九州大学学術情報リポジトリ
Kyushu University Institutional Repository

マッシュアップ・リソースとマッシュアップ・グルー

森，雅生
九州大学大学評価情報室

中藤，哲也
九州大学情報基盤研究開発センター

廣川，佐千男
九州大学情報基盤研究開発センター

https://hdl.handle.net/2324/14662

出版情報：WebDB forum. 2008, 2008-12-01
バージョン：
権利関係：

マッシュアップ・リソースとマッシュアップ・グルー

森　雅生[†]　中藤哲也[††]　廣川　佐千男[††]

近年，店舗や駅などの検索結果と地図情報とを融合するサービスの実現手法としてマッシュアップが利用されるようになど，実用的で興味深いマッシュアップの例が多数見られるようになった．しかし，手元ににあるデータと Web 上のサービスを組み合わせて簡単な処理を施し結果をまとめることや，それらの処理プログラムを試行錯誤して実務者が開発するといったような，実務的作業の能率をあげるといった観点でのマッシュアップ開発は少ない．本稿ではまず，マッシュアップを，対象 (mashup resouce) と結合法 (mashup glue) という二つの観点で捉えるプログラミング・スタイルを提案する．マッシュアップ対象は，API や検索サイト，そして手元の CSV ファイルなどであり，入力型と出力型が規定された機械としてとらえる．マッシュアップ結合法は，出力と入力の単純結合の他に，マージ，ソート，CGI リンク，各種グラフ表示などのフィルター機能からなる．さらに，ブラウザ上で開発できるマッシュアップ開発環境を構築した．

Mashup Resource and Mashup Glue

Masao Mori ,[†] Tetsuya Nakatoh [††] and Sachio Hirokawa[††]

Recently many entertaining mashups have been introduced, where mashup technique has been used to realize integration of web services, for example, web mapping services and search results about shops and stations from web search engines. However there are few studies how to develop practical mashups such as reporting and integrating both web data and spread sheet data on local PC, and how to improve business efficiency using mashup. In this paper, we propose a new programming style of mashup with respect to two perspectives, objects(mashup resource) and their functional composition(mashup glue). Mashup resources includes WebAPI, web search engines and local data on users' computers. We consider them as I/O machines with data types. Mashup glue includes simple composition from output to input, filtering functions such as merging, sorting and CGI links, and visualizing components. In addition we developed a Web-based programming environment of mashups.

1. 序　文

ブログに代表される CMS(コンテンツ・マネージメント・システム) は，Web のコンテンツ整形と公開とを自動的におこなうシステムである．CMS の登場により，HTML や Web の専門的知識がなくても，容易に Web 作成をすることが可能となった．その一方で，巨大なデータベースをバックエンドに持つ多くの Web サイトが，情報の共有化によるサービスの質向上を目的として，WebAPI を公開している．WebAPI とは，そのサイトが内包するデータベースを，ブラウザを経由せずにプログラムなどによって操作することができる関数群のことである．CMS が持つ重要な機能の 1 つとして，他サイトの情報を WebAPI を用いて取得し，加工したものを自分のサイトに表示することがある．これを**マッシュアップ** と呼び，複数の Web サイトからの情報を組み合わせてコンテンツを作り，異なった価値を生み出そうとする，コンテンツの新しい提示スタイルである．

例えば，不動産情報のマッシュアップサイト HousingMaps[5] は，コミュニティサイト craigslist[2] が提供する住宅情報から住所データをプログラムによって抽出し，その場所を Google Map によって表示するといった検索機能の直列的結合を行っている．通常，住宅物件を探している人は，craigslist のようなサイトで気に入った物件を検索し，個々の住所データを手動で Google Map の検索サイトにコピーし，あらためて地図の検索を行うであろうが，このサイトはそれを一度に実行している．ここで，マッシュアップにより以下に述べるような省力化と合理化が可能であると考えられる．

ユーザへの省力化　… ユーザが情報検索で行う定型

† 九州大学 情報基盤研究開発センター, Research Institute for Information technology, Kyushu University
†† 九州大学 大学評価情報室, Institutional Research Office , Kyushu University

的なコピー・ペーストの自動化.

