マッシュアップ技術のモデル化とその評価

山田 成仁† 松原 裕人‡ 大谷 真†

湘南工科大学 情報工学科†　　湘南工科大学 大学院 電気情報工学専攻‡

1. はじめに

マッシュアップとは、複数の異なる提供元のコンテンツを複合し、新しいサービスを形作ることである。近年広く使われるようになってきた。

マッシュアップの形態には様々なものが考えられる。しかし、明確なモデル化がなされておらずマッシュアップの特性が活かしきれていないことも多い。そのためにモデル化を行った。

本研究では、マッシュアップの形態を 3 つにモデル化し、モデルごとに標準的な実装方法を定義した。更に、各モデルとその標準実装方法を実際のシステムに適用し、各モデルの特徴を評価した。

2. マッシュアップモデル

マッシュアップは、様々なサーバに存在するコンテンツをプログラムとスクリプトを用いてまぜ合わせ利用しやすい形でユーザに表現する技術である。

サーバ側、クライアント側での処理内容の組み合わせにより、多様な形態が可能である。これをマッシュアップ実行場所に着目して分類すると、図 1～3 の 3 つのモデルで表現できる。ブラウザ側でマッシュアップを行う Client 型、Server 側でマッシュアップを行う Server 型。及び、Server 側とブラウザ側の両方でマッシュアップを行う Client+Server 型である。

図に示すようにモデルは次の要素で構成される。

PC:Server に対してリクエストを送信を行うマシン。

Server:PC から URL を指定して直接アクセスを行うマシン。

WebServer: Server 以外の Web サービスを提供するサーバ機

コンテンツ:Server や WebServer が提供する情報

WebAPI:Web サービスの機能を HTTP を用いて関数のように利用できるようにしたもの。

XML:文書やデータの意味や構造を記述するためのマークアップ言語。一般的に WebServer は XML の形式でコンテンツを送り返す。

Script:HTML 内のスクリプト。JavaScript 等がある。

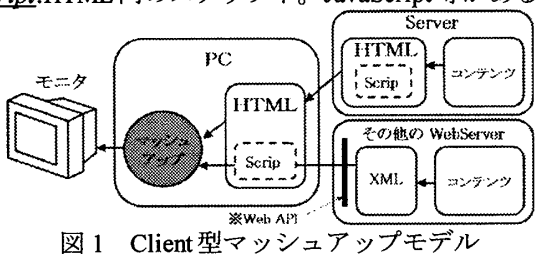

図 1　Client 型マッシュアップモデル

Modeling and Evaluation of Mashup Technology,
Naruhito Yamada†,Hiroto Matsubara†,Makoto Oya†,
Shonan Institute of Technology†

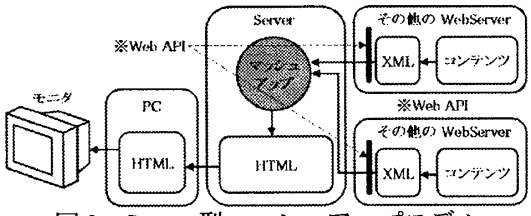

図 2　Server 型マッシュアップモデル

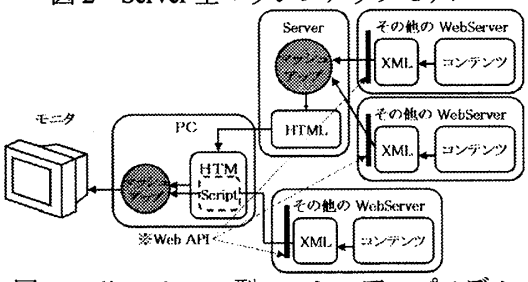

図 3　Client+Server 型マッシュアップモデル

3. モデルの実装方法

3 つのモデルの標準的な実装方法は次のとおりである。

(1) Client 型マッシュアップモデル

HTTP リクエストを受けた時点で、Server はコンテンツを HTML として生成する。この HTML 内にその他の WebServer からコンテンツを受け取るための Script を埋め込んでおく。

PC 側に該当 HTML がダウンロードされた後、その中の Script の内容に従って、WebServer から WebAPI を使ってコンテンツを受け取り、HTML の再描画しブラウザに表示させる。HTML の内容を動的に変更する場合は DynamicHTML を使用、非同期通信が必要な場合には Ajax を用いる。

(2) Server 型マッシュアップモデル

HTTP リクエストを受けた時点で、Server 上に存在するプログラムがその他の WebServer に対し WebAPI を使ってコンテンツを要求する。WebServer 側はリクエストされた WebAPI に対して API 仕様で決められた XML を返す。

Server 側プログラムでは返された XML を受け取り、コンテンツを HTML として生成する。その後、PC 側に Server 側で生成された HTML がダウンロードされブラウザに表示させる。

(3) Client+Server 型マッシュアップモデル

HTTP リクエストを受けた時点で、Server 上のプログラムが Server 型を同様に WebAPI をアクセスする。Server 側にある HTML が PC 側にダウンロードされた後、Client 型と同じようにその他の WebServer を呼び出す。

