

研究種目：若手研究（B）
 研究期間：2006～2008
 課題番号：18700029
 研究課題名（和文） インタラクティブな WEB アプリケーションの実現を容易にする計算環境
 研究課題名（英文） Application Framework for Selective Browsing of Mashed-Up Data
 研究代表者
 鎌田 十三郎（KAMADA TOMIO）
 神戸大学・大学院工学研究科・助教
 研究者番号：20304131

研究成果の概要：

本研究の目的は、ユーザの要望に応じて Web 上の各種情報を効率的に融合するインタラクティブな Web アプリケーションを、容易に実現可能にすることである。我々の提案・開発したフレームワークは、データモデル部と GUI 部を分離し、高い抽象度での Web アプリケーション記述を可能としている。一方で、データ計算はシステムが要求駆動で自動的に実行されており、提供する GUI コンポーネント群と協調動作することで、高い性能と柔軟な操作性の両立を実現する。

交付額

(金額単位：円)

	直接経費	間接経費	合計
2006年度	1,500,000	0	1,500,000
2007年度	1,000,000	0	1,000,000
2008年度	1,000,000	300,000	1,300,000
年度			
年度			
総計	3,500,000	300,000	3,800,000

研究分野：プログラミング言語

科研費の分科・細目：ソフトウェア

キーワード：Web アプリケーション開発，マッシュアップ，情報融合，要求駆動計算，RIA

1. 研究開始当初の背景

近年、一般向けに公開された Web サービス／アプリケーションを対象に、複数種の情報を手軽に統合し、新たなサービス／アプリケーションとして作成する動きが高まっている。例えば、宿泊施設検索を利用する際も、ホテル周辺のレストラン情報が重要な場合もあれば、特定の会場へのアクセス情報や、指定期間の天候情報が重要な場合も存在する。これら複数の情報を組み合わせることで、

各種ユーザのもつ多様なニーズに応じたアプリケーションを作成することも可能となる。

一方で、このような多様なユーザの要望に応じるために、それぞれ個別のアプリケーションを準備するのは非現実的である。我々は、ユーザが各種要望に応じて、インタラクティブに情報を選択・閲覧できるようなアプリケーションにより、このようなニーズに対応しようと考えた。ただし、このようなアプリケ

ーションを作成する場合、効率的に融合データを準備し、そして、インタラクティブにユーザに提示していく上で、プログラミング上の課題も多い。

2. 研究の目的

本研究の目的は、ユーザの要望に応じて Web 上の各種情報を効率的に融合するインタラクティブな Web アプリを、容易に実現できるようにすることである。まず、本フレームワークで作成された Web アプリ事例(図 1)を通して、その特徴について述べる。

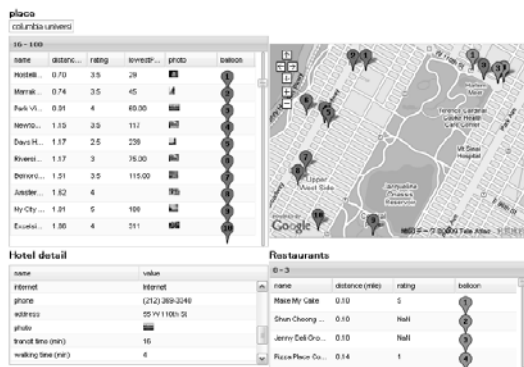


図 1 アプリケーション例

図 1 は、ホテル検索アプリの例であり、左上入力欄から場所に関するキーワードを受け取り、検索したホテル情報を左上テーブルに一覧表示し、右上地図にその場所を示している。このアプリケーションの特徴は、ユーザがインタラクティブに評価項目を選択・追加できる点にある。例えば、左表の列名部分をクリックすることで、付近の写真・ブログの評判・指定した目的地までの所要時間といった情報が動的に追加できる。ユーザは、これらの評価項目を見ながら、表をスクロールダウンしながら希望するホテルを選ぶことができる。興味に合致しない候補が多い場合は、各列の値に対しフィルタ条件を動的に設定し、除外することもできる。興味がある候補があれば、クリックすることで詳細情報を下部の二つの表を通してみることができる。左は、選択ホテルに関する詳細情報を羅列したものであり、右は付近のレストラン情報を表示したものである。

このような Web アプリを実現する場合、ユーザの操作に応じて必要な箇所のみ計算・Web サービス 問い合わせを行うことが重要となる。例えば、大量の検索結果がヒットした際に、ユーザが閲覧するかどうかわからない候補についてまで Web サービス に問い合わせるのは明らかに無駄であるし、フィルタ条件により表示されないような候補についても同様である。

