

# Web アプリケーションのための サービスマイグレーションの実現とその応用

## On a Service Migration for Web Application and its Applications

田辺 正喜<sup>†</sup> 大園 忠親<sup>†</sup> 伊藤 孝行<sup>‡</sup> 新谷 虎松<sup>†</sup>

Masaki Tanabe Tadachika Ozono Takayuki Ito Toramatsu Shintani

<sup>†</sup> 名古屋工業大学大学院情報工学専攻 <sup>‡</sup> 名古屋工業大学大学院産業戦略専攻

Dept. of Computer Science and Engineering, Nagoya Institute of Technology

Master course of Techno-Business Administration, Nagoya Institute of Technology

tanabe,ozono,itota,tora@ics.nitech.ac.jp

Web2.0 の普及とともに、近年多くの有益な Web アプリケーションが Web 上で無償公開されている。本論文では、Web アプリケーションを既存 Web ページに対して、Web ページの管理者ではなく閲覧者が Web アプリケーションの付加を行なうシステムの提案を行なう。Web アプリケーションとユーザの閲覧している Web ページとの仲介をシステムが行う事でユーザの Web ページから得る情報の質を向上させる事ができる。提案するシステムは付加した Web アプリケーションが既存ページに影響を与える事無く、Web ページの遷移に追従する。また、本論文ではシステムの応用に関して実装した試作システムである Web ページへの Web アプリケーション付加システム Web Glimpse 述べる。Web Glimpse は、ユーザが自由に Web アプリケーションを指定し拡張する事ができる。

## 1 はじめに

本論文では、実行環境を考量したプロセスマイグレーションとして、サービスマイグレーションの提案を行う。グリッドおよびユビキタス等の言葉が知名度を上げる中、計算機間をプログラムが行き来するマイグレーションにも注目が集まり、様々なマイグレーション手法が提案されている [3]。

本研究では、こういったマイグレーションの中でも、ユーザとの対話が必要なサービスに注目した。ここでのサービスとは、ユーザがシステムとの対話を行なう事で情報等の利益を得るシステムである。サービスでは、システムとユーザ間の対話を重視する必要がある。ユーザとの対話を考慮したマイグレーションを行う場合、マイグレーションする物は、プログラムのコードおよび実行状態が移動する従来のプロセスマイグレーションだけではなく、プロセスの実行されている様子を表す計算機内の環境をマイグレーションする必要がある。本研究ではこのように、プロセスだけではなく、プロセスの実行されている環境をプロセスマイグレーションに含めたマイグレーションをサービスマイグレーションと定義した。

本論文の構成を示す。第 2 章において、サービスマイグレーションを実現するにあたってのサービス

マイグレーションを定義する。第 3 章では、Web 上でのサービスマイグレーションの実現に関して述べる。第 4 章では、サービスマイグレーションの実装例としてサービスマイグレーションを利用した Web ページへの Web アプリケーション付加システム Web Glimpse に関して述べる。ここでの Web アプリケーションとは、HTTP リクエストを利用し、Web 辞書や Web 検索等のサービスを提供する Web ページおよびそのシステムをさす。第 5 章では、Web Glimpse のシステム構成に関して述べる。第 6 章では、Web Glimpse を例に Web 上でのサービスマイグレーションの手法に関して述べ、最後に、本論文をまとめる。

## 2 サービスマイグレーション

サービスマイグレーションとは、プロセスマイグレーションに、プロセスの動作する計算機の環境の情報を付加したものである。よって、プロセスマイグレーションの対象となるプロセスを  $P$  とし、サービスマイグレーションの対象となるサービスを  $S$  とすると、 $P, S$  は次の様に表せる。

$$P = \langle c, e \rangle$$

$$S = \langle P, v \rangle$$

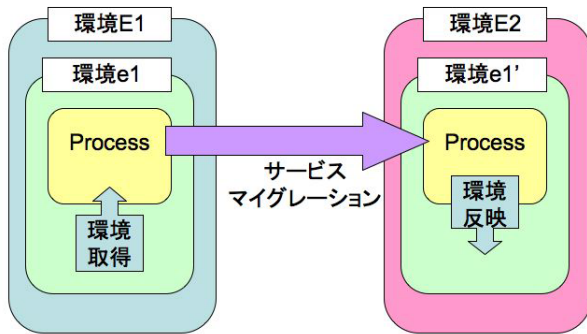


図 1: サービスマイグレーションモデル

ここで、 $c$  はプログラムのコード、 $e$  はプロセスの実行状態、 $v$  はプロセスの動作する計算機の実環境情報である。サービスマイグレーションのモデルを図 1 に示す。ある二つの環境を  $E_1, E_2$  とする。環境とはマイグレーションを行なうプロセスが実行されている環境全体を示す。例えば、環境にはプロセスが実行されているウィンドウ位置、およびプロセスが利用するファイルの情報、CPU のクロック数等も含まれる。環境  $E_1$  の環境情報  $v_1$  は、次の式で環境から取り出される。

