlinks among web pages, 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.
- From the query's result returned by 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

^{*} 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.

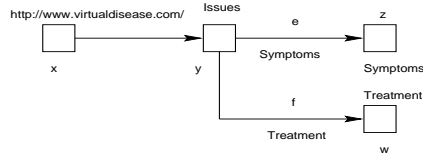


Fig. 1. Coupling framework (query graph) of 'Symptoms'.

files on the user's hard disk. However, this method is tedious, and it does not allow the user to retain the *coupling framework*.

- The set of downloaded documents can be refreshed (or updated) only by repeating the above procedure manually.
- If a user 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 organize, manage and infer further useful information from them.

In this paper, we introduce the concept of *Web Information Coupling* (WIC) to help overcome the limitations of present search engines. WIC enables us to efficiently manage and manipulate coupled information extracted from the Web. We use coupling because it is a convenient way to relate information located separately on the WWW. In this paper, we discuss two types of coupling; *global* and *local* web coupling.

Global coupling enables a user to retrieve a set of collections of inter-related documents satisfying a coupling framework regardless of the locations of the documents in the Web. To initiate global coupling, a user specifies the coupling framework in the form of a *query graph*. The actual coupling is performed by the WIC system and is transparent to the user. The result of such user-driven coupling is a set of related documents materialized in the form of a *web table*. Thus, global web coupling eliminates the problem of manually visiting and downloading Web documents as files in user's hard disk.

Coupling is not limited to the extraction of related information directly from the WWW. *Local* coupling can be performed on *web tables* [15] materialized by global coupling. This form of web coupling is achieved locally without resorting to the WWW. Given two web tables, local coupling is initiated explicitly by specifying a pair(s) of web documents 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. The following example briefly illustrates global and local web coupling.

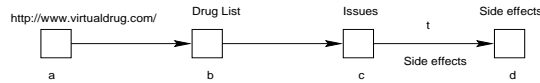


Fig. 2. Coupling framework (query graph) of ‘Drug list’.

Example 1. Suppose Bill wish to find a list of diseases with their symptoms and treatments, and a list of drugs and their side effects on diseases on the WWW. Assume that there are web sites at <http://www.virtualdisease.com/> and <http://www.virtualdrug.com/> which integrate disease and drug related information from various web sites respectively. Bill figured that there could be hyperlinks with anchor labels ‘symptoms’ and ‘treatments’ in the web site at <http://www.virtualdisease.com/> and labels ‘side effects’ in the web site at <http://www.virtualdrug.com/> that might be useful.

In order to initiate global web coupling (i.e., to couple these related information from the WWW), Bill constructs coupling frameworks (query graphs) as shown in Figures 1 and 2.

The *global web coupling operator* is applied to retrieve those set of related documents that match the coupling frameworks. Each set of inter-linked documents retrieved for each coupling framework is a connected, directed graph (also called *web tuples*) and is materialized in web tables `Symptoms` and `Drug_list` respectively. A small portion of these web tables is shown in Figures 3 and 4. Each web tuple in `Symptoms` and `Drug_list` contains information about the symptoms and treatments of a particular disease, and the side effects of a drug on the disease respectively.

Suppose a user want to extract information related to the symptoms and treatments of cancer and AIDS, and a list of drugs with their side effects on them. Clearly, these information are already stored in tables `Symptoms` and `Drug_list`.

The *local web coupling operator* enables us to extract these related information from the two web tables. A user may indicate the documents (say y and b) in the coupling frameworks of `Symptoms` and `Drug_list` and the keywords (in this case “cancer” and “AIDS”) based on which local web coupling will be performed. A portion of the coupled web table is shown in Figure 5. ■

A Web Information Coupling (WIC) system is a database system for managing and manipulating coupled information extracted from the Web. To realize this system, we first propose a data model called the *Web Information Coupling Model* (WICM) to describe and abstract web objects. We then introduce the operators to perform global and local coupling.

2 Web Information Coupling Model

We proposed a data model for a web warehouse in [5, 15]. The data model consists of a hierarchy of web objects. The fundamental objects are *Nodes* and *Links*.

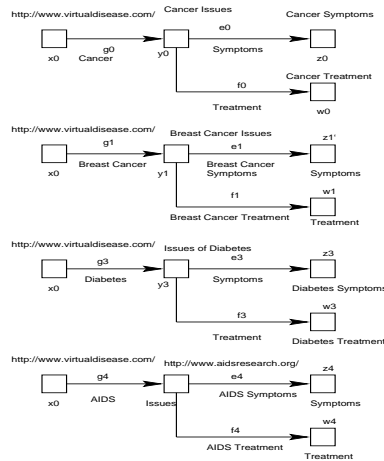


Fig. 3. Partial view of 'Symptoms' web table.