一方で、現状の開発環境上でこのような Web アプリを作成するのは、非常な労力が必要となる。インタラクティブな GUI アプリケーションは、操作的モデルに基づいて作成されるのが一般的であるが、その際、GUI イベントに応じて閲覧対象領域を決定、Web サービスの非同期呼び出し・結果到着時の再描画処理などを実現する必要がある。これらの同期処理は、熟練したプログラミング技能を必要とする。

本研究では、上記のような柔軟かつ効率的に動作する Web アプリを、容易に開発できるようなアプリケーションフレームワークの提案・実装技術の開発を行う。

3. 研究の方法

前述の目標を達成するため、提案フレームワークでは、宣言的かつ抽象度の高いアプリケーション記述によりアプリ開発を容易にすると同時に、システムが必要な部分のみ計算を行うことで、アプリケーションの効率的実行の実現を図る。

開発者はアプリ作成のために、Web サービスの組み合わせ方と、合成されたデータの表示方法を設定ファイルとして記述する(図 2 参考)。システム上も、データモデルと GUI 部は分離しており、効率的に Web 上データの融合を実現する要求駆動エンジンと、高い操作性を提供するための GUI コンポーネント(widget)群が協調動作することで、アプリケーションの柔軟性と高い実行性能を目指す。利用者は、フレームワークが提供する GUI コンポーネントを通して、現在関心のある部分について選択的に閲覧を進めていくことができ、その間、システムは閲覧に必要なデータのみを作成する。

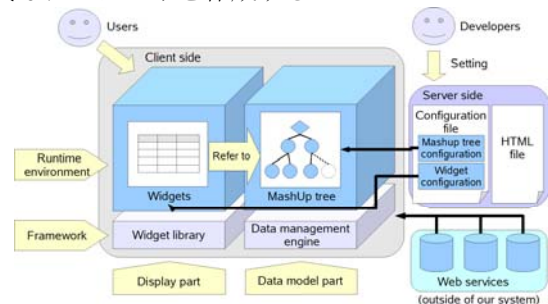


図 2 フレームワーク概要

本研究は、以下の3段階に分けて進められた。

(1) アプリケーションの基本モデル策定
最初のステップとして、本フレームワークにおけるアプリケーションの基本モデルを策定し、種々のアプリケーションを、簡潔かつ柔軟な形で記述可能であるか検討を行った。加えて、作成されたアプリケーションを通してユーザがインタラクティブなデータ閲覧

を行う際、ユーザ動作に応じた要求駆動に基づく漸進的データ構築が有効であるか、プロトタイプシステムの実装を含めて、検討・評価した。

(2) 効率的な要求駆動データ構築エンジンの実現及び widget 部との協調

本システムでは、ユーザの閲覧動作に応じて、データ構築エンジンが閲覧に必要なデータだけを自動的に構築し、widget 部を通じて随時提示する。GUI イベント処理ならびに Web サービスの非同期呼び出し処理は、widget 部ならびにデータ構築エンジンが自動的に行う。これらの処理は Ajax 技術を用いて実現され、作成された Web アプリは通常の Web ブラウザで利用可能である。このため、軽量かつ種々の widget と協調可能な柔軟な実装技術が必要である。

(3) widget ライブラリ及びデータ表示モデルの拡充

最後に、各種アプリケーション分野において、本フレームワークが有効に機能し、効率的なデータ提示ができるように、各種 Web アプリの作成と並行して、widget ライブラリ及びデータ表示モデルの拡充を行った。

4. 研究成果

(1) アプリケーションの基本モデル
([発表 1,8])

我々のフレームワークでは、マッシュアップデータを表現するためのデータモデル部と、データを閲覧するための表示部に分離したモデル化を行っており、開発者はマッシュアップ定義と表示設定をそれぞれ行う。

本フレームワークでは、マッシュアップデータを図 3 のような木構造 (Mashup Tree) として扱う。

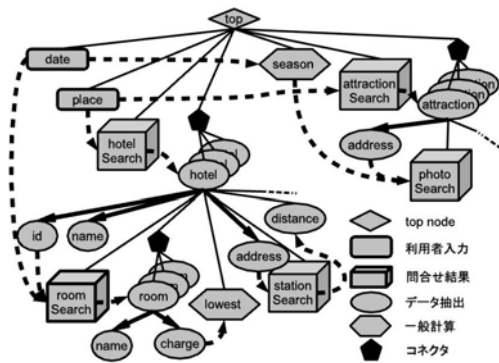


図 3 Mashup Tree

データの流は図中矢印で表現されており、例えば入力された地名からホテル検索を行い、その結果から各ホテルの情報を得るといった一連の流れが表現されている。ホテル一覧のようなリスト形式データは、リスト要素をまとめるコネクタと、要素を表す複数の子ノードの形で表現される。図中実線で表現