$$v_1 = g(E_1)$$

$g$  は、プロセスが環境情報  $v$  を取得するために利用した環境取得関数である。 $E_1$  より  $g$  が取得可能な環境を  $e_1$  とする。 $e_1$  は以下の式を満たす。

$$e_1 \in E_1$$

$$g(e_1) = g(E_1)$$

プロセスが環境を適応するために利用した関数を環境反映関数  $f$  とする。 $v_1$  は、 $f$  の引数として利用される。よって、 $E_2$  に構築される環境  $e'_1$  は以下の式で表せる。

$$e'_1 = f(v_1)$$

$$e'_1 \in E_2$$

$e_1$  と  $e_1$  反映した環境である  $e'_1$  が等しいサービスマイグレーションをストロングサービスマイグレーションと定義する。逆に  $e_1$  と  $e_1$  反映した  $e'_1$  が等しくないサービスマイグレーションをウィークサービスマイグレーションと定義する。ストロングサービスマイグレーションは以下の式を満たす。

$$e_1 = f(g(e_1))$$

また、 $E_1$  の全ての要素をプロセスが取得可能な場合、つまり、 $v_1$  が  $E_1$  の全ての情報を表現することが可能なサービスマイグレーションを、「完全なサービス

表 1: サービスマイグレーションの分類

	$e_1 = e_2$	$e_1 \neq e_2$	$e_1 = \phi$
$E_1 = e_1$	A	B	
$E_1 \neq e_1$	C	D	E

- A: 完全なストロングサービスマイグレーション  
 B: 完全なウィークサービスマイグレーション  
 C: 不完全なストロングサービスマイグレーション  
 D: 不完全なウィークサービスマイグレーション  
 E: プロセスマイグレーション

マイグレーション」であると定義する。逆に、 $v_1$  が  $E_1$  の全ての情報を表現できないサービスマイグレーションを「不完全なサービスマイグレーション」として定義する。完全なサービスマイグレーションは以下の式を満たす。

$$E_1 = e_1$$

また、取得可能環境が存在しない場合、サービスマイグレーションはプロセスマイグレーションと同一である。

以上の定義に基づくことで、サービスマイグレーションは、表 1 を用いて 4 種類に表す事ができる。

### 3 Web アプリケーションのためのサービスマイグレーション

#### 3.1 Web 上でのサービスマイグレーション

通常、Web ページの遷移によって、Web ページ上の情報は初期化される。しかし、検索結果等のように初期化されたくない情報も存在する。例えば、Google マップ<sup>1</sup>等では、ユーザの閲覧した地点は保存しない限り Googlemap を再度読み込んだ時点で初期化されてしまう。Web 上でのサービスマイグレーションはユーザのページ遷移に左右されない。よって、永続的な Web ページ上でのプログラム動作を可能とする。Web 上でのサービスマイグレーションを定義する。プロセスは Web 上のプログラムである。 $pc$  は JavaScript になる。 $E$  は、Web ブラウザの設定および Web ブラウザが動作する環境となる。 $e$  は、JavaScript が取得可能な環境を表す。 $S, R$  は JavaScript のプログラムとなる。サービスマイグレーションのタイミングとしては、ユーザの Web ページ遷移とする。つまり、ユーザの Web ページ遷移のタイミングに合わせ、Web ページ上を移動するプログラムが Web 上でのサービスマイグレーションとなる。