データベースの管理運営の省力化 … 住宅情報と地図情報のデータベースを自前で持たず，他の専門サイトの機能を必要に応じて利用している.

この 2 つの点から見て，マッシュアップ技術が複数のデータベースを統合し実務的なサービスを提供するツールとしての価値があると言える.

本稿では，マッシュアップを，対象 (mashup resource) と結合法 (mashup glue) という 2 つの観点で捉えるプログラミング・スタイルを提案する．マッシュアップの対象は，API や検索サイト，手元の CSV ファイルなどであり，入力型と出力型が規定されたコンポーネントと考える．また，マッシュアップにおけるの結合法とは，コンポーネント間の出力と入力の単純結合と，マージ，ソート，CGI リンク，グラフ表示などのフィルター機能との結合からなる．そして本研究では，ブラウザ上でマッシュアップを作成することができる GUI 開発環境を構築した．この開発環境は，専門的知識のないユーザでも簡単にマッシュアップできることを目標にしている．さらに，複数のユーザでマッシュアップを共有する枠組を持ち，ユーザ間で過去に作成したマッシュアップの再利用が可能となっている.

2. 関 連 研 究

2005 年頃から数多くのマッシュアップが登場しているが，近年では実務的なマッシュアップの研究開発が行われている．柳澤ら [19] は，環境問題に関する個人レベルの意見や提案を収集するためのマッシュアップを GoogleMap を用いて開発している．これは，文字による情報と地理的な位置情報を統合したマッシュアップである．GoogleMap を利用することで，実際に起こっている環境問題の記事やコメントが地図上に示される．福井ら [22] は，行政機関に散在する電子申請システムの入力手続きを一元化するシステムを，マッシュアップ技術を利用して開発した.

このように，マッシュアップの実用面での利用価値が注目されてきているが，専門的知識の必要性と開発の繁雑さが無視できないことも指摘されており，より簡潔なマッシュアップ開発手法が求められている．横山ら [20] は，JavaScript を用いたマッシュアップ開発に関して，定型的な手続きを整理し，マッシュアップのコーディングが簡潔になるようなモジュール化手法を提案した．一方，幸城ら [23] により，マッシュアップ開発における重複した手続き呼び出しを合理化するためアスペクト指向を導入し，開発者に見通しのよいプログラミング環境が提案されている．また，マルチメディア・コンテンツの部分的引用を実現するために開発された Annphony[18] は，マッシュアップの多様性を高めている.

Maximilien らの Swashup DSL[6] や Sabbouh らの WMSL[13] は，マッシュアップのためのドメイン固有言語 (domain specific language, DSL) である．JavaScript や CGI を実装するスクリプト言語など，汎用的なプログラミング言語の使用しないことで，マッシュアップ開発の簡易化をはかっている．また，ブラウザ上で一貫して作業することを目標にして，ブラウザでのマッシュアップ開発と実行ができる統合開発環境も提案されている．例えば Marmite[15] は，HTML データを自動的に解析して属性付きデータを抽出し，パイプライン処理によってデータを加工するマッシュアップ開発環境で，クライアント PC で動作する．ユーザが PC 上でパイプライン処理を指定し，Web データとロカールな表データを統合することができる．同様に，MashMaker[3] もローカルな PC 上で個人的なツールとして手軽にマッシュアップを作ることができる開発環境である．また，RSS フィードを統合するマッシュアップの開発環境として，Yahoo! Pipes[12] や Mashup Feeds[14] が挙げられる．状況依存型アプリケーションとしてのマッシュアップを開発する環境 Damia[1] などがある.

このように，マッシュアップの効率的な開発について，様々な開発手法や開発環境が提案されている．しかし，実務的なマッシュアップを行うには，手軽にマッシュアップをプログラムできる GUI 環境や，手元にあるデータと Web のデータを統合して処理する豊富なツールが必要である．また，最初からプログラムするのではなく，他のユーザが作成したマッシュアップを利用して新しいマッシュアップを開発できるといった共有機能も必要である.

著者らは，対象とそれらの結合法と言う観点から，マッシュアップ CGI を生成するドメイン固有言語 *PSM* とその実行環境を 2006 年に発表した [7][8][17]．現在，情報を配信する仕組みの観点から，WebAPI は大きく次の 3 つに分類することができる.