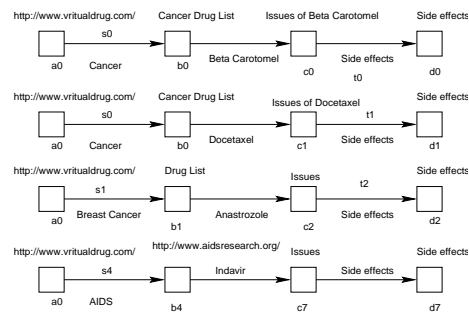


Fig. 4. Partial view of 'Drug list' web table.

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]

WICM supports structured or topological querying; different sets of keywords may be specified on the nodes and additional criteria may be defined for the hyperlinks among the nodes. Thus, the query is a graph-like structure and is used to match portions of the WWW satisfying the conditions. In this way, the query result is a set of directed graphs (called *web tuples*) instantiating the query graph. Formally, a web tuple $w = \langle N_w, L_w, V_w \rangle$, is a triplet where N_w is

a set of nodes in web tuple w , L_w is a set of links in web tuple w and V_w is the set of *connectivities* (next section). A collection of these web tuples is called a *web table*. If the web table is materialized, we associate a *name* with the table. defined as 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 DNF, P is a set of predicates in DNF. The web schema of the web table is the query graph that is used to derive the table. It is defined as 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 DNF, P is a set of predicates in DNF. A set of web tables constitutes a *web database*.

We illustrate the concept of web schema with the following examples. Consider the query graphs (Figures 1 and 2) in Example 1. The schemas of these query graphs are given below:

Example 2. Produce a list of diseases with their symptoms and treatments, starting from the web site at <http://www.virtualdisease.com/>.

We may express the schema of the above query by $M_i = \langle X_{i,n}, X_{i,\ell}, C_i, P_i \rangle$ where $X_{i,n} = \{x, y, z, w\}$, $X_{i,\ell} = \{e, f, -\}$, $C_i \equiv k_{i_1} \wedge k_{i_2} \wedge k_{i_3}$ such that $k_{i_1} = x(-)y$, $k_{i_2} = y(e)z$, $k_{i_3} = y(f)w$, and $P_i \equiv p_{i_1} \wedge p_{i_2} \wedge p_{i_3} \wedge p_{i_4} \wedge p_{i_5} \wedge p_{i_6}$ such that $p_{i_1}(x) \equiv [x.url \text{ EQUALS } "http://www.virtualdisease.com/"]$, $p_{i_2}(y) \equiv [y.title \text{ CONTAINS } "issues"]$, $p_{i_3}(e) \equiv [e.label \text{ CONTAINS } "symptoms"]$, $p_{i_4}(z) \equiv [z.title \text{ CONTAINS } "symptoms"]$, $p_{i_5}(f) \equiv [f.label \text{ CONTAINS } "treatments"]$, $p_{i_6}(w) \equiv [w.title \text{ CONTAINS } "treatments"]$. ■

Example 3. Produce a list of drugs and their side effects starting from the web site at <http://www.virtualdrug.com/>.

The schema of the above query is $M_j = \langle X_{j,n}, X_{j,\ell}, C_j, P_j \rangle$ where $X_{j,n} = \{a, b, c, d\}$, $X_{j,\ell} = \{t, -\}$, $C_j \equiv k_{j_1} \wedge k_{j_2} \wedge k_{j_3}$ such that $k_{j_1} = a(-)b$, $k_{j_2} = b(-)c$, $k_{j_3} = c(t)d$ 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 $p_{j_1}(a) \equiv [a.url \text{ EQUALS } "http://www.virtualdrug.com/"]$, $p_{j_2}(b) \equiv [b.title \text{ CONTAINS } "Drug List"]$, $p_{j_3}(c) \equiv [c.title \text{ CONTAINS } "Issues"]$, $p_{j_4}(d) \equiv [d.title \text{ CONTAINS } "side effects"]$, $p_{j_5}(t) \equiv [t.label \text{ CONTAINS } "side effects"]$. ■

The query graphs (web schemas) as described in Examples 2 and 3 express Bill's need to extract a set of inter-linked documents related to symptoms and treatments of diseases, and the side effects of drugs on these diseases from the WWW. Since conventional search engines cannot accept a query graph as input and return the inter-linked documents as the query result, a global web coupling operator is required. The global web coupling operator matches those portions of the WWW that satisfy the query graphs. The results of global web coupling is a collection of sets of related Web documents materialized in the form of a web table.