<sup>1</sup><http://maps.google.co.jp/>

Web 上でのサービスマイグレーションに置いて、 $g$  はブラウザの動作する計算機環境の設定が取得できない。よって、不完全なサービスマイグレーションとなる。また、 $g$  によって、取り出した環境情報は、マイグレーション先の Web ページによって常に  $f$  が環境にすべてを適応できるとは限らない。なぜならば、ページによってフレーム構成が違い、複数ドメインにまたがる Web ページ等では、システムの JavaScript が影響を及ぼせる範囲に制限がかかるためである。また、javascript を用いて、Web ブラウザが動作する OS などの情報も取得する事ができるが、これらの情報は javascript を用いて反映する事ができない。以上より、Web 上でのサービスマイグレーションは「不完全なウィークサービスマイグレーション」となる。

### 3.2 Web 上でのサービスマイグレーションの実現

図 2 にサービスマイグレーションの動作行程を示す。ユーザが Bookmarklet を起動した際に、ユーザの閲覧しているウィンドウ (以下メインウィンドウとする) に対して、サービスマイグレーションの対象となるプログラム (以下本システムとする) を付加すると同時に、本システムは小さなウィンドウ (以下サブウィンドウとする) を作成する。本システムはサービスマイグレーションのための関数をもっている。作成したサブウィンドウはメインウィンドウの後ろに配置する事で、ユーザのブラウジングの妨げになる事は無い。サブウィンドウには、メインウィンドウと同様の URL を表示する。同一ドメインの Web ページを表示する事によって、サブウィンドウの HTML をメインウィンドウのサービスマイグレーション対象のプログラムが操作する事が可能になる。

メインウィンドウ上の本システムは、サブウィンドウの表示しているページに対してサービスマイグレーションのための JavaScript によって記述されたプログラムを付加する。メインウィンドウの閲覧ページが遷移する場合、遷移通知を受けた環境取得関数が Web ページの情報を収集し、サブウィンドウへ退避させる。その後、サブウィンドウの JavaScript プログラムが、メインウィンドウに表示される Web ページの監視を行う。メインウィンドウが次の閲覧するための Web ページを読み込むとサブウィンドウは読み込んだ Web ページに対して、JavaScript によるプログラムである本システムを付加する。付加された本システムは環境反映関数をもって、新しい Web ペ

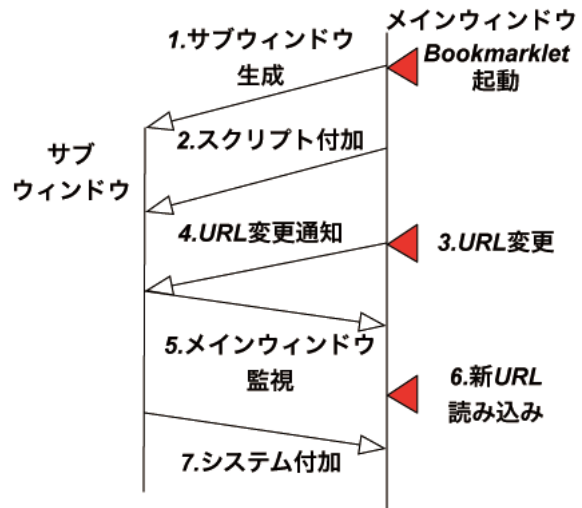


図 2: サービスマイグレーション動作行程

ジに対して、継続した環境をサービスマイグレーションの対象となるプログラムに提供する。以上によって本システムを用いて Web ページ上でのプログラムを永続化する事が可能になる。

本システムは JavaScript のセキュリティの制限のため、メインウィンドウとサブウィンドウが表示している Web ページが同一のドメインの場合のみ、JavaScript をメインウィンドウに付加する事ができる。よってサービスマイグレーションは同一ドメインに限られる。メインウィンドウとサブウィンドウのドメインが一致しない場合、サブウィンドウは自動的にウィンドウを閉じシステムの提供を終了する。

## 4 Web アプリケーション付加システム Web Glimpse

本論文で提案するサービスマイグレーションの実装例として作成した Web Glimpse に関して述べる。近年、AJAX を初めとする、Web2.0 が騒がれるようになり、Web 上で様々な Web アプリケーションが無料で公開されている。また、Google<sup>2</sup> や Amazon<sup>3</sup> 等では、自社の Web アプリケーションに対しての API を一般に提供している。Web 上ではこれらの Web アプリケーションを拡張したマッシュアップと呼ばれる種類の Web アプリケーションの公開も行なわれている。これらの Web アプリケーションは、他のアプリケーションが利用する事を前提とした Web サービスと違い、Web アプリケーションの置かれた Web ページ上