される階層関係は、各ホテルと、その名前や空室情報のリストといったデータ間の親子関係を表現する。

マッシュアップデータの表示は、widget に対して、表示対象となる Mashup Tree 中の部分木を指定することで行う。指定法については、項目(3)で解説する。

(2) システムの実装技術
([発表 1, 3, 4, 7, 8])

本システムでは、データ構築はユーザの widget 操作に応じて、表示対象となるデータを要求駆動に構築する。例えば、テーブル widget は表示対象列を動的に変更することができるが、非表示列のデータ構築は行われない。また、スクロールバー付きのテーブルの場合、ユーザのスクロールダウン操作に応じてデータ構築が漸進的に行われる。ユーザが、表示項目にフィルタ条件を設定した場合も、フィルタ条件を満たさない要素については、それ以外のデータ構築を行わない。データ構築に伴う Web サービスの非同期呼び出しや到着データの随時表示は、システムが自動的に行う。

widget 部は、ユーザ操作に応じてデータ構築エンジンにデータ要求を行い、該当するデータが未構築の場合、エンジンは必要となる Web サービス呼び出しを非同期に行う。結果データが到着した時点で、widget の表示対象領域のうち、到着データに依存する領域の計算を行い、widget に再描画を促す。ただし、widget のデータ閲覧範囲は、ユーザ操作に応じて動的に変化するものであり、また、データ構築エンジンは各種 widget との協調動作が必要となる。このため、効率的な Mashup Tree の部分更新手法を新たに開発した。本手法では、Mashup Tree 上の値変化(データ到着など)に依存する領域をシンボリックな形で求めることで、各 widget の管理する表示領域との重複判定を瞬時に行えるようにしている。結果、低コストで無駄なデータ構築を避けることができるだけでなく、データ構築部に関する複雑な処理を各 widget ライブラリから分離することにも成功した。加えて、ユーザが各種入力を試すようなケースや、Web サービスの結果が随時更新されるようなケースにおいても、不要なデータ再構築を抑えることを可能としている。

性能評価により、要求駆動データ構築による不要な Web サービス呼び出しの削減が確認された。例えば、先に示したアプリケーション事例の場合、ベースとなるホテル情報は複数ページに渡る多くの結果を返すことができるが、ユーザが目的地付近のホテルに興味がある場合や、宿泊料金が安いホテルを探している場合は、実際には多くの候補は詳細データを調べないで済むケースが多い。

我々のフレームワークでは、ユーザ操作に応じてデータ構築を行い、到着データから随時表示していくことで高い反応性能も達成している。ある基本性能評価の場合、データ到着から 200msec 以内で構築データの表示（描画）が完了しており、加えて、その大半は描画処理に費やされたものであった。データ構築のための内部処理に要した時間は 20msec 以内であり、前述の領域重複判定も合計で 5-6msec しか要しておらず、軽量である。

(3) widget ライブラリ及びモデルの拡充とアプリケーション評価（[発表 5]）

各種アプリ領域において、Web アプリ上での効果的なデータ表示法を検討し、本フレームワークでの実現法を検討した。その際、Web アプリを作成する上で必要となった、複数 widget 間の連携関係の取りまとめを行い、widget モデルの拡充を行った。現在、以下の 3 種類の仕組みを提供している。

- ① Mashup Tree 中の静的に定まる部分木の表示
- ② widget の property 値に基づく widget 間連携
- ③ 即値データに基づくイベント処理

①は、widget の表示対象領域を Mashup Tree 上の静的な部分木として指定するものである。前述のアプリ例（図 1）では、左上の表がホテル検索結果一覧を表示しているのに対応する。

一方、先のアプリ例では、一覧表示されたホテル候補をクリックすることで、その詳細情報を下部表に表示していた。このような widget 間の連携関係を実現するのが②である。システム提供の widget は、各種 property を備えており、例えば、クリック選択されたノードや、画面表示中の行（相当ノード）を表すための、いくつかの property を提供している。開発者は、「一覧表」の "selected" ノードを、「詳細表」に表示するといった性的な連携関係を記述するだけで、property 値の更新や、widget の再描画はシステムが自動的に行う。

③は、イベント処理によって即値データを取り込み、widget のプロパティやデータモデルの値を更新する際に用いる。具体的には、地図上のクリック座標の取得や、ボタンクリックによりユーザ入力を受け取るようなケースに用いられる。