Although global web coupling retrieves data directly from the WWW, the full potential of web coupling lies in the fact that it can couple related information residing in two different web tables in a web database. Suppose Bill wish to know the symptoms and treatments associated with cancer and AIDS, and a list of drugs with their side effects on them. There are two methods in a web database to gather the composite information:

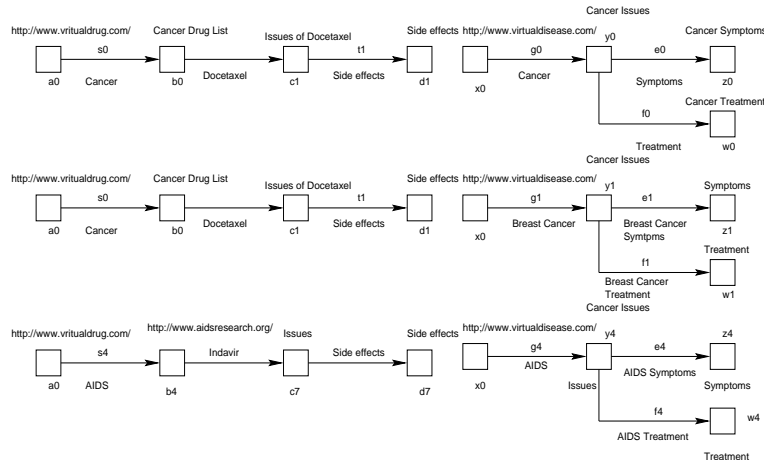


Fig. 5. Web coupling.

1. Bill may construct a new web query for this purpose. The disadvantage of this method is that the information (stored in web tables) created by the queries in Examples 2 and 3 are not being used for this query.
2. Browse the web tables of queries in Examples 2 and 3 and select those tuples containing information related to cancer and AIDS and then compare the results manually. However, there may be many matching web tuples, making the user's task of going over them tedious.

This motivates us to design a local web coupling operator that allows us to gather related information from the two web tables in a web database.

3 Global Web Coupling

In this section, we discuss global web coupling; a mechanism to couple related information from the WWW. We begin by formally defining the global web coupling operator. Next we explain how a coupled web table is created.

3.1 Definition

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.

Some computability issues arise when applying the global web coupling operator to WWW. The global web coupling operator, is *bound* if and only if all

variables that begin a connectivity in the schema specified for the operator are bound. A query which embeds a bound Γ operator is always computable. Let us see why. Suppose a web query with schema M is posed against the WWW, i.e., we wish to compute $\Gamma(M)$. Intuitively, the Γ operator is evaluable when there are starting points in the WWW from which we can begin our search. With current web technology, there are two methods to locate a web resource; we either know its URL and access the resource directly or we go through a search engine by supplying keywords to obtain the URL's.

Let x be a node variable, then a predicate such as `[x.url EQUALS "a-url-here"]` in a query allows us to use the URL specified to locate the document corresponding to x . The second method is embedded by predicates such as `[x.text CONTAINS "some-keywords"]`, `[x.title EQUALS "a-title-here"]`, and `[e.label CONTAINS "some-keywords"]`. Here, x and e are the bound variables. When a node or link variable is bound, we can access the resource it corresponds to either directly or through a web search engine. Variables that begin connectivities and are bound provide the starting point in the WWW for retrieving web tuples. Hence, queries with such variables are computable.

3.2 Web Table Creation

We now discuss briefly how to create the coupled web table. Given a web schema (query graph), Γ extracts a set of web tuples satisfying the query graph. We discuss how to extract the set of web tuples from the WWW. Our approach to determine the set of web tuples from the WWW is as follows:

1. Check if the given web schema is computable. If it is, then obtain a set of URL(s) as the starting point of traversal by analyzing the predicates in the schema.
2. Get the node variables representing these start URL(s) and identify connectivities which contain the start node variables. Note that the start node variable will always be in the left hand side of a connectivity.
3. Download the documents from the WWW that satisfy the predicates for the nodes and that contain links that satisfy the link predicates for the outgoing edges of this node.
4. Get the web documents (nodes) pointed by the links and check whether these documents satisfy the predicates of the node in the schema. Repeat this until we reach the right hand side of the connectivity.
5. Repeat the above two steps for all the connectivities in the schema.
6. Once all the web documents are collected by the above procedure, create individual web tuples by matching the set of nodes and links with the schema.