(a) HTML データの配信や RSS フィードなど，一方向型の情報配信.

(b) 検索エンジンや WebAPI を用いて検索機能など，双方向型の情報配信.

(c) サーバの持つ情報の追加や更新を許す情報共有型のサービス.

本論文では，(a) と (b) の 2 つのサービスのマッシュ

アップに焦点を当て，これまでに著者らが開発してきた開発環境に加え，GUI を用いたプログラミング環境とマッシュアップを共有する枠組を提案する．

3. マッシュアップ・リソース

マッシュアップを行う上で,「異なる Web のデータや個人が持つ表データなど，形式が異なるデータをシームレスに処理したい」という要件がある．ここでは，マッシュアップの対象となる Web サイトなどのリソースとデータの抽象化を行う．

マッシュアップの素材となるものを**マッシュアップ・リソース**（または，単にリソース）と呼ぶ．マッシュアップ・リソースの入力は，属性を付きの検索キーワードの組み合わせ (**レコード**) である．また，リソースからの出力は入力に対する検索結果であり，属性付きデータの組み合わせの集合，すなわちレコードのリストである．

マッシュアップ・リソースとして対象となるものを次のように 3 つに分類した．

(1) WebAPI を備えた Web データベース．多くの場合，検索結果は XML 形式で返される．
(2) WebAPI を備えていない Web データベース．検索結果をレイアウトした HTML データが返される．
(3) 個人の PC に保存されている CSV ファイル．

(1) は XML データではあるが，使われるタグが固定的なレコードデータのリストであることが多い．そこで，注目すべきタグを属性としておけば，本質的に表データと同等のものが得られる．このリソースに対するアクセスモジュールでは，WebAPI を呼び出して XML データを取得し，タグ名を属性名としてレコードのリストに変換している．全文検索か詳細検索かは，WebAPI のサービス内容によって様々であるので，利用できるものを利用した．(2) については，出力される HTML を解析しラッパーを開発する必要があり，利用する Web データベースごとにラッパーを作成し，アクセスジュールに組み込んだ．(3) については，事前にサーバにアップロードし，アクセスモジュールを自動生成する．アクセスモジュールの動作は，あるカラムにキーワードを与え，そのカラム内の全文検索を全レコードで行う．

例として，情報処理学会電子図書館および電子情報通信学会の論文データベースをマッシュアップ・リソースとして，ラッパーとアクセスモジュールを作成したものを解説する．図 1 では，それらのリソースの入出力属性を示している． リソースの各属性を**ポート**と呼ぶ．リソース ISPJ の結果に表れる著者名を，データベース IEICE のキーワードポートに渡すことを**単純機能結合**と呼び，マッシュアップのドメイン固有言語 PSM[7)8)17)] では，次のように表現する．

	属性名	値の説明
入力	keyword	キーワード
出力	title	題名
	author	著者名
	datevol	発行年月
	cat	分野
	page	ページ
	pdf	PDF

図 1 リソース IPSJ(情報処理学会電子図書館) と IEICEJ(電子情報通信学会論文 DB)

	属性名	値の説明
入力	item	データ
出力	who	データ
	point	値
	hist	∗ グラフ

図 2 フィルタ Histgram

```
ISPJ.author -> IEICEJ.keyword
```

リソース名とポート名はドット ('.') で区切り，矢印 ->の左辺と右辺にある項を**コンポーネント**と呼ぶ．マッシュアップ・リソースの場合はリソース・コンポーネントと言う．

4. マッシュアップ・グルー

マッシュアップ・リソースの検索結果を別のマッシュアップ・リソースに渡すには，リソースと入出力ポートの対応を記述すればよいが，ユーザに検索結果をどのように提示するかが問題である．ここで紹介する**マッシュアップ・グルー** とは，マッシュアップ・リソース間のデータフローを繋ぐものであり，先に述べたマッシュアップ・リソースを単純機能結合と，ユーザに表示するための出力機能とデータ加工のためフィルタ機能などの総称である．マッシュアップ・グルーは次の 7 つの機能からなる．

