マッシュアップを利用したWebサービス構築支援システムとその応用

A Support System for Implementing Web-based Mashup Applications

佐野 博之*1 Hiroyuki Sano　平田 紀史*1 Norifumi Hirata　大囿 忠親*1 Tadachika Ozono　新谷 虎松*1 Toramatsu Shintani

*1名古屋工業大学大学院 工学研究科 情報工学専攻
Dept. of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

Web service providers, such as “Google”, “Amazon”, “Yahoo!” and so on, provide Web application programming interfaces(Web APIs) that can be accessed over the Internet. By calling Web APIs, users execute requested services on a remote system and get data that Web service providers own. In Web development, developers combines data from one or more Web APIs into their Web applications. That is called “Mashup Programming”. However, there are some points to be considered on mashup. To use the output of a Web API as the input of a different one, the output and the input format must be consistent. To address the problem, we propose a support system for implementing mashup Web applications. The system enables users to create mashup Web applications easily. Our system can automatically find a chain of Web APIs to achieve a user’s goal. Our goal is that non-programmers can create new mashup Web applications by using the system.

1. はじめに

Web ビジネスを展開する多くの企業が，外部に対して自社の Web サービスを提供するための手段として，Web API を公開している．開発者はこれらの Web API を使用することにより，インターネット上の様々なサービスを自分の Web サイトに組み込む事が可能となる．また，これらの Web API を複数組み合わせてひとつの新たな Web サービスを構築することをマッシュアップと呼ぶ．

複数の Web サービスを組み合わせるためには，それらの Web サービスを呼び出すための Web API を適切に組み合わせる必要がある．しかしここで問題となるのが，“Web API の入出力の整合性” である．具体的には，wa_1，wa_2 という 2 つの Web API の入出力を組み合わせてマッシュアップを行う場合，wa_1 の実行結果が wa_2 の入力として使用できることが前提となる．wa_1 の実行結果の全てが wa_2 の入力として使用できないとしても，一部が使用できる可能性もある．また現時点で公開されている Web API は多数あり，wa_1 の実行結果とともに wa_3 という Web API の実行結果を組み合わせて wa_2 の入力として使用できる場合も存在する．開発者がマッシュアップを用いて Web サービスを構築する際には，これらの入出力の整合性を考慮しつつプログラミングを行う必要があり，煩雑である．

本稿では上記の問題を解決するために，入出力の整合性を考慮した Web API の組み合わせを推論するための手法について検討する．

本研究では，ユーザがあるタスクを実行する際に，“既知の入力パラメータ”，“最終的に欲しい出力の種類” をシステムに対して与えることで，そのタスクを完了するための Web API の組み合わせを自動で導出することを最終目標としている．

連絡先: 佐野博之，名古屋工業大学大学院 工学研究科 情報工学専攻，〒 466-8555 愛知県名古屋市昭和区御器所町，Tel:052-733-6550，Fax:052-735-5584，E-Mail:hsano@toralab.ics.nitech.ac.jp

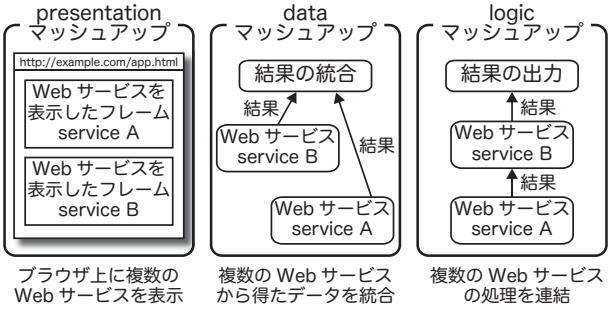

図 1: Dornan によるマッシュアップの分類

2. マッシュアップ支援

2.1 マッシュアップの分類

文献 [Dornan 07] では図 1 に示すようにマッシュアップを presentation，data，そして logic の 3 種類に分類している．

presentation マッシュアップでは，同一 Web ページ内に複数のコンテンツを組み合わせて表示する．iGoogle や My Yahoo のようなポータルサイトが例として挙げられる．もっとも単純なマッシュアップで，HTML の編集でも実現できる．

data マッシュアップでは，より高度で，複数の情報源から得たデータを合成してサービスを提供する．地図上に別のデータを重ね合わせる例が挙げられる．マッシュアップ開発環境を用いることで，プログラミング無しでも実現可能である．

logic マッシュアップでは，さらに高度で，複数の Web サービスの入出力を連結させる．一般的にプログラミングが必要とされる．旅行支援アプリケーションなどで，最安値のフライトを発見し予約するような，ワークフローを含むアプリケーションが挙げられる．しかし，連結可能な Web サービスの組み合わせを考える際には，Web API の入出力の整合性について考える必要があり，手間と時間がかかる．