4 Local Web Coupling

Once we have the ability to couple useful information directly from the WWW using the global web coupling operator, we need to introduce an additional operator to facilitate local web coupling, i.e., extracting useful information locally from two web tables.

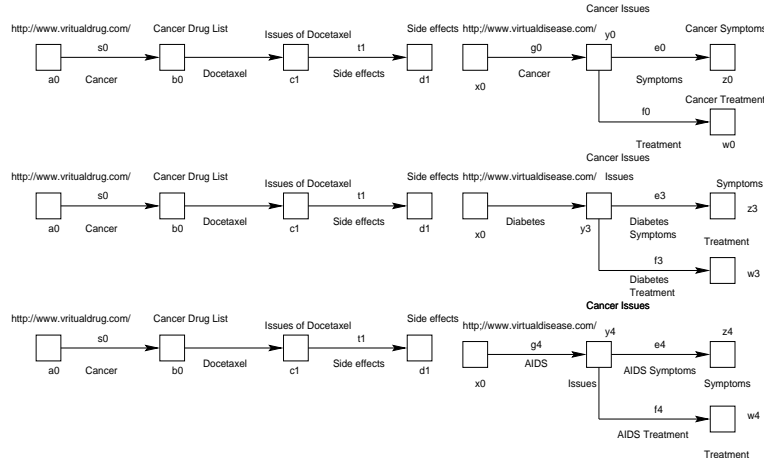


Fig. 6. Web cartesian product.

4.1 Definition

The local web coupling operator combines two web tables by *integrating* web tuples of one web table with web tuples of another table whenever there exists *coupling nodes*. Let W_i and W_j be two web tables with schemas $M_i = \langle X_{i,n}, X_{i,\ell}, C_i, P_i \rangle$ and $M_j = \langle X_{j,n}, X_{j,\ell}, C_j, P_j \rangle$ respectively. Suppose we want to couple W_i and W_j on node variables n_{c_i} and n_{c_j} as they both contain information about diseases, and we want to correlate web tuples of W_i and W_j related to cancer. Let w_i and w_j be two web tuples from W_i and W_j respectively, and $n_c(w_i)$ and $n_c(w_j)$ be instances of 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 $n_c(w_j)$) respectively contain information related to cancer and appear 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). Thus, coupling nodes are $n_c(w_i)$ and $n_c(w_j)$. We store the coupled web tuple in a separate web table. Note that coupling-compatibility of two web tuples depends on the pair(s) of node variables and keyword(s) specified explicitly by the user in the local coupling query. We now formally define coupling-compatibility.

Definition 1. Let $K(n, w, W)$ denote a set of keywords appearing in a web document (represented by node n) in web tuple w of web table W . Two web tuples w_i and w_j of web tables W_i and W_j are *coupling-compatible locally* on the node pair $(n_c(w_i), n_c(w_j))$ based on some keyword set K_c if and only if the following conditions are true: $n_c(w_i) \in N_{w_i}$, $n_c(w_j) \in N_{w_j}$, $K_c \subseteq K(n_c(w_i), w_i, W_i)$ and $K_c \subseteq K(n_c(w_j), w_j, W_j)$.

The new web tuple w derived from the coupling of w_i and w_j is defined as:
 $N_w = N_{w_i} \cup N_{w_j}$, $L_w = L_{w_i} \cup L_{w_j}$ and $V_w = V_{w_i} \cup V_{w_j}$. ■

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 created by the coupling operation satisfying 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 from 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. Note that in order to couple the two web tables, the keyword(s) should be present in at least one instance of the coupling node variable pair. Furthermore, there may be more than one pair of coupling node variables on which local web coupling can be performed.

Local web coupling is a combination of two web operations: a web cartesian product followed by a web select based on some selection condition on the coupling nodes. Like its relational counterpart, 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 have n and m web tuples respectively, then the resulting web cartesian product has $n \times m$ web tuples. The schema of the resultant web table W' is given as $M' = \langle X_n', X_\ell', C', P' \rangle$ where $X_n' = X_{i,n} \uplus X_{j,n}$, $X_\ell' = X_{i,\ell} \uplus X_{j,\ell}$, $C' = C_i \uplus C_j$ and $P' = P_i \uplus P_j$. The symbol \uplus refers to the *disambiguation* [5, 15] of nodes and link variables. Let us now illustrate web cartesian product with an example.