初期クエリ・コンポーネント Start … ユーザから最初の検索キーワードを受け取るコンポーネント．リソース IPSJ の keyword ポートに文字列を渡す場合は次のように指定する．

```
Start.x -> IPSJ.keyword,
```

定数コンポーネント Const … 出力ポートの属性名を文字列として出力するコンポーネント．

出力コンポーネント Output … リソース・コンポーネントからの検索結果やフィルタ・コンポーネントの処理結果などを表形式でブラウザに示すため

のコンポーネント．Output1,Output2 などと複数指定することができる．例えば，リソース・コンポーネント IPSJ からの検索結果で，論文題名と著者名だけを表示したいときは次のように記述する．

```
IPSJ.title:author -> Output.title:author,
```

複数のポートと記述する場合は，コロン (':') で区切る．出力コンポーネントの入力ポートは，左辺のコンポーネントの出力ポートの属性もそのままコピーするので，次のように省略することもできる．

```
IPSJ.title:author:page -> Output,
```

フィルタ・コンポーネント … 入力された属性のデータについてフィルタ処理を行うコンポーネント．現在，ソート Sort，ヒストグラム Histgram(図 1)，出現名詞頻度を NounFreq を計算するコンポーネントがある．例えば，リソース・コンポーネント IPSJ に出現する論文の著者でヒストグラムを作る場合は次のように指定する．

```
IPSJ.author -> Histgram.item,
Histgram.who:point -> Output1,
```

出力コンポーネント Output1 にヒストグラムが得られる．

可視化コンポーネント … フィルタ処理済みのデータをグラフ化するコンポーネント．棒グラフ，円グラフ，タグクラウドなどがある．例えば，上記フィルタ・コンポーネントで処理した著者名出現頻度を棒グラフで表示するには次のように指定する．

```
IPSJ.author -> Histgram.item,
Histgram.who:point -> BarGraph.id:value,
```

BarGraph の出力属性 id が項目名，出力属性 value が対応する値となってグラフが描画される．

マージ … 明示的にコンポーネントは無いが，1 つのコンポーネントへ複数のコンポーネントからの入力がある場合は，各コンポーネントからのデータが 1 つのコンポーネントへマージされる．

CGI リンク … マッシュアップの実行で，ブラウザに表示される文字列をキーワードにして，指定したマッシュアップ・リソースで検索するためのトリガーとなるハイパーリンク．例えば，リソース IPSJ から論文名と著者名を出力コンポーネントで表示し，著者名でリソース IEICEJ に検索をする CGI リンクを作るときは次のように指定する．

```
IPSJ.title:author -> Output.title:author,
Output.author -> IEICEJ.keyword,
```

この場合，共著の論文では複数の人名が 1 つのキーワードとして渡されるので，人名ごとに個別に渡したい場合は，

```
Output.author* -> IEICEJ.keyword,
```

と指定する．CGI リンクは出力コンポーネントの他にタグクラウド・コンポーネントでも指定できる．

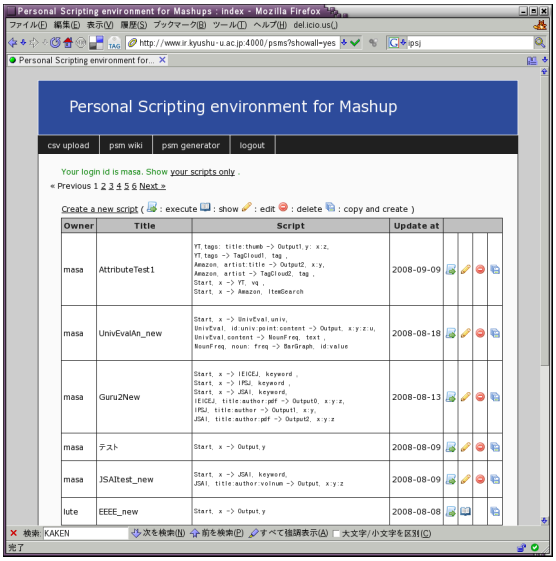

図 6 GUI 環境メイン画面

このように，マッシュアップ・リソースとマッシュアップ・グルーを組み合わせてデータフローの記述をすることで，マッシュアップを作成することができるが，同時にこの作業は，コンポーネントを点としポートの対応を枝とするグラフを構成することにもなっている．PSM の記述で与えたマッシュアップは，それが表すグラフについて次の 2 つの条件を満たさなくてはならない．

