

データの編集・登録が可能なエンドユーザ向け情報統合環境

熊谷 良夫[†] 仙波 雅也[†] 鎌田十三郎[†]

[†] 神戸大学大学院システム情報学研究科情報科学専攻 〒 657-0013 兵庫県神戸市灘区六甲台町 1-1

E-mail: †{kumagai,senba,kamada}@cs26.scitec.kobe-u.ac.jp

あらまし 近年, 様々な Web サービスが提供される中で, 個人データなどのより身近なデータを扱えるものも増えてきている。そして, マッシュアップによりそれら Web サービスを複数組み合わせることで, 新たな Web サービスや Web アプリケーションの提供が可能となっている。本研究では, エンドユーザが複数の Web サービスをその場の興味に応じて結合しながら, 個人のデータを編集・登録可能な情報統合環境の提案と実装を行う。提案環境では, 閲覧中のデータに対して連想されるデータを関連付けながら, 元データの編集や登録を行うことができる。本論文では, データ統合を簡単化するために, データモデルとして Web サービス上のデータをテーブルとしてアクセスできる環境を想定している。

キーワード 情報統合環境

An environment for information integration with editable views of joined Web services

Yoshio KUMAGAI[†], Masaya SENBA[†], and Tomio KAMADA[†]

[†] Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501 Japan

E-mail: †{kumagai,senba,kamada}@cs26.scitec.kobe-u.ac.jp

Abstract In recent years, many types of information have been provided through Web services. Mashup technology allows us to combine different types of information in order to provide a new integrated service. We propose and implement an environment for information integration on which end-users can join several Web services instantly, and can create and edit personal data. We assume Web services that can be treated as tables. The user can join tables by GUI operation specifying join conditions, and edit the joined table to add or modify associative data on a single view.

1. はじめに

近年, Web サービスを利用することで, 各種情報をアプリケーション上で簡単に利用することができるようになっている。また, 複数の Web サービスを組み合わせた新たなサービスを作るマッシュアップも盛んに行われている。開発者は, 複数の Web サービスの API を組み合わせることで, マッシュアップアプリケーションを作成することができる。各種マッシュアップツールも提供されており, 例えば, Yahoo! Pipes [8] では, Web サービスの接続関係を GUI 操作で示すだけで, 簡単にマッシュアップを新たなサービスとして実現できる。一方で, エンドユーザがマッシュアップを利用する環境として, 閲覧ページに対して, 付加可能なマッシュアップ要素を選択表示できる MashMaker [2] [1] などの環境も提案されている。

本研究では, エンドユーザがより身近なデータを様々な用途

に利用できるようにするべく, Web サービス上のデータを自在に統合利用でき, ユーザ自身のデータを他のデータと連携させた形で登録/編集できる環境を実現する。また, 連携データを自在に結合し, 編集することの有用性を検討するとともに, 連携データを結合し単一ビュー上で編集するための課題と解決方法について述べる。データ統合の簡単化のため, データモデルへの仮定としてデータ表現は統一が図られているとする。加えて, Web サービス上のデータはテーブルとしてアクセスできるものとし, ユーザデータも同様のテーブルとして格納できるものとする。また情報統合としては結合演算のみを扱うこととした。このような仮定の下で, 各種データの結合表示や編集が可能な情報統合環境の作成を行い, そのアプリケーションとしての有用性を議論する。

連携されたデータを自在に統合利用できることのメリットは, 各データをただ一つの用途だけでなく, データのつながりを工

夫することで、別の意味あるデータとして利用できることにある。例えば、レストラン訪問時の履歴・コメントなどを残すケースを考える。レストランに関する情報は Web サービスなどで提供されており、例えば、店舗名や電話番号、緯度経度といった情報が利用可能である。一方のユーザコメントは、投稿者・個々のレストランへの参照・コメント内容などの組を持つ。本環境では、それぞれレストランテーブルやコメントテーブルとして利用し、特定の地域や条件で検索・表示したレストランに対して、ユーザコメントが気になれば、各レストラン要素に対するコメント一覧を表示するためのリンクボタンを準備できる。また、各レストランに対する自分のコメントを眺めながら、新規レストランへのコメントを登録する場合には、レストランに自身のコメントを左結合した結果を一覧表示しながら、コメント登録することも可能である。さらに、このレストランとコメントのテーブルは他の用途にも使うことができ、例えば、ユーザが自身の投稿したコメントに対してレストラン情報を左結合することで、ユーザの食べ歩き記として利用することもできる。