これらの widget に関するモデルは、要求駆動データ構築と協調可能なようデザインされており、widget による Mashup Tree 中データへの直接参照は許されていない。これは、要求駆動によるデータ構築／更新の際、データの曖昧性が起きるのを避けるためである。

各種アプリ事例によるケーススタディでは、効率的なデータ提示法についても検討し

た。ここでいう「効率的」とは、Web サービスへの漸進的なデータ要求により（つまり、ユーザを待たせる事なく）、ユーザが自身の興味範囲の絞り込みを適切に助けることができるどうかを指す。例えば、地図上でホテルを探す場合、ズームレベルに応じて情報密度を変えるなどの対応をしなければ、俯瞰状態あたりをつけて、拡大地図で詳細情報を調べるといったことができなくなってしまう。先ほどのモデルを使えば、ズーム操作に応じて表示対象を制限するといった処理も記述可能である。一方で、現在のデータモデルに関する課題も確認されている（後述）。

関連研究比較

近年、マッシュアップに代表される Web データ融合の研究は盛んに行われており、各社からも先進的なツールが提案・開発されている（Yahoo Pipes, Intel Mash Maker Google Mashup Editor, Microsoft Popfly など）。中には GUI エディタ上で Web サービス／アプリの接続ができるなど、容易にマッシュアップを作成するための研究も多い。

一方で、本フレームワークで提案しているようなユーザ操作に応じて、漸進的にデータ構築をおこなうようなシステムは存在しない。例えば、Mash Maker では、マッシュアップフラグメントの動的適用などは可能であるが、GUI 操作とデータ構築が連動することはない。

現在の進行状況

現在、指導学生を中心にアプリ開発環境の GUI 化や widget 拡充の試みが進められており、2008 年下期採択の未踏ユースとして採択、現在進行中である。これとは別に、アプリ開発事例を通じて、selection, join といったデータベース演算導入の重要性が把握されたため、データモデル及びその実装に関するさらなる拡充を予定している。

5. 主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 3 件）

- ① 池田宗平, 草野直樹, 長嶺貴一, 鎌田十三郎, “要求駆動によるマッシュアップと AJAX Widget による効率的な閲覧”, コンピュータソフトウェア (2008 年採録決定, 掲載予定)
- ② 草野直樹, 鎌田十三郎, “型安全な再利用可能アスペクトを目指した MJ ベースの AOP 言語とその型システムの提

案”, 情報処理学会 論文誌 プログラミング (PRO), 査読あり, Vol. 2, No.2, pp15-131, 2009

- ③ 長嶺貴一, 池田宗平, 鎌田十三郎, 草野直樹, “マッシュアップデータの選択的閲覧における効率的な部分更新”, 日本データベース学会論文誌 (DBSJ Journal), 査読あり, Vol. 7, No. 4, pp1-6, 2009

[学会発表] (計 5件)

- ④ 長嶺貴一, 池田宗平, 鎌田十三郎, 草野直樹, “マッシュアップデータの選択的閲覧における効率的な部分更新”, WebDB Forum2008, 査読あり, 2008年12月1~2日, 東京
- ⑤ Sohei Ikeda, Takakazu Nagamine, Tomio Kamada, “Application framework with demand-driven mashup for selective browsing”, International Conference on Information Integration and Web-based Applications & Services (iiWAS2008), refereed, 14~15.Nov. 2008, Linz, Austria
- ⑥ 草野直樹, 鎌田十三郎, “型安全な再利用可能アスペクトを目指した MJ ベースの AOP 言語とその型システムの提案”, 情報処理学会第71回プログラミング研究会, 2008年10月29日, 松江
- ⑦ 長嶺貴一, 鎌田十三郎, 池田宗平, 草野直樹, “入力データの部分的変更に対応したマッシュアップアプリケーション開発環境”, 電子情報通信学会 第19回データ工学ワークショップ(2008), 2008年3月9~11日, 宮崎
- ⑧ 池田宗平, 草野直樹, 長嶺貴一, 鎌田十三郎, “要求駆動によるマッシュアップとAJAX Widgetによる効率的な閲覧”, 日本ソフトウェア科学会第24回大会 2007年9月12~14日, 奈良

[その他]

研究成果に関する web ページ :
<http://www.cs26.scitec.kobe-u.ac.jp/farm/PL/?Connecollect>

6. 研究組織

(1) 研究代表者

鎌田 十三郎 (KAMADA TOMIO)
神戸大学・大学院工学研究科・助教
研究者番号 : 20304131

(2) 研究分担者

なし

(3) 連携研究者

なし