- グラフは少なくとも 1 つの初期クエリ・コンポーネントか定数コンポーネントを含まなくてはならない．
- グラフはサイクルを含むことができる．但し，サイクルのパス上に少なくとも 1 つの出力コンポーネントが存在しなくてはならない．

1 番目の条件は，マッシュアップが実行されるには初期値が必要であることを表しており，2 番目の条件はサイクルが無限ループにならないことの条件である．一般に，コンポーネント間を流れるデータはレコード・リストである．リソースの単純機能結合において，n 個のレコード・リストの各々に m 個の検索結果が返されれば，全体として $n \times m$ 個のレコード・リストが返されることになる．

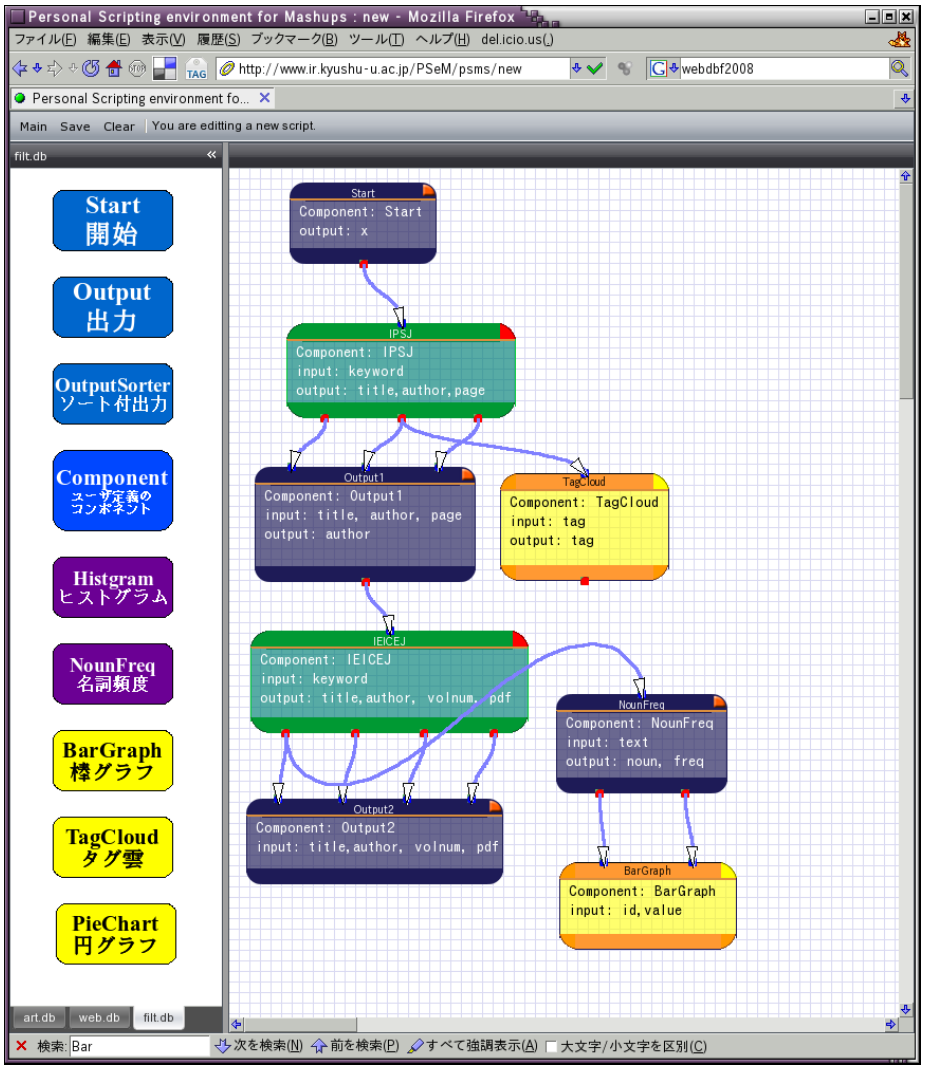

図 3 GUI プログラミング画面

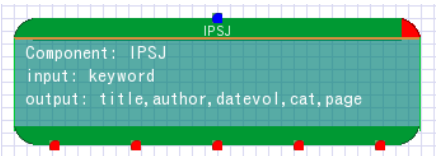

図 7 GUI 環境でのコンポーネント

5. GUI プログラミング環境と共有機能

前節で述べたマッシュアップのプログラムは，ドメイン固有言語 PSM での記述であった．本研究ではさらに容易なプログラミング環境として，PSM コードを生成する GUI によるプログラム開発環境を構築した．この開発環境は，マッシュアップ・コードのデータベースとなっており，ユーザ認証を持つ Web アプリケーションである．他のユーザが作成したマッシュアップ・コードも閲覧，実行ができるようになっている．図 6 はログイン後のメイン画面である． 登録されているマッシュアップ・コードがリスト表示されている．左から，作成者，タイトル，コード (PSM 言語)，更新日時とユーザ権限によって実行・編集・閲覧・削除・複写のアイコンが並ぶ．