データ閲覧は、表形式が基本であるが、緯度経度などの属性をもつテーブルについては、地図上表示なども可能となっている。当該環境は、GUI 操作可能な Web アプリケーションとして整備中であり、javascript フレームワークである ExtJS を使用した実装を行っている。

本稿の構成は以下のとおりである。2 章では本研究と関連する先行研究について述べる。3 章では本システムが想定するデータモデルと利用スタイルについて実例を交えながら説明する。4 章では本システムの実現において特徴的なトピックとして、結合テーブル上での編集の実現手法などについて述べる。そして 5 章では本研究のまとめを述べる。

2. 関連研究

既存のマッシュアップツールには、サービス間のデータ接続関係を静的に記述し、何度も利用できるようなマッシュアップアプリやサービスを作成する立場と、エンドユーザが閲覧している個々のデータについて、その場で組み合わせを行える環境を作成する立場が存在する。

前者の中には、Yahoo! Pipes [8] や Popfly [5], Plagger [6] のように、Web サービスの接続関係を GUI 操作などにより指定できるようにすることで、開発者が、より簡単にマッシュアップを行うことを目指したツールがある。また、Popfly には、取得データの種類に応じた表示を行うための UI Block も備えている。これらツールにより、開発者はスクラッチからプログラミングせずに新たな Web サービスや Web アプリケーションを簡単に作成できる。しかし、データ接続には、Web サービスの入出力データ形式の理解なども必要であり、データを閲覧中に、思いついた Web サービスをその都度組み合わせしていくという使い方を想定したものではない。

一方、エンドユーザがデータを閲覧・操作する形で、マッシュアップを行っていくことのできるツールとして、Marmite [7] や Google Fusion Tables [3] があげられる。Marmite では、スプレッドシート形式で結合データを表示し、カラムを追加する

形で追加要素の表示を行っていく。ただし表示形式からも、フラットなデータ構造しか扱うことができない。Google Fusion Tables [3] は、Google Docs に保管しているテーブル形式データを扱い、ユーザ間でのデータ共有も可能である。ユーザは、見ているテーブルに対して属性を選び、別のテーブルを左結合することができる。また、各種データ形式に応じて、地図や棒グラフなどの表示形式も選ぶことができる。結合データをテーブル形式で表示した際には、ビュー上において属性値のデータ編集が可能であるが、結合条件指定した属性値の編集はできない。また、左テーブルの要素に関連する右テーブルの要素がない場合、その要素の生成を行えない。

Mash Maker [1], [2] は、エキスパートユーザが各 Web ページに対し、有用と思われるマッシュアップを作成し、ウィジェットとして公開する事で、一般のエンドユーザが興味に応じて閲覧ページに対して付加可能なウィジェットを選択表示できるようにしたものである。ウィジェットは、対象ページをソースとする他の Web ページとの静的なデータ接続関係を定義し、表示形式も含めて定義するものである。木構造のデータモデルを採用するなど、柔軟なウィジェット作成ができる半面、エンドユーザは、作成されたウィジェットの中からしか、マッシュアップを選択することはできない。実際のデータ合成は、ウィジェット表示の必要に応じて行われる。我々のグループでもマッシュアップ環境 [4] を作成しており、Web サービスの接続関係を宣言的に記述するが、インタラクティブなビューアを備えることで、ユーザが画面上で表示させようとした項目についてのみ要求駆動でデータ合成を行うという環境を実現している。

3. システム概要

今回作成している情報統合環境では、データモデルの簡単化のため、Web サービス上のデータやユーザデータが、関係データベースのテーブルとして利用可能な状況を想定している。エンドユーザは、GUI 操作が可能なデータビューア（以下、単にビューアと呼ぶ）を通して、これらのテーブルを結合した結果を表示したり、結合結果を見ながらデータの登録/編集操作などをおこなうことができる（図 1）。以下では、システムが想定するデータモデルの概略と本環境の利用スタイルについて、実例を交えながら説明する。

3.1 データモデル

本システムでは、データはテーブルに格納されているものとして扱う。ユーザデータを格納するためのテーブルについては CRUD 操作が可能とし、Web サービスによって検索取得可能な情報については、読み取り操作のみが可能とする。本システムでは、利用に先立ってユーザのログイン認証を行い、ユーザデータについては、当該利用者の権限で可視なデータのみ検索でき、同様にデータ更新についても権限設定されているものとする。ユーザデータのアクセス制御モデルについては、別論文 [9] として発表する。