Example 4. Consider the web tables `Symptoms` and `Drug-list` in Figures 3 and 4. The web cartesian product of these two web tables is shown in Figure 6. Due to space limitations, we only show a small portion of the resultant web table. The schema of the resultant web table is $M' = \{X_n', X_\ell', C', P'\}$ where $X_n' = X_{i,n} \uplus X_{j,n} = \{x, y, z, w, a, b, c, d\}$, $X_\ell' = X_{i,\ell} \uplus X_{j,\ell} = \{t, e, f, -\}$, $C' = C_i \uplus C_j \equiv k_1' \wedge k_2' \wedge k_3' \wedge k_4' \wedge k_5' \wedge k_6'$ such that $k_1' = x(-)y$, $k_2' = y(e)z$, $k_3' = y(f)w$, $k_4' = a(-)b$, $k_5' = b(-)c$, $k_6' = c(t)d$, and $P' = P_i \uplus P_j \equiv p_1' \wedge p_2' \wedge p_3' \wedge p_4' \wedge p_5' \wedge p_6' \wedge p_7' \wedge p_8' \wedge p_9' \wedge p_{10}' \wedge p_{11}'$ such that $p_1'(x) \equiv [x.url \text{ EQUALS } "http://www.virtualdisease.com/"]$, $p_2'(y) \equiv [y.title \text{ CONTAINS } "issues"]$, $p_3'(e) \equiv [e.label \text{ CONTAINS } "symptoms"]$, $p_4'(z) \equiv [z.title \text{ CONTAINS } "symptoms"]$, $p_5'(f) \equiv [f.label \text{ CONTAINS } "treatments"]$, $p_6'(w) \equiv [w.title \text{ CONTAINS } "treatments"]$, $p_7'(a) \equiv [a.url \text{ EQUALS } "http://www.virtualdrug.com/"]$, $p_8'(b) \equiv [b.title \text{ CONTAINS } "Drug List"]$, $p_9'(c) \equiv [c.title \text{ CONTAINS } "Issues"]$, $p_{10}'(d) \equiv [d.title \text{ CONTAINS } "side effects"]$, $p_{11}'(t) \equiv [t.label \text{ CONTAINS } "side effects"]$. ■

A web select operation is performed after web cartesian product to filter out web tuples where the specified nodes cannot be related based on the keyword(s) conditions. These conditions impose additional constraints on the node variables participating in local web coupling. We denote this sequence of operations as local web coupling and we can replace the two operations

$$W' = W_i \times W_j$$

$$W = \sigma_{(\langle \text{node_pair} \rangle, \langle \text{keyword_condition}(s) \rangle)}(W')$$

with $W = W_i \otimes_{(\langle \text{node_pair} \rangle, \langle \text{keyword}(s) \rangle)} W_j$. The symbol σ denotes web selection. The result of a local web coupling operation is a web table having one web tuple for each combination of web tuple—one from W_i and one from W_j —whenever there exist coupling nodes. Let us illustrate web coupling with an example.

Example 5. Consider the web tables `Symptoms` and `Drug_list` as depicted in Examples 2 and 3. Suppose Bill wish to find symptoms, treatments details of “Cancer” and “AIDS” and the list of drugs with their side effects on these diseases. The coupled web table is shown in Figure 5. Note that the third and fourth web tuples in Figure 6 are excluded in the coupled web table since they do not satisfy the keyword conditions. The schema of the coupled web table is $M = \langle X_n, X_\ell, C, P \rangle$ where $X_n = X_n'$, $X_\ell = X_\ell'$, $C = C_i'$ and $P = P'$. The construction details of the coupled schema and the coupled web table will be explained in Section 4.3. ■

4.2 Terminology

We introduce some terms we shall be using to explain local web coupling in this paper.