<sup>2</sup><http://google.co.jp/>

<sup>3</sup><http://amazon.co.jp>

のみでの利用が原則となっている。しかし、Web アプリケーションの入力および出力には他の Web ページと組み合わせる事でより良い情報が得られる事が少なくない。

一般のユーザにも可能な Web ページと Web アプリケーションの融合として、google ツールバー<sup>4</sup>等ブラウザの機能の拡張、Web サービスと同等の機能をもったブラウザの開発、エージェントを用いて Web サービスを連携させる [2] 研究、および Proxy サーバを利用して Web サービスを Web ページに付加する [1] などの研究が行われている。また、Firefox<sup>5</sup>等の一部のブラウザでは、ユーザスクリプトと呼ばれる JavaScript によるユーザの閲覧中の Web ページのカスタマイズが可能な機能が存在する。しかし、これらの研究では特別なプラグイン、ブラウザおよび、ブラウザの設定を行なう必要がある。

本論文では、サービスマイグレーションの実装例として、Web アプリケーション付加システム Web Glimpse[4] を開発した。Web Glimpse はユーザの日常的な Web ブラウジングと同様の操作によって、ユーザが自由に Web アプリケーションを現在閲覧中の Web ページ上から利用する事ができる。複数のウィンドウおよび、インナーフレームを利用する事によって、辞書およびデータベース検索などの Web アプリケーションが提供するサービスをユーザの閲覧している Web ページ上において実現する。Web Glimpse を利用する事で、ユーザは閲覧中の Web ページの情報を基に Web Glimpse に登録されたアプリケーションに対して HTTP リクエストを行ない、結果を Web 上のインナーフレームを用いて閲覧する事ができる。

## 5 Web Glimpse のシステム構成

Web Glimpse のシステム構成に関して述べる。Web Glimpse のシステム構成を図 3 に示す。また、本システムの動作画面を図 4 に示す。

Web Glimpse は、Web アプリケーションを利用する際に Web ページに付加したページエージェント、サーバ上のプロキシエージェント、および、サービスマイグレーションを提供するサブウィンドウから構成される。

ページエージェントとは、Web ページ上でユーザの閲覧中の Web ページと proxy エージェントとの仲介と Web アプリケーションの出力結果のコントロー

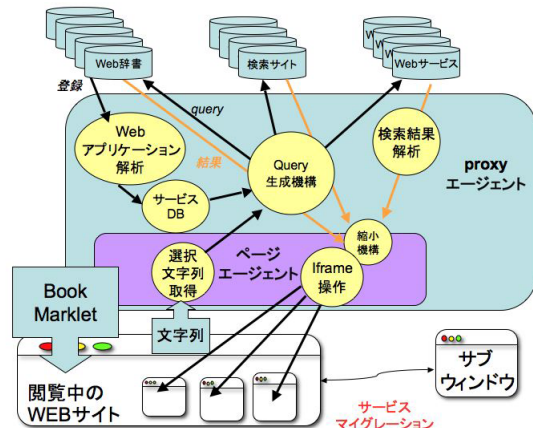


図 3: システム構成

ルを行う。ページエージェントは、ユーザの登録した Web アプリケーション、および前回起動時の状態を保存したデータベースのデータを反映し CGI によって作成される。Web Glimpse は Bookmarketlet によってユーザの閲覧している Web ページに対してページエージェントを付加する。ページエージェントが付加されると、閲覧中の Web ページの左上部にコントローラが表示される。また、初期起動として登録された Web アプリケーションの結果表示用のインナーフレームが表示される。コントローラには、検索用の HTTP リクエストを行なうテキストフィールドおよび Web Glimpse の設定用 Web ページを開くボタンが存在する。Web Glimpse では、閲覧中の Web ページ中の文字列を選択する事で選択文字列をサーバの行なう HTTP リクエストの query に利用するパラメータとする事ができる。テキストフィールドは、文字列の選択以外の方法で Web Glimpse に対してパラメータを指定した要求を行う事に利用される。

プロキシエージェントとは、サーバ上でページエージェントと Web アプリケーションとの仲介を行うエージェントであり、query の生成および Web アプリケーションの結果への縮小化機能の付加を行う。Web アプリケーションの結果表示にはインナーフレームを利用する。ページエージェントは、CGI によって生成され Web ページに付加される。結果表示用のインナーフレームに表示される個別の Web アプリケーションは、外部に公開されている Web アプリケーションを含む。よって、本システムのユーザは、設定用の HTML より既存の Web アプリケーションを登録する事によって、ユーザの望んだ拡張を行なう事ができる。