本論文では、各行相当をエントリと呼び、エントリの属性とその型は、テーブルごとに決められる。各エントリは識別のための ID が与えられ、テーブルエントリを参照するための参照

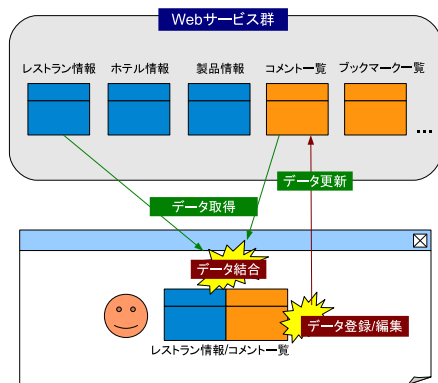


図 1 システム概観

型データとして利用される．参照型には，任意のエントリを参照可能な型と，指定テーブルのエントリのみを参照可能な型を準備している．データ型としては，緯度経度に代表されるような一般的に利用されるデータ構造を表現するためのレコード型も用意し，例えば，緯度経度のレコード型属性であれば，緯度と経度を 1 つの組として値を保持する．

これらのテーブルは，各種用途に応じてアプリケーション開発者相当が定義するものであり，一般のエンドユーザが作成するものではない．加えて，テーブル定義時には，エンドユーザがテーブル検索を行うための，キーワードなどを定義可能であり，テーブル構造とともに，メタ情報としてシステムに管理される．エンドユーザは，自分が利用したいテーブルをビュー上で検索して用い，キーワード，属性名や型（参照型・レコード型）情報などを検索条件に利用可能である．例えば，緯度経度情報を持ち，レストランというキーワードを持つテーブルを検索するといった具合である．

ビュー上では，検索したテーブルをそのまま表示するだけでなく，現在表示しているテーブルに別のテーブル情報を左結合する形で表示することができる．例えば，レストランテーブルに対して，コメントテーブル（コメント対象を示す任意テーブルへの参照型属性を持つ）を結合して表示する．また，ユーザが更新可能なエントリについては，ビュー上で登録・編集・削除などを行う．この場合，自身のコメントの追加・編集が該当する．ビュー上でのユーザの要求をうけて，システムは各テーブルに対する検索・更新操作を要求し，権限がある操作であれば実行されることになる．

3.2 ビューを通しての情報閲覧

本節では，1 節で述べたレストラン・コメント情報の閲覧の例を用いて，ビュー上でどのように情報の閲覧・結合・編集を行うことができるのか，ユーザの利用スタイルを紹介する．

まず最初にレストランテーブルを表示する場合は，テーブル自体の検索から開始することになる．単にレストランというキーワードを指定するだけでなく，レストランの場所を検索条件として指定したり地図上に図示したいような場合は，緯度経度型属性の存在を条件として指定することになる．テーブルが気に入れば，ブックマークして次回以降使うことも可能である．

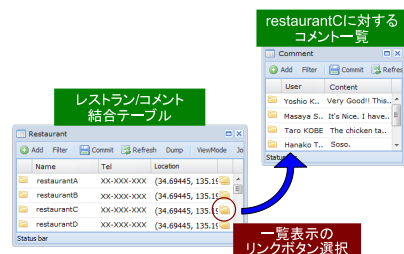


図 2 各レストランに対するコメント一覧操作

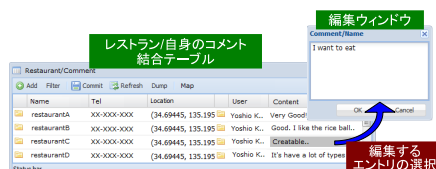


図 3 新規レストランへのコメント登録操作

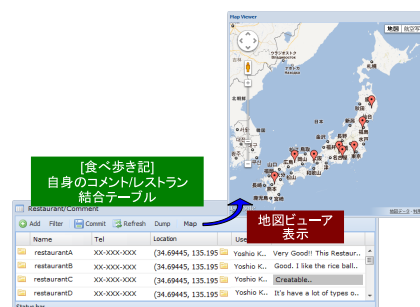


図 4 自身の食歩き記の地図ビュー上での表示

レストラン情報の閲覧時に，各レストラン要素に対する他人のコメントが気になり，レストランテーブルに対してコメントテーブルを統合する．コメントテーブル自体は頻繁に利用されるものであり，参照型をもちいた結合条件もあらかじめ決めておけるため，左結合候補として結合設定コンソールにあらかじめ登録している．結合内容の表示については，各レストラン要素に対し，コメント一覧を表示したいようなケースなので，図 2 のように一覧表示のためのリンクボタン表示を選択する．ユーザが各リンクボタンをクリックする度に，対応するコメントの検索が実行され，別のパネルに結果が表示されることになる．