- **Coupling nodes:** Two web tuples w_i and w_j of web tables W_i and W_j respectively can be coupled if there exist at least one node $n_c(w_i)$ and $n_c(w_j)$ in w_i and w_j which can be coupled with the other based on *similar information content*. We refer to these nodes as **coupling nodes**. We express the coupling nodes of w_i and w_j as coupling pairs since they cannot exist as a single node. Formally, $(n_c(w_i), n_c(w_j))$ is a coupling pair where node $n_c(w_i)$ is coupled with $n_c(w_j)$ of w_j . The attributes of $n_c(w_i)$ and $n_c(w_j)$ are called **coupling attributes**. For example, the coupling nodes of the first web tuple in Figures 3 and 4 are y_0 and b_0 respectively. The coupling pair for these nodes is (y_0, b_0) , The coupling attributes of y_0 and b_0 are text, title etc.
- **Coupling-activating links:** All the incoming links to the coupling nodes $n_c(w_i)$ and $n_c(w_j)$ are called **coupling-activating links**. Formally, $\ell_{n_c(w_i)}$ is the coupling-activating link of the coupling node $n_c(w_i)$. For example, the link g_0 in Figure 3 is the coupling-activating link of node y_0 .
- **Coupling keywords:** The keyword condition(s) specified by the user based on which coupling between node variables are performed are called **coupling keywords**.

4.3 Web Table Creation

We now discuss the process of deriving the coupled web table from two input web tables. Given two web tables, a set of coupling keyword(s), and pair(s) of

node variables, we first construct the schema of the coupled web table and then proceed to create the table itself. Let web tables W_i and W_j with schemas M_i and M_j be participating in the local web coupling process. Let the coupled web table be W with schema $M = \langle X_n, X_\ell, C, P \rangle$.

Construction of the coupled schema We now determine the four components of M in the following steps:

Step 1: Determine the Node set: Node variables in X_{n_i} and X_{n_j} can either be nominally distinct from one another or there may exist at least one pair of node variables from X_{n_i} and X_{n_j} which are identical to one another. If the node variables are not nominally distinct, we disambiguate one of the identical node variable(s). The node set of the coupled schema is given as: $X_n = (X_{n_i} \uplus X_{n_j})$.

Step 2: Determine the Link set: Similarly, we disambiguate the identical link variables in X_{ℓ_i} and X_{ℓ_j} if necessary, and the link set of the coupled schema is given as: $X_\ell = X_{\ell_i} \uplus X_{\ell_j}$.

Step 3: Determine the Connectivity set: If the node and link variables are not nominally distinct, we replace any one of the identical variables in C_i or C_j with the disambiguated value. The connectivity set of the coupled schema C is given as: $C = C_i \uplus C_j$.

Step 4: Determine the Predicate set: Our approach to determine the predicate set of the coupled schema is similar as above. If the node and link variables are not nominally distinct we replace any one of the identical node variables in P_i or P_j with the disambiguated value. The predicate set of the coupled schema is given as: $P = P_i \uplus P_j$.

Construction of the coupled web table The coupled web table is created by *integrating* the two input web tables. We describe the steps below:

Step 1: Given two web tables, we first perform a web cartesian product on the two web tables.

Step 2: For each web tuple in the web table created by web cartesian product, the specified nodes are inspected to determine whether the web tuple is coupling-compatible locally (based on the coupling keyword(s) provided by the user). In order to be coupling-compatible, the specified pair of nodes in the web tuple must satisfy some *coupling-compatibility conditions*. We determine these conditions in the next section. We inspect each web tuples in the web table created by web cartesian product to determine if the specified pair(s) of node satisfy any one of the coupling compatibility conditions.

Step 3: If a pair of nodes satisfy none of the conditions, the corresponding web tuple is rejected. If the nodes satisfy at least one of the above conditions, the web tuple is stored in a separate web table (coupled web table).

Node	URL	Title	Text
y_0	http://www.virtualdisease.com/cancerindex.html	Cancer Issues	Cancer
b_0	http://www.virtualdrug.com/cancer.html	Cancer Drug List	Cancer

Table 1. Node attributes of y and b .

Link	From Node	To Node	Label	Link Type
g_0	x_0	y_0	Cancer	local
s_0	a_0	b_0	Cancer	local

Table 2. Link attributes of g and s .

Step 4: Repeat steps 2 and 3 for other web tuples in the resultant web table created by web cartesian product.

4.4 Coupling-Compability Conditions

Local web coupling-compability conditions may be based on node attributes of the instances of specified node variables and/or attributes of the instances of incoming link variables of the specified node variables (coupling-activating links). Let us define some terms to facilitate our exposition.

Given a web tuple w of web table W with schema $M = \langle X_n, X_\ell, C, P \rangle$, let $n(w)$ be a node of w and $\ell_{n(w)}$ be incoming link to node $n(w)$ such that:

- $\text{attr}(n(w)) \in \{\text{url}, \text{text}, \text{title}, \text{format}, \text{date}, \text{size}\}$ is a node attribute;
- $\text{attr}(\ell_{n(w)}) \in \{\text{source_url}, \text{target_url}, \text{label}, \text{link_type}\}$ is a link attribute and
- $\text{val}(n(w))$ and $\text{val}(\ell_{n(w)})$ are the values of $\text{attr}(n(w))$ and $\text{attr}(\ell_{n(w)})$ respectively.