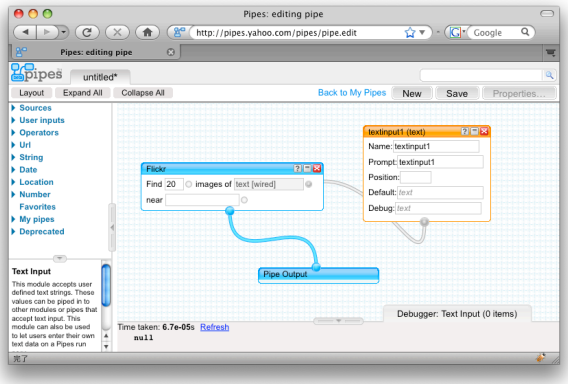

図 2: Yahoo!Pipes の実行例

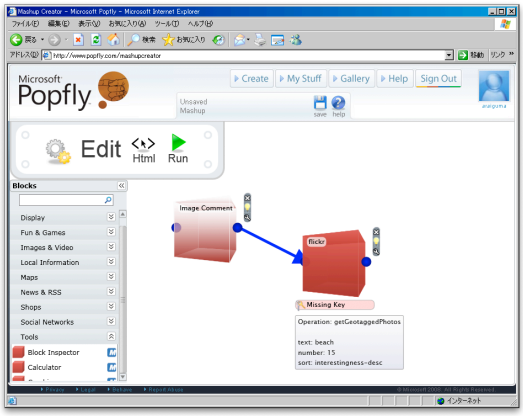

図 3: Microsoft Popfly の実行例

2.2 既存のマッシュアップ支援システム

本システムに関連して，プログラミングの知識がないユーザがマッシュアップを行うための開発支援環境を紹介する．複数の Web サービスを組み合わせるためには，それぞれの Web サービスの入出力の整合性をチェックする必要があるが，ユーザにとって手間や知識が必要であり，プログラム開発経験のない一般的なユーザには敷居が高い．そのようなユーザでもマッシュアップを可能にする支援環境として，Yahoo!Pipes[*1] と Microsoft Popfly[*2] を紹介する．これらの支援環境では，図式化された Web サービスを，グラフィカルなエディタで編集することで，プログラミング不要のマッシュアップを実現している．

Yahoo!Pipes は，Yahoo!社が提供するマッシュアップ支援環境である．Yahoo!Pipes では，Web API の提供する機能が図式化されたモジュールとして提供されている．ユーザは，“Pipes Editor” と呼ばれるグラフィカルなエディタを用いて，使いたい機能を持つモジュールをパイプで接続することでマッシュアップを行う．ユーザはプログラムを書く必要がなく，モジュールをドラッグ&ドロップしていくだけでマッシュアップを行える．図 2 は，Yahoo!Pipes を用いて 2 つのモジュールをパイプで接続する例である．

Microsoft Popfly は，Microsoft 社が提供するマッシュアップ支援環境である．図 3 は，その実行画面である．Yahoo!Pipes と同様，グラフィカルなエディタ上でモジュールを配置し、それらのモジュールを接続するだけでマッシュアップが完成する．Popfly で作成したマッシュアップは Silverlight[*3] アプリケーションとして実行可能である．

これらの支援システムでは，システムに登録済みの Web API のみを利用可能であり，ユーザが任意の Web API を利用することができない．誰でも簡単に Web API を登録できるような仕組みの開発が必要である．Web API の登録をユーザに許可した場合，その Web API の入出力の整合性をチェックするための機構が必要となる．

また，これらのマッシュアップ支援システムとは少々異なるが，Softbot[Etzioni 94] という，情報収集エージェント環境が存在する．Softbot とは，ユーザの要求を満たすような UNIX コマンド系列を，エージェントが自動でプランニングするためのエージェントである．生成される UNIX コマンド系列には World Wide Web からの情報収集も含まれている．

*1 http://pipes.yahoo.com/
*2 http://www.popfly.com/
*3 Web ブラウザ上でのマルチメディアコンテンツ再生環境．http://www.microsoft.com/silverlight/を参照．