一方で，各レストランに対する自分の評価・コメントを眺めながら，新規レストランへのコメントを登録する場合は，図 3 のように，結合結果をテーブル表示を行い，自分のコメントの登録や編集の操作を行うこともできる．これは，当該ユーザが，同じレストランに対し，ただだか 1 回しかコメントを書かないつもりである場合に有効である．コメントテーブルへの検索条件の追加（自身のコメントへの制限）と，表示形式の変更は，表示パネル上の結合要素を表すカラムなどを通して行うことができる．コメントを左結合して表示する際，まだコメントが投稿されていない場合，コメントテーブルの当該エントリには，Creatable と書かれたマスクがかかっており，このようなエン

トリに対し、データの登録操作をすることでコメントを投稿できる。この例では、レストランテーブルは読み取り操作しか許されていないため、編集・登録はシンプルであるが、両テーブルともに編集可能な場合には、4.1 節で述べるような編集制約を設けることで、編集状態が意味をなさないようになることを防いでいる。

次に、コメントテーブルに対してレストランテーブルを左結合した場合について述べる。ユーザは、自身の食べ歩き記を表示することを意図しており、まずは、コメントテーブルを選択し、自身のレストランに対するコメントに限定して検索し、その結果を表示する。検索条件の指定は、各パネルのフィルタコンソールを利用することで指定可能である。レストラン情報の結合は、属性表示されているレストラン参照のヘッダを操作し、結合テーブルの表示を行う。レストランの場所の位置関係を見たいと考えれば、そのデータを地図表示する。図 4 は、食べ歩き記のデータを地図ビューアで表示している様子である。各表示パネルには緯度経度属性を地図ビューア表示するためのボタンが配置される。

このように、エンドユーザの要望に応じてテーブルの結合ができれば、その意図に応じた情報の取得も可能である。例えば、あるレストランに対するコメント一覧を表示している際、コメントをつけている人が「どのような人」であるのか興味をもつこともあるであろう。例えば、レストランの客層を知るために、ユーザの年齢情報を表示したいかもしれない。あるいは、別の有名店に対してどんなコメントをつけていないか興味があるかも知れないし、どんなお酒・スポーツに興味があるのか知りたいかもしれない。このように、ユーザは興味に応じているようなデータをつなぎ合わせながら情報を閲覧することができる。

4. システム実現

本節では、本システムの実現において特徴的なトピックについて述べるとともに、現在のシステム実現状況について述べる。

4.1 結合テーブルに対する編集と整合性

左結合テーブルに対する編集を行う際、例えば、結合条件に用いた属性を編集すると、結合対象が変わってしまうかもしれないため、表示状態が意味をなさなくなる恐れがある。この種の問題に対応するため、左結合テーブルを表示する際、パネルは編集可能な属性を制限することで、表示が不整合な状態に陥るのを防ぐ。

まず、右テーブルの結合条件に関わる属性の値の編集をさせないという制約を設け、結合条件に関わらない属性のみの編集を許可する。この制約に加え、結合条件において左右エントリの属性値等価判定を行っている場合、右テーブル当該属性を予め非表示にする GUI を提供する。

一方の、左テーブルの属性については、結合条件に関わる属性が編集された場合は、新たな属性値に基づいて、右テーブルのエントリが自動的に再取得するという方針を採用した。ただし、右エントリが編集集中である場合は、再取得によって編集集中データのクリアを防ぐため、左エントリの結合条件属性の編集を禁止することにした。右エントリのコミットもしくは編集

キャンセルが行われるまで、禁止状態が継続する。同様の理由で、右テーブルの結合条件自体の変更についても、右テーブルにおいて編集項目がある間、禁止することにしている。