ページエージェントは Web ページ上の情報にアクセスし、必要な情報をサーバ側の proxy エージェント

<sup>4</sup><http://toolbar.google.com/>

<sup>5</sup><http://www.mozilla-japan.org/products/firefox/>





図 4: Web Glimpse 実行画面

に送信する．また，proxy エージェントより受け取った Web アプリケーションの結果をインナーフレームに表示する．ページエージェントは Web ページ上に生成したインナーフレームの動作管理を行う．

サーバ上の proxy エージェントは，ページエージェントより送信された情報から，設定された Web アプリケーションへの query を作成し，送信する．Web アプリケーションからの結果に対しては，縮小化機能を付加しページエージェントに送信する．縮小化機能を付加する目的としては，インナーフレームでは十分な描画領域を確保できない場合に対応するためである．表示する結果 HTML に対して HTML の描画縮小機能を付加することで，Web ページに表示可能な情報量を増やす事が可能になる．

Web Glimpse が利用する Web アプリケーションは，ユーザが URL の形で指定する．指定された URL に対して Web アプリケーション解析部が HTML を解析し，要求に必要な query のパラメータを入手，サービス DB に格納する．

出力には，インナーフレームを利用する以外に，通常の Web ページと同様ウィンドウを利用する事ができ，サービス指定時に利用形態を設定する．Web ページの場合，描画範囲にディスプレイ全体を利用できるため縮小化機構を埋め込む必要が無い．しかし，閲覧のためにはウィンドウの表示順を切り替えるなど一目での視認性にはインナーフレームの利用が効果的と考えられる．

## 6 Web Glimpse でのサービスマイグレーション

Web Glimpse でのサービスマイグレーションに関して述べる．

Web Glimpse は，サービスマイグレーションを JavaScript の付加によって，実現している．ユーザが Web サービスをページエージェントを利用して付加する場合，Web サービスを利用したい Web ページ毎にページエージェントを付加する必要がある．本研究で提案するシステムでは，サブウィンドウを利用する事でドメイン内での Web サービスのマイグレーションを可能にする．マイグレーションの対象となるページエージェントは JavaScript によって記述され，Bookmarklet を利用して Web ページに付加される．

Web Glimpse が  $g$  を用いて入手する情報には，現在サービスを提供している Web アプリケーションの種類に加え，提供に利用したインナーフレームの位置，大きさ，現在の表示状態が取得される．次に， $f$  はこれらの情報を Web ページに適応するが，この際ユーザ閲覧中の Web ページ内に他のドメインがフレーム形式で存在した場合，全ての環境情報を Web ページに適応する事ができなくなる．またユーザの閲覧している Web ページが完全に別ドメインの場合，サービスの提供そのものが失敗する．

## 7 おわりに

本研究では，環境に注目したプロセスマイグレーションとして，サービスマイグレーションの提案を行なった．サービスマイグレーションは，プロセスの周りの環境を同時にマイグレーションの対象とする事で，Web アプリケーションなどユーザとの対話が必要なシステムにおいて，効果を発揮する．サービスマイグレーションの試作として，Web を対象としたサービスマイグレーションの実装を行なった．また，本手法の実装例として Web ページへの Web アプリケーション付加システム Web Glimpse の試作を行った．

## 参考文献

- [1] 坂本 暁, 北 英彦, 高瀬 治彦, 林 照峯, "proxy 技術を利用した Web サービスのためのプラットフォームの提案", 情報処理学会研究報告, Vol.2002, No.031, 2002.
- [2] 石川 冬樹, 田原 康之, 吉岡 信和, 本位田 真一, "Web サービス連携のためのモバイルエージェント動作記述", 情報処理学会論文誌, Vol.45, No.6, 2004.
- [3] 矢野 浩邦, 佐藤 記代子, 前田 誠司, "実計算機間の OS マイグレーション技術の設計", 情報処理学会研究報告, Vol.2003 No.18, 2003.
- [4] 田辺 正喜, 大園 忠親, 伊藤 孝行, 新谷 虎松, "ユーザの閲覧ページに合わせた BookMarklet によるドメインへの Web サービス付加システム," 第 68 回情報処理学会全国大会論文集, 2006.