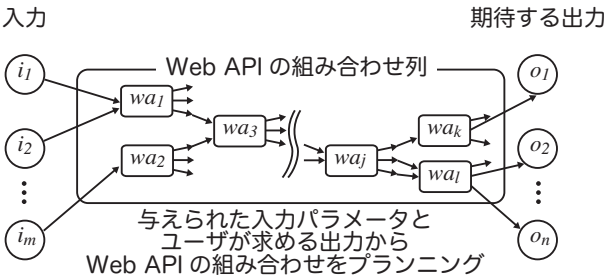

図 4: 本研究の最終目標

3. マッシュアッププランナー

本研究の最終目標は，初期値と目的とする出力の種類をシステムに対して与えると，システムが自動でその入出力を満たすための Web API の組み合わせを求めることである．具体的には，図 4 に示すように，ユーザが初期値 $I = \{i_1, i_2, \cdots i_m\}$ と目標の種類 $O = \{o_1, o_2, \cdots o_n\}$ をシステムに対して与えると，初期値から目標を得られるような Web API の組み合わせ，つまり図 4 の四角で囲まれた Web API の組み合わせ列を，システムが自動で求めることである．

3.1 Web API 入出力の整合性

ある Web API の実行結果を他の Web API の入力として使用可能なことを，Web API の入出力の整合性と呼ぶこととする．図 5 は Geocoding API[*4] と座標から最寄駅 API[*5] をマッシュアップするときの入出力の例である．Geocoding API に対して地名やランドマーク名をパラメータとして与えると，その地名，ランドマーク名の緯度と経度が XML 形式で取得できる．図 5 の右側がその実行結果の XML 構造であり，XML をパースして “lat”（緯度），“lng”（経度）を切り出し，それを入力パラメータとして座標から最寄駅 API を呼び出している．

しかし，図 5 から分かるように，Geocoding API の実行結

*4 地名やランドマーク名から，その緯度と経度を取得するための Web API．http://www.geocoding.jp/api/
*5 緯度と経度から，その場所に近い駅に関する情報を取得するための WebAPI．http://www.ekidata.jp/tools/api_station_c.html

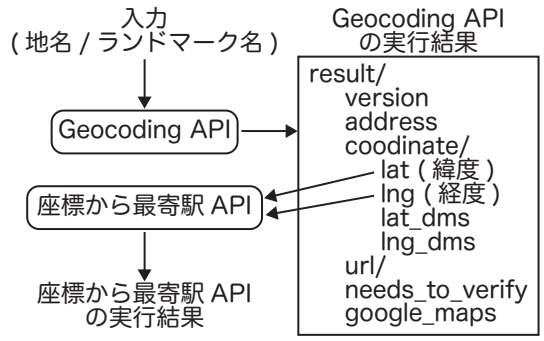

図 5: Web API をマッシュアップするときの入出力例

果には座標から最寄駅 API に必要のない情報も含まれている．開発者がマッシュアッププログラミングを行うときには，ユーザが Geocoding API の実行結果から座標から最寄駅 API に必要な情報だけを切り出し，切り出した情報を座標から最寄駅 API の入力として渡すようなプログラムを作成する必要がある．そのようなプログラムを作成することで，Geocoding API の実行結果を座標から最寄駅 API の入力として使用することができるため，Geocoding API と座標から最寄駅 API は整合性があると言える．

この入出力の整合性に関する問題を解決することがマッシュアッププログラミングにおける課題であると考える．本稿で提案するシステムでは，帰納推論を用いて XML のタグ構造の分類を行い，この課題の解決を試みる．

3.2 整合性推論機構

文献 [Eki 08] では，XML のタグ構造やタグのデータ構造を述語として表現し，制約論理プログラミングに基づく帰納推論によって XML のタグ構造を分類するという手法を提案している．本システムではこの手法を，Web API リポジトリに保持されている XML に対して適用する．

本システムには，Web API リポジトリという Web API の情報を保持しておくための専用のリポジトリを用意した．このリポジトリには，以下の情報を XML で保存しておく．これらの情報はいずれも Web API の解説がしてある Web ページから取得することができる．

- name: Web API の名前
- url: Web API を利用する際に指定する URL
- param, type: Web API の入力パラメータ名と，そのパラメータのデータ型
- output, type: Web API の出力と，その出力のデータ型

これらの XML に対して帰納推論に基づく XML のタグ構造分類を行うことにより，入出力の整合性を考慮した Web API の組み合わせを推論することが可能となる．

