D_040

WebDBをコンポーネントとするセマンティック・メタ検索の提案

A proposal of the semantic metasearch with WebDBs as components

森 雅生 †　中藤 哲也 ‡　廣川 佐千男 ‡
Masao Mori　Tetsuya Nakatoh　Sachio Hirokawa

1 序文

本稿では、人手で行うウェブ検索の反復作業を自動化・統合化するアーキテクチャー personal semantic metasearch (以下 **PSM**) ーを提案する。一般の検索エンジンはキーワードを受け取り、それに応じた検索結果をURLのリストにして返すが、本稿で対象とするウェブデータベース（以下 **WebDB**）は、複数のフィールドごとにキーワードを指定する複雑なクエリを入力として、レコードのリストを検索結果として返す。ユーザが日常的に使っている WebDB 群の出力と入力とをどう結合させるか、また検索結果をどのように表示するかを記述したスクリプトにより PSM は実現される。本研究の特色として以下の3点である。

- WebDB と出力形式とがシステムの中でコンポーネントとして等価な対象として扱われること。
- 検索結果の出力の中の文字列をすぐに次の検索に使うことができる CGI リンクの埋め込み。

PSM はスクリプト言語を使って記述され、対応する CGI を逐次生成して実行することで実現され、自由度と汎用性が高い。

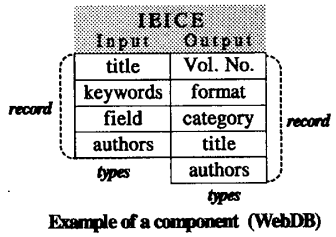

図1 WebDB コンポーネント

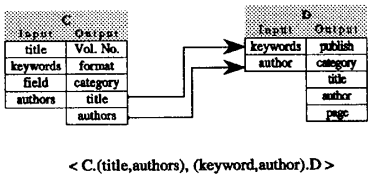

図2 コンポーネントの合成

2 WebDB コンポーネントとその結合

反復する検索の手作業を自動化することがこの研究の動機であった。そこで、WebDB 利用時の人手による検索作業を振り返ってみる。利用頻度の高い WebDB はユーザごとに定まっており、検索結果をブラウザに表示するたびに、その中から新しいキーワードを探してきて次の WebDB にキーワードとして与える。たとえば、マウスによるカット＆ペーストを使って検索作業を繰り返す。WebDB 群の合成は入出力チャンネルの結合によって実現されるが、実際の WebDB の結合はキーワードとなる文字列などの手動コピーの反復作業をしており、これを自動的に行うことが本研究の主たる目的の一つである。また、表示している検索結果から単純にクリックするだけで冗長な手動検索作業が回避できる CGI リンクと呼ばれるリンクを表示データに埋め込む手法も本研究の特徴である。結合をする場合、出力データのレコードの各タイプが何であるか理解しておく必要がある。この意味で、PSM は単なるメタ検索ではない。

PSM は、WebDB 群と通信する CGI プログラムを生成し、ユーザからの問い合わせをその CGI で実行することで実現される。この CGI は、WebDB コンポーネント自身の情報との結合情報が記述されたスクリプトから生成する。PSM が生成する CGI は3つの部品（コンポーネント情報、結合情報、自由定義の出力形式）から構成される。

2.1 コンポーネント

入出力を行う対象はチャンネルを持ったコンポーネントとして記述される。これらのチャンネルにはタイプが名付けられている。いくつかのタイプのインスタンスの集合をレコードと呼ぶ。一般にコンポーネントは部分レコードのリストを入出力する。ここで言う出力はブラウザへの出力ではなく、結合が定義された次のコンポーネントへの出力を指す。コンポーネントは3種類ある。

開始コンポーネント　出力チャンネルだけを持つコンポーネントで、ユーザからのクエリを出力する。

WebDB コンポーネント　対応する WebDB とその出力のラッパー関数を持ったコンポーネント（図1）。入力された値を WebDB へ質問し、結果の値をラッパー関数によって PSM 内の統一レコード形式に整形しにして出力する。

ユーザコンポーネント　ユーザへのインターフェースとなるコンポーネント。このコンポーネントに入力された値はブラウザを通して表示され、ユーザがブラウザを通して入力した値はこのコンポーネントの出力として他のコンポーネントに渡される。

2.2 コンポーネントの合成

C を出力チャンネル $o_1,\ldots,o_p$ を持つコンポーネント、D を入力チャンネル $i_1,\ldots,i_q$ を持つコンポーネントとする。次のチャンネルの対を C から D への結合と呼び、これによりコンポーネントの合成を定義する（図2）。

$$\langle C.(o_1,\ldots,o_p),(i_1,\ldots,i_q).D\rangle$$

ただし、チャンネルは $o_{p_m} \rightarrow i_{q_m} (m = 1,\cdots,k)$ で対応しており、この対応でデータをパイプする。対応するチャンネルの要素がひとつの場合、$\langle C.o,i.D\rangle$ のように略する。

このようにコンポーネントを節にして結合を枝にすると有向グラフが得られる。これを結合グラフと呼ぶ。結合グラフの中におけるサイクルは、そのサイクルが少なくともひとつ CGI リンクに対応する枝をもつときに許される。CGI リンクは以下の次節で紹介する。

† 九州大学大学院システム情報科学研究院, Faculty of Information Science and Electrical Engineering, Kyushu University
‡ 九州大学情報基盤センター, Computing and Communications Center, Kyushu University