図 3 はマッシュアップ・コードの GUI 編集画面である．左側のペインからコンポーネントをドラッグしキャンバス上にドロップすると，コンポーネントが定義される．図 7 は，リソース・コンポーネントである．矩形の上部にある青色の点が入力ポート，下部にある赤色の点が出力ポートである．属性名は矩形の

図 4 マッシュアップ実行画面 1

中の input，output の記述にある．記述の順に上下のポートが対応している．矩形を任意の部分をダブルクリックするとコンポーネントの記述が自由に編集できる．入力と出力のポートの対応は，赤い出力ポートをドラッグし青い入力ポートにドロップすると定義される．

図 3 の例では，リソース IPSJ から得られた検索結果を Output1 に表示し，同時に著者名の出現頻度を表すタグクラウドを表示している．また，Output1 の出現する著者名で，リソース IEICEJ への CGI リンクを作成しているが，それが実行されると，Output2 に検索結果が表示され，そのなかのタイトルに含まれる単語の出現頻度がグラフ表示される．このマッシュアップの実行結果を図 4 に示す．また，CGI リンクの実行結果を図 5 に示す．

PSM 言語は Perl で実装したが，GUI 環境とマッシュアップ・コードのデータベースはそれぞれ JavaScript と Ruby on Rails で開発した．

6. まとめと課題

本論文では，マッシュアップ・リソースとマッシュアップ・グルーという 2 つの概念でマッシュアップ開発をとらえ，新しいマッシュアップ・プログラミングのスタイルを示した．それに基づき，ブラウザ上でマッシュアップを作成することができる GUI 開発環境を構築した．この環境は数分で簡単にマッシュアップを作成できるようになっており，複数のユーザとマッシュアップを共有する枠組を持つ．また，過去に作成したマッシュアップの再利用が可能である．