4. マッシュアップモデルの適用と実装

3つのモデルが4の方法で実現可能であることの確認、及びその場合の各モデルの評価を目的として、実際に3つのモデルに従ったシステムを構築した。

4.1 Client型モデルの適用

湘南工科大学サイトとGoogleMAPを用いてClient型マッシュアップの実験を行った(図4)。

大学サイトはServerに、GoogleMAPはその他のWebServerに該当する。

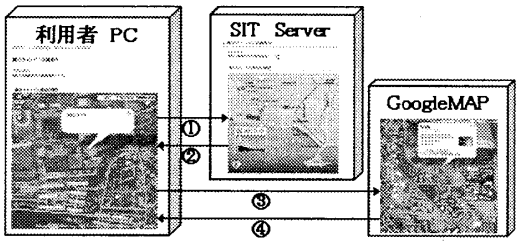

図4 Client型マッシュアップのシステム構成

(1)システムの流れ

① 利用者PCからSITServerにリクエスト送信
② HTMLを利用者PCにダウンロード
③ ダウンロードされたHTML内のJavaScriptからWebAPIでGoogleMAPにリクエストを送信
④ GoogleMAPから返されたXMLと画像をHTML内で決められた場所に表示

(2)JavaScriptの内部処理

まず、'document.getElementById("map")'が地図表示を行う場所を指定、'new GMap2'でその場所にダウンロードする。

4.2 Server型モデルの適用

大学の周辺の店舗情報コンテンツを独立した各Server内に作成した(図5)。カテゴリは食事、病院、遊び、宿泊施設それぞれ別の組織によって管理されているものと仮定する。

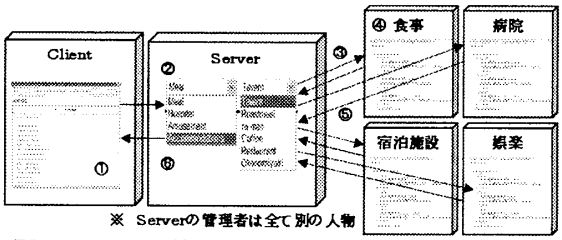

図5 Server型マッシュアップのシステム構成

(1)システムの流れ

① 利用者側からServerにリクエスト送信
② Server側のプルダウンリストでコンテンツの項目を選択ずる
③ プルダウンリストから選択されたコンテンツのあるその他のServerにAPIを送信
④ その他のSerever は受け取ったAPIに従い、選択されたコンテンツのあるXMLを返す
⑤ Server側では、返されたXMLをPHP内部のHTMLで定められている場所に表示する
⑥ Server側からHTMLをダウンロードする

(2)Server内プログラムの内部処理

WebAPIを呼び出す。その後、返されたXMLのHTTPヘッダを'explode'で削除し、SimpleXMLElementクラスで扱う用のデータ型へ変換する。変換したデータからXMLのタグを個別に取り出し、HTMLとJavaScriptを生成しClientへ返す。なお、この実験のプログラミングはPHPで行った。

4.3 Client+Server型モデルの適用

4.1と4.2を併せたシステムである(図6)。

Server型とClient型の複合でServerは4.2と同じようにXMLをAPIを使い取得した。その後、4.1の様にAjaxを使用した

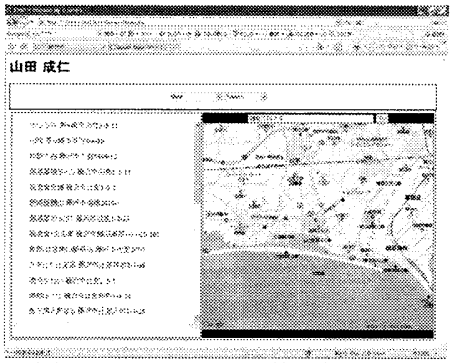

図6 Client+Server型の表示画面

5. 評価

(1)Client型の評価

アクセスマップを動的に表示することが可能になった。また、Clientで直接マッシュアップをするのでServer側のCPU負荷が低減できる。

(2)Server型の評価

WebServerから返されるデータの形式をWebAPIで統一しているので、WebServerの管理者が異なったり、内部が変更されてもマッシュアップを行うことが可能である。これにより複数の異なるサービスを粗結合できる。

(3)Client+Server型の評価

双方のマッシュアップの特徴を生かせ、その他のWebServerの情報をClient側から行う非同期通信や内容を動的に変更できるDynamicHTMLを使えるようになり、更なる付加情報を付け加えられる。 そのため、その他のServerで更新された情報などをいち早くマップや音楽といった形でも再現できるモデルである。

6. まとめ

マッシュアップ形態を3つにモデル化した。モデルごとに標準的な実装方法を定義した。各モデルとその標準実装方法を適用して実際のシステムを作成し、評価を行った。

参考文献

秋元 祐樹,PHP×Webサービス APIコネクションズ ソフトバンククリエイティブ, 2006