近年，インターネット上に存在する様々な情報を，コンピュータがその意味を考えることにより組み合わせて利用可能とするための試みがある．インターネットに膨大な情報が溢れるようになった今日，情報を探し出すことが大変困難になってきている．インターネットの情報は人間が理解することを目的に書かれており，全文検索エンジンなどの機械が処理できるように考えられて書かれていない．それぞれの情報が何を意味するのかを記述した情報，つまりメタデータを用意しておくことにより，コンピュータが意味を理解して自動処理を行えるようにし，より高度で精度の高い検索を可能にしようとい

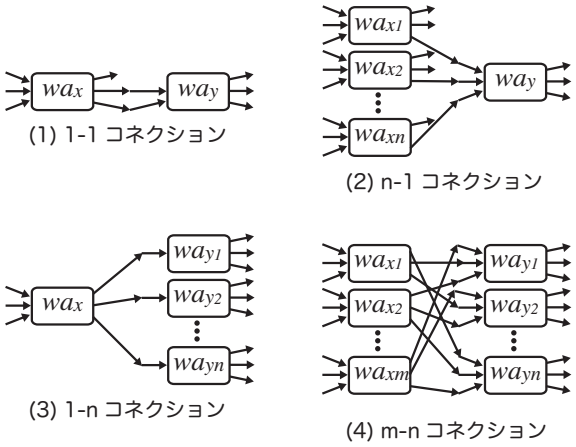

図 6: Web API の組み合わせパターン

う考えが出てきた．機械がメタデータを利用して意味を理解し，高度な情報検索を行うという次世代の Web を Semantic Web[Knublauch 06] と呼ぶ．Web の情報に対して，メタデータという機械可読式目録のようなデータを入力し，効率よく検索できるようにするので，Semantic Web はウェブ上の情報を 1 つの世界規模のデータベースとして使用しようとしていると言える．

さらに高度な検索を行うためには，語彙と語彙の関係などの概念の体系を表す必要がある．語彙間の関係を表すものをオントロジー (Ontology)[Bechhofer 04] と呼ぶ．オントロジーとは存在論を意味し，哲学の分野で用いられてきた用語である．

このような概念を Web API リポジトリに取り込む事により，人手ではなく，計算機による高度なマッシュアップが可能になると期待できる．

3.3 Web API の組み合わせ

logic マッシュアップにおける Web API の組み合わせ方には，図 6 に示すように，以下の 4 つのパターンが考えられる．

(1) $wa_x \rightarrow wa_y$
ひとつの Web API の実行結果を 1 つの Web API の入力として使用する．(図 6 左上)

(2) $\{wa_{xi} \mid i = 1, 2, \cdots, n(n \geq 2)\} \rightarrow wa_y$
複数の Web API の実行結果を 1 つの Web API の入力として使用する．(図 6 右上)

(3) $wa_x \rightarrow \{wa_{yj} \mid k = 1, 2, \cdots, n(n \geq 2)\}$
1 つの Web API の実行結果を複数の Web API の入力として使用する．(図 6 左下)

(4) $\{wa_{xi} \mid i = 1, 2, \cdots, m(m \geq 2)\}$
$\rightarrow \{wa_{yj} \mid j = 1, 2, \cdots, n(n \geq 2)\}$
複数の Web API の実行結果を複数の Web API の入力として使用する．(図 6 右下)

これらのマッシュアップをそれぞれ，(1) から順に “1-1 コネクション”，“n-1 コネクション”，“1-n コネクション”，“m-n コネクション” と呼ぶこととする．1-n コネクションは 1-1 コネクションの延長であり，また，m-n コネクションは n-1 コネクションおよび 1-n コネクションの複合であるため，本システムでは基本となる 1-1 コネクションと n-1 コネクションを対象とする．