関連するシステムの代表的なものに Yahoo Pipes

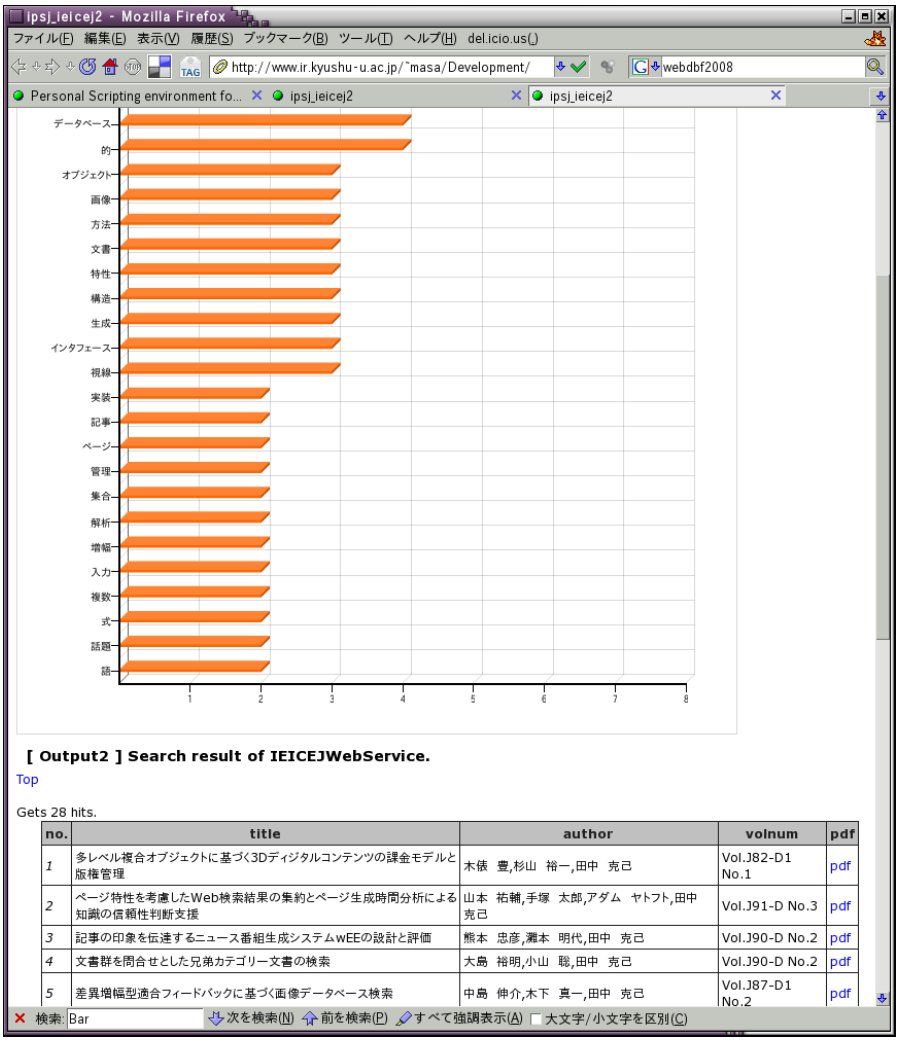

図 5 マッシュアップ実行画面 2

があるが，提案するシステムとの違いを述べておく．まず，想定される利用目的が異なることが挙げられる．Yahoo Pipes の目的は，フィードを加工することである．フィードは，ユーザが受動的に取得する情報である．これに対して，提案するシステムの目的は，ユーザによる情報検索の結果を加工することである．言い替えれば，ユーザがある目的で取得した情報について，統計量を計算したりグラフなどで可視化するといった利用を想定している．2 つ目に，マッシュアップ処理後の出力先の多様性が異なる．Yahoo Pipes の出力先は単一である．よって，出力をブログなどの部品として用いるのに適している．一方，提案するシステムの出力は，複数のコンポーネントで構成することができる．CGI リンクの導入により，実行時の状況変化に応じて出力先に柔軟性を持たせる必要があったため，出力は複数のコンポーネントに分けられるように設計した．

今後の課題を以下に挙げる．

マッシュアップ・リソースのアクセスモジュールは，開発環境から直接プログラムできない．WebAPI やラッパーが必要な Web データベースは， アクセルモジュールを別途プログラムして環境に組み込む必要がある．また，アクセスモジュールには，通信プロトコルや XML および HTML の処理に関する専門的知識が必要である．

マッシュアップ・コードはデータフローを表す単純な構造しか持っておらず，汎用的なプログラミング言語で表現される制御構造は無い．但し，CGI リンクを用いて，検索の繰り返しを行うことは可能である．例えば，Yahoo! Pipes はフィルター処理をする部分で

制御構造を導入しているが，本論文の枠組でおこなうマッシュアップ開発にそのような制御構造が必要であるか，また，制御構造を導入したときと同じ表現能力が，導入する前の枠組で可能かどうか検討する必要がある．