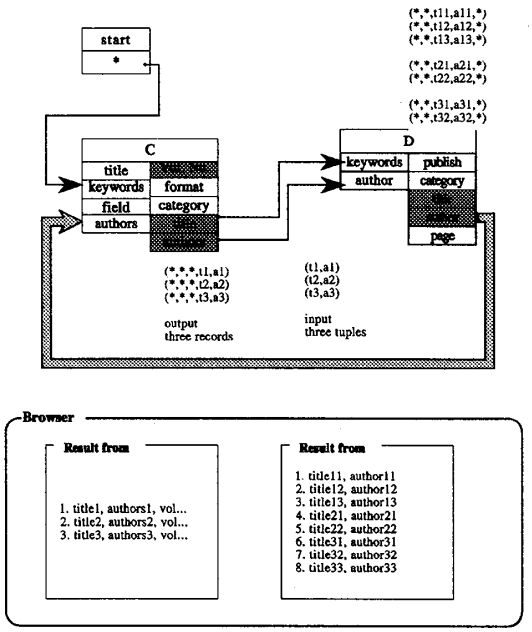

図 3　結合グラフ

3　自由定義の出力形式

検索結果の出力にユーザ定義が可能になるように自由度を持たせるため、3 つの主要な機能を提案する。

3.1　表示チャンネル

結合グラフが与えられたとき、ブラウザに表示する出力チャンネルの集合を表示チャンネル群と呼ぶ。図 2 の結合グラフの例では $C.(volno, title, authors)$ と $(title, author).D$ が指定されており、下のブラウザには出力されたレコードから指定されたチャンネルのタイプだけを抽出したリストが表示されている。

3.2　CGI リンクの埋め込み

図 3 の枝 $\langle C.author, authors.D \rangle$ で、枝のソースであるチャンネルがディスプレイチャンネル群の中に含まれている。このような枝を **CGI** リンクと呼び、ブラウザでは対応する文字列が次の検索のクエリになるように CGI の URL と文字列が埋め込まれて表示されている。例えば図 3 のコンポーネント C からの第 3 番の出力 '3.title13, author13' は HTML データでは次のように記述されることになる。

3. title13, `<a href="http://..../**.cgi">` author13 `</a>`.

ユーザはこのリンクをクリックするとストレスなく次の検索を行うことが出来る。

3.3　検索結果出力のための基本フィルタ

PSM でやり取りされるレコードは統一されているので、単純に検索結果をリスト表示するだけでなく、簡単な統計情報を抽出することができる。図 4 の例は WebDB コンポーネントに学術論文のデータベースを取り上げ、ひとつのキーワードに対し複数の WebDB コンポーネントからの結果をリストしているものである。ここで、下の表は上のリストから共著者を数え上げて作ったヒストグラムである。まったく同じ機能を単純なキーワード検索の結果に適用し適切な並べ替えを行うと、そのキーワードで論文を執筆している研究者と論文本数の表が得られる。

4　結論

ユーザの好みに応じて出力形式や WebDB の結合を定義できるセマンティック・メタ検索 (PSM) を提案した。PSM は、

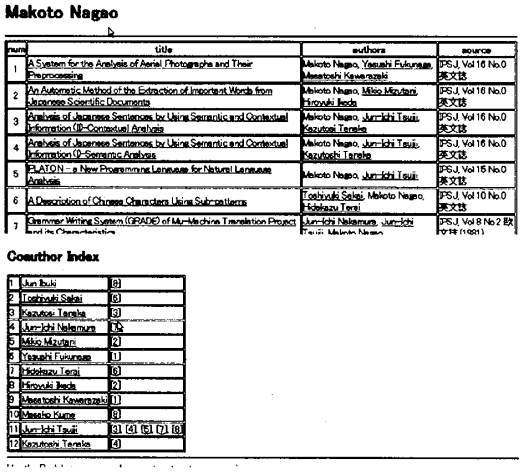

図 4　共著者リスト

WebDB とその結合の情報と表示するチャンネルの情報を記述したスクリプトから CGI プログラムを生成し、それを実行することで実現される。従来のメタ検索との違い、(1) 表示チャンネルの選択により必要な情報だけを抽出し、(2)CGI リンクの埋め込みによって無駄な反復作業を排除することが出来る。また、(3) 基本フィルタの機能によって複数の WebDB から新しい情報を抽出できる。

参考文献

[1] H. He, W. Meng, C. Yu, Z. Wu, *WISE-Integrator: A System for Extracting and Integrating Complex Web Search Interfaces of the Deep Web*, Proceedings of the 31st International Conference on Very Large Data Bases (VLDB2005), Trondheim, Norway, August 30 - September 2, 2005. pp.1314- 1317.

[2] W3C-TR, *Web Services Choreography Description Language*, `http://www.w3.org/TR/ws-cdl-10/`

[3] Alexander Ahern and Nobuko Yoshida, *Formalising Java RMI with Explicit Code Mobility*, Proc. OOPSLA'05, 2005.

[4] Philip Wadler, *"Links"* `http://groups.inf.ed.ac.uk/links/`

[5] Z. Wu, V. Raghavan, C. Du, K. Sai C, W. Meng, H. He and C. Yu, *SE-LEGO: creating metasearch engines on demand*, Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '03), 2003.

[6] *Project DAISEn: Directory Architecture for Integrated Search Engines*, `http://daisen.cc.kyushu-u.ac.jp/`