For example, consider Tables 1 and 2 which depict some of the attributes of node variables b, y and link variables g, s . For node b_0 and $\text{attr}(b_0) = \text{title}$, and $\text{val}(b_0) = \text{Cancer Drug List}$. For link s_0 (incoming link to node b_0), with $\text{attr}(s_0) = \text{label}$ and $\text{val}(s_0) = \text{Cancer}$.

Let n_{c_i} and n_{c_j} be node variables in schemas M_i and M_j of web tables W_i and W_j respectively participating in the local web coupling and K_c be the coupling keywords. Let w_i and w_j be two web tuples of W_i and W_j such that $n_c(w_i)$ and $n_c(w_j)$ are instances of n_{c_i} and n_{c_j} respectively. Moreover, let the web cartesian product of W_i and W_j be W' and let w' be a web tuple in W' which is the cartesian product of w_i and w_j .

Web documents represented by nodes $n_c(w_i)$ and $n_c(w_j)$ can be coupling nodes (that is web tables W_i and W_j are coupling-compatible) if they satisfy at least one of the coupling-compatibility conditions given below:

1. title of the web documents is equal to K_c or contains the coupling keyword K_c , i.e., $\text{attr}(n_c(w_i)) = \text{attr}(n_c(w_j)) = \text{title}$, $\text{val}(n_c(w_i))$ and $\text{val}(n_c(w_j))$ is equal to K_c or contains K_c .

2. text of the web documents contains K_c , i.e., $\text{attr}(n_c(w_i)) = \text{attr}(n_c(w_j)) = \text{text}$, $\text{val}(n_c(w_i))$ and $\text{val}(n_c(w_j))$ contains K_c .
3. The coupling keyword K_c is contained in the text of one web document and in the title of the other document, i.e., $\text{attr}(n_c(w_i)) = \text{text}$, $\text{attr}(n_c(w_j)) = \text{title}$, $\text{val}(n_c(w_i))$ is equal to or contains K_c and $\text{val}(n_c(w_j))$ contains K_c .
4. The coupling keyword is contained in the file name of URL of the web documents, i.e., $\text{attr}(n_c(w_i)) = \text{attr}(n_c(w_j)) = \text{url.filename}$, $\text{val}(n_c(w_i))$ and $\text{val}(n_c(w_j))$ contains K_c .
5. The coupling keyword is contained in the text of one web document and in the file name of the URL of other document, i.e., $\text{attr}(n_c(w_i)) = \text{text}$, $\text{attr}(n_c(w_j)) = \text{url.filename}$, $\text{val}(n_c(w_i))$ and $\text{val}(n_c(w_j))$ contains K_c .
6. The coupling keyword is contained in the file name of the URL of one document and in the title of the other document, i.e., $\text{attr}(n_c(w_i)) = \text{url.filename}$, $\text{attr}(n_c(w_j)) = \text{title}$, $\text{val}(n_c(w_i))$ contains K_c and $\text{val}(n_c(w_j))$ contains or is equal to K_c .
7. The label of the incoming links $\ell_{n_c(w_i)}$ and $\ell_{n_c(w_j)}$ to the web documents contains the coupling keyword K_c , i.e., $\text{attr}(\ell_{n_c(w_i)}) = \text{attr}(\ell_{n_c(w_j)}) = \text{label}$, $\text{val}(\ell_{n_c(w_i)})$ and $\text{val}(\ell_{n_c(w_j)})$ are equal to K_c or contains K_c .
8. The label of the incoming link $\ell_{n_c(w_i)}$ and the title of node $n_c(w_j)$ contains or are equal to K_c , i.e., $\text{attr}(\ell_{n_c(w_i)}) = \text{label}$, $\text{attr}(n_c(w_j)) = \text{title}$, $\text{val}(\ell_{n_c(w_i)})$ and $\text{val}(n_c(w_j))$ are equal to K_c or contains K_c .
9. The label of the incoming link to one document contains or is equal to K_c and the text of the other web document contains the coupling keyword, i.e., $\text{attr}(\ell_{n_c(w_i)}) = \text{label}$, $\text{attr}(n_c(w_j)) = \text{text}$, $\text{val}(\ell_{n_c(w_i)})$ is equal to or contains K_c and $\text{val}(n_c(w_j))$ contains K_c .
10. The label of the incoming link contains or is equal to K_c and the file name of the URL of the other web document contains K_c , i.e., $\text{attr}(\ell_{n_c(w_i)}) = \text{label}$, $\text{attr}(n_c(w_j)) = \text{url.filename}$, $\text{val}(\ell_{n_c(w_i)})$ is equal to or contains K_c and $\text{val}(n_c(w_j))$ contains K_c .