4.2 各種データ型の表示と編集

レコード型データの典型例として、緯度経度がある。緯度経度は緯度と経度を 1 つの組として値を保持するが、表示の際は、地図上に表示した方が効果的である。このため、緯度経度のレコード型属性を持つテーブルに対しては、地図パネル表示のためのボタンを設けている。また、緯度経度の編集は、地図パネル上のマーカーの移動として操作できるようにしている。日時・時刻データについても、同様の表示・編集パネルを準備中である。

参照型属性は、その値自身をテキスト表示してもあまり意味がないため、表示の際は、参照先のエントリを表示するためのボタンを配置している。一方で、エントリ ID 相当のアイコンも表示するようにし、各種コンソール表示のために利用する。

表形式パネルの上では、ドラッグ&ドロップによってセル単位での値のコピーを行うこともでき、レコード型や参照型属性（エントリ ID 相当を含む）の編集に利用することができる。その際、型に互換性がない値のコピーは無効となる。ドラッグ&ドロップ操作は、テーブル間の結合条件を指定するための結合設定コンソールにおいて、利用属性や型、定数条件を指定する際にも利用することができる。

4.3 システム実装

本システムの実装にあたっては、javascript フレームワークである ExtJS を使用し本システムの環境を実装しており、サーバサイドは、並行して開発をおこなっているアクセス制御機構付き Web データベース [9] を用いて実験を行っている。Web サービスデータについては、あらかじめ上記データベースに値を格納することで対処しているが、現在、実 Web サービスへのプロキシ機能も実装中である。

システム実装に用いた ExtJS の特徴の一つは、多彩な GUI コンポーネントであり、ウィンドウやボタンなどの各種機能が提供されている。加えて、サーバデータとの連携の仕組みが充実しており、データ型を表現するための Model、集合を表現し CRUD API を有した Store、Store への CRUD 操作をデータサーバに伝達することやエンコード/デコードを担当する Proxy などの仕組みが提供されている。このため、例えば GUI コンポーネント上でフィルタ条件を設定すると、Store へのフィルタ設定として伝わり、Proxy を介して XMLHttpRequest が発行されるといった仕組みが、基本的に提供されている。

各テーブルの実現も、Model 及び Store を用いて実現しており、テーブル構造に応じた Model 定義は、サーバから取得したテーブル構造情報をもとに、動的に行うことで解決している。加えて、左結合テーブルについても結合エントリ及びテーブル相当の Model、Store を準備することで実現を行っているが、本論文の主題から外れるため詳細は割愛する。

システムは、現時点では一部実装が終わっておらず、論文で図示したマッシュアップ表示などは完成しているが、テーブル検索や結合条件指定コンソールは現在実装中である。

5. ま と め

本論文では、各 Web サイトが提供する Web サービスや、個人のデータを自由に作成・編集可能な Web データベースを使用したエンドユーザ向けの情報統合環境の提案し、その利用事例の検討とシステムの要件を取りまとめた。現在システムは実装中である。

謝 辞

本研究の一部は、科学研究費補助金基盤研究 (C) 22500091 の補助を受けたものである。

文 献

- [1] Rob Ennals, Eric Brewer, Minos Garofalakis, Michael Shadle, and Prashant Gandhi, “Intel Mash Maker: join the web,” *SIGMOD Record*, Vol. Vol. 36, No. No. 4, pp. 27–33, (2007).
- [2] Rob Ennals and David Gay, “User-friendly functional programming for web mashups,” in *the 2007 ACM SIGPLAN international conference on Functional programming (ICFP ’07)*, pp. 223–234, ACM, (2007).
- [3] Google, “Google fusion tables,” <http://www.google.com/fusiontables/Home/>.
- [4] Sohei Ikeda, Takakazu Nagamine, and Tomio Kamada, “Application framework with demand-driven mashup for selective browsing,” Vol. 15, No. 10, pp. 2109–2137, (2009).
- [5] Microsoft, “Microsoft popfly,” <http://www.popfly.com/>.
- [6] Tatsuhiko Miyagawa, “Plagger,” <http://plagger.org/>.
- [7] Jeffrey Wong and Jason I. Hong, “Making mashups with marmite: towards end-user programming for the web,” *CHI ’07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 1435–1444, New York, NY, USA, (2007), ACM.
- [8] Yahoo!, “Yahoo! pipes,” <http://pipes.yahoo.com/pipes/>.
- [9] 仙波雅也, 鎌田十三郎, “個人/共有データを格納するためのロールベースアクセス制御機構付き Web データベースの提案,” *Proc. of DEIM2012*, (2012).