参 考 文 献

1) Mehmet Altinel, Paul Brown, Susan Cline, Rajesh Kartha, Eric Louie, Volker Markl, Louis Mau, Yip-Hing Ng, David Simmen, and Ashutosh Singh. Damia: a data mashup fabric for intranet applications. In *Proceedings of th 33rd international conference on Very Large Databases*, pp. 1370–1373. VLDB Endowment, 2007.
2) craigslist. http://www.craigslist.org/.
3) R. J. Ennals and M. N. Garofalakis. Mashmaker: Mashups for the masses. In *SIGMOD 2007: ACM SIGMOD International Conference on Management of Data*, pp. 1116–1118, 12 June 2007 through 14 June 2007 2007.
4) Kevin Hemenway and Tara Calishain. *Spidering Hacks*. O'Reilly & Associates Inc., Mar. 2003. ISBN-13 978-0596005771.
5) HousingMaps. http://www.housingmaps.com/.
6) E. M. Maximilien, A. Ranabahu, and S. Tai. Swashup: Situational web applications mashups. In *22nd International Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA 2007 - Companion*, pp. 797–798, 21 October 2007 through 25 October 2007 2007.
7) Masao Mori, Tetsuya Nakatoh, and Sachio Hirokawa. Functional composition of web databases. In *Proceedings of International Conference Asian Digital Libraries 2006*, Lecture note in Computer Science 4312. Springer Verlag, 2006.
8) Masao Mori, Tetsuya Nakatoh, and Sachio Hirokawa. Links anc cycles of web databases. In *Fourth Italian Workshop on Semantic Web Applications and Perspectives*, pp. 21–30. Dip. di Informatica, Universitá degli Studi di Bari, December 2007.
9) Tetsuya Nakatoh, Keisuke Ohmori, and Sachio Hirokawa. A report on metadata for web databases. In *IPSJ SIG Technical Reports*, 2004-ICS-138(17), pp. 95–98, 2004.
10) Tetsuya Nakatoh, Keisuke Ohmori, Yaushiro Yamada, and Sachio Hirokawa. Complex query and metadata. In *Proceedings of ISEE2003*, pp. 291–294, 2003.
11) Tetsuya Nakatoh, Yasuhiro Yamada, and Sachio Hirokawa. Automatic generation of deep web wrappers based on discovery of repetition. In *Proceedings of the First Asia Information Retrieval Symposium (AIRS 2004)*, pp. 269–272, 2004.
12) Yahoo! Pipes. http://pipes.yahoo.com/.
13) Marwan Sabbouh, Jeff Higginson, Salim Semy, and Danny Gagne. Web mashup scripting language. In *Proceedings of the 16th international conference on World Wide Web 2007*, pp. 1305 – 1306. ACM Press, May 2007.
14) Junichi Tatemura, Arsany Sawires, Oliver Po, Songting Chen, K. Selcuk Candan, Divyakant Agrawal, and Maria Goveas. Mashup feeds: Continuous queries over web services. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pp. 1128 – 1130. ACM Press, June 2007.
15) J. Wong and J. I. Hong. Making mashups with marmite: Towards end-user programming for the web. *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1435–1444, 28 April 2007 through 3 May 2007 2007.
16) Jian Yang. Web service componentization. *Communications of the ACM*, Vol. 46, No. 10, pp. 35–40, 2003.
17) 森雅生, 中藤哲也, 廣川佐千男. マッシュアップの軽量実装のための提案. In *Proceedings of Data Engineering Workshop 2007*, C7-152. IEICE, 2007.
18) 梶克彦, 長尾確. 部分引用の管理に基づく web コンテンツのマッシュアップ. 情報処理学会第 69 回 全国大会, 2007.
19) 柳澤剣, 小室匡史, 綿貫理明, 大西寿郎. 集合知を利用した環境情報システム. 情報処理学会研究報告 IS, 2008. 2008-IS-103.
20) 横山昌平, 的野晃整, サイド ミルザパレビ, 小島功. Web2.0 における javascript コードのモジュール化とマッシュアップの枠組み. *DBSJ Letters*, Vol. 5, No. 3, 2006.
21) 中藤哲也, 大森敬介, 廣川佐千男. 「検索エンジンを部品とするエージェントの構成について」. In *The Third Joint Agent Workshop & Symposium (JAWS2004)*, 2004.
22) 福井宏紀, 岩田彰, 若山公威, 鈴木春洋. マッシュアップ技術を用いた官民連携統合電子申請システムの提案. 情報処理学会研究報告 CSEC, 2007. 2007-DPS-130, 2007-CSEC-36.
23) 幸城祐樹, 三村治朗, 上田賀一. アスペクト指向を用いたマッシュアップ構築支援システムの開発. 情報処理学会研究報告 SE, 2008. 2008-SE-159.