5 Related Work

We would like to briefly survey web data retrieval and manipulation systems proposed so far, and compare them with web information coupling. There has been considerable work in data model and query languages for the World Wide Web [9], [11], [12], [13]. To the best of our knowledge, we are not aware of any work which deals with web information coupling in web databases. 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 flattened immediately into linear tuples. This limits the expressiveness of queries to some extent as complex queries involving operators such as local web coupling are not possible. Konopnicki and Shmueli [11] proposed a high level querying system called 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.

Past query result are not used for the evaluation of future queries. This limit the usage of web operators like local web coupling to derive additional information from the past queries. Fiebig, Weiss and Moerkotte [9] extended relational algebra to the World Wide Web by augmenting the algebra with new domains (data types), and functions that apply to the domains. The extended model is known as RAW (Relational Algebra for the Web). Only two low level operators on relations, *scan* and *index-scan*, have been proposed to expand an URL address attribute in a relation and to rank results returned by web search engine(s) respectively. RAW made minor improvements on the existing relational model to accommodate and manipulate web data and there is no notion of a coupling operation similar to the one in WICM. Inspired by concepts in declarative logic, Lakshmanan, Sadri and Subramanian [12] designed WebLog to be a language for querying and restructuring web information. But there is no formal definition of web operations such as web coupling. Other proposals, namely Lorel [1] and UnQL [8], 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. Moreover, in both proposals there is no notion of web coupling operation similar to the one in WICM. Website restructuring systems like Araneus [4] and Strudel [10], exploit the knowledge of a website's structure to define alternative views over its content. Both these models do not focus on web information coupling similar to the one in WICM. 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, semistructured data and web site 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. But there is no notion of web coupling operation similar to ours.

6 Summary and Future Work

In this paper, we have motivated the need for coupling useful information residing in the WWW and in multiple web tables from a web database. We have introduced the notion of global web coupling and local web coupling that enable us to couple useful related information from the WWW and associate related information residing in different web tables by combining web tuples whenever they are coupling-compatible. We have shown how to construct the coupled web table globally and locally from the WWW and two input web tables respectively.

Presently, we have implemented the global web coupling operator and have interfaced it with other web operators. The current global web coupling operator can be used efficiently for simple web queries. We are in the process of implementing the local web coupling operator and finding ways to optimize web coupling.

References

1. 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.
2. S. ABITEBOUL, V. VIANU. Queries and Computation on the Web. *Proceedings of the 6th International Conference on Database Theory*, Greece, 1997.
3. G. AROCENA, A. MENDELZON WebOQL: Restructuring Documents, Databases and Webs. *Proceedings of ICDE 98*, Orlando, Florida, February 1998.
4. P. ATZENI, G. MECCA, P. MERIALDO Semistructured and Structured Data in the Web: Going Back and Forth. *Proceedings of Workshop on Semi-structured Data*, Tuscon, Arizona, May 1997.
5. S. S. BHOWMICK, W.-K. NG, E.-P. LIM. Join Processing in Web Databases. *Proceedings of 9th International Conference on Database and Expert Systems Applications (DEXA'98)*, Vienna, Austria, August 24-28, 1998.
6. S. S. BHOWMICK, S. K. MADRIA, W.-K. NG, E.-P. LIM. Web Bags: Are They Useful in A Web Warehouse? *Submitted for publication*.
7. S. S. BHOWMICK, S. K. MADRIA, W.-K. NG, E.-P. LIM. Semi Web Join in WICS. *Submitted for publication*.
8. 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.
9. 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.
10. M. FERNANDEZ, D. FLORESCU, A. LEVY, D. SUCIU A Query Language and Processor for a Web-Site Management Systems. *Proceedings of Workshop on 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, 1996.
14. W.-K. NG, E.-P. LIM, S. S. BHOWMICK, S. K. MADRIA An Overview of A Web Warehouse. *Submitted for publication*.
15. 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.