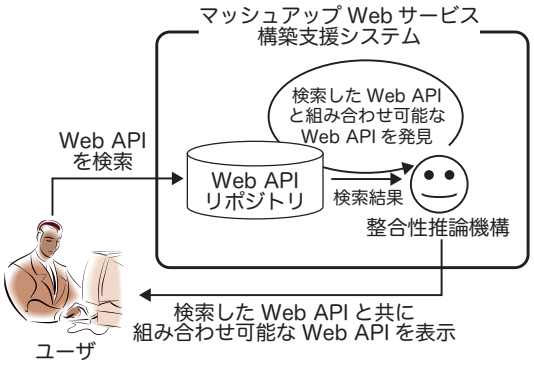

図 7: システム利用の流れ

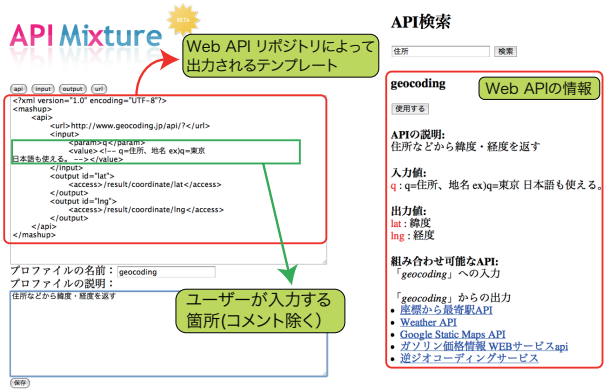

図 8: システムの実行例

3.4 組み合わせの探索

ユーザが求める出力から初期値へと向かって探索を進めることによって，Web API の組み合わせを求める．第 3.3 節で述べた 1-1 コネクション，n-1 コネクションを考慮し，与えられた初期値になるべく一致するような Web API の組み合わせを求める．ユーザによって与えられた入力値を入力パラメータとして使用する Web API が含まれる状態が得られた場合に，探索を終了する．

ただしユーザが求める要求を満たす Web API の組み合わせを求めるにあたって，ユーザによって与えられた入力値だけでは不十分である可能性もある．その場合には，システムがユーザに対して他に必要なパラメータを示すこととする．

4. システムの試作

本稿で提案するマッシュアップ Web サービス構築支援システムは，ユーザが登録した Web API の情報をもとに Web API の入出力の整合性を推論し，組み合わせ可能な Web API をユーザに対して掲示する．ユーザが本システムを利用する時の流れを図 7 に示す．

ユーザは本システムを使用するとき，まずは Web API リポジトリ [Goto 09] に対して Web API の検索を行う．検索結果は整合性推論機構に渡され，整合性推論機構はその検索結果をもとに，Web API リポジトリに対して検索結果の Web API と組み合わせ可能な Web API を推論し発見する．検索結果と推論結果をユーザに対して表示することで，ユーザは Web API の組み合わせを知る事が可能となる．本システムでは，このようなマッシュアップ Web サービスの構築支援を行う．

図 8 に本システムの実行例を示す．ユーザーは図 8 右部の検索窓に検索ワードを入力することで Web API に関する情報を得ることができる．検索された Web API の情報は，図 8 右部に表示される．Web API に関する情報には組み合わせ可能な Web API の一覧が含まれており，即座に組み合わせ可能な Web API を見つけることができる．

5. おわりに

本稿ではマッシュアップを利用した Web サービスを構築する際に問題となる Web API 入出力の整合性について考察を行い，それらの整合性を推論しユーザに対して組み合わせ可能な Web API を掲示するためのシステムを試作した．

本稿では Web API リポジトリを用意し，その中にそれぞれの Web API の情報を XML で保存しておく．保存された XML に対して制約論理プログラミングに基づく帰納推論を適用することで，XML のタグ構造を分類し，組み合わせ可能な Web API を推論するという手法を提案した．

Web API リポジトリに対してセマンティクスを導入し，また，知的なプログラムであるエージェント技術により Web API の抽象化が進めば，エージェント間の連携に基づくマッシュアップが可能になり，より高度で知的な Web アプリケーションをプログラミングなしに実現できる．Web API に関するマッチメイキングのような，エージェントに基づくマッシュアップに関して，今後の研究を進めていく必要がある．

参考文献

[Bechhofer 04] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein, “OWL Web Ontology Language Reference, ” `http://www.w3.org/TR/owl-ref/`, W3C Recommendation, 10. Feb. 2004.

[Dornan 07] Andy Dornan, “Mashup Basics: Three for the Money,” Network Computing, `http://www.networkcomputing.com/showitem.jhtml?articleID=201804223`, 2007.

[Eki 08] 驛昌弥, 大囿忠親, 新谷虎松, “帰納推論に基づく XML タグ構造分類による XML 検索,” 第 70 回情報処理学会全国大会論文集. Mar. 2008.

[Etzioni 94] Oren Etzioni, Daniel Weld, “A Softbot-Based Interface to the Internet,” Communication of the ACM, Vol.37, No.7, pp.72-76. July. 1994.

[Goto 09] 後藤友和, 大囿忠親, 新谷虎松, “マッシュアップ・プロファイルを考慮した Web API リポジトリの試作,” 第 71 回情報処理学会全国大会論文集. Mar. 2009.

[Knublauch 06] Holger Knublauch, Daniel Oberle, Phil Tetlow, Evan Wallace, “A Semantic Web Primer for Object-Oriented Software Developers,” `http://www.w3.org/TR/sw-oosd-primer/`, W3C Working Group Note, 9. Mar. 2006.