

Webサービスのダイナミック連携技術

Dynamic Coordination of Web Services

あらまし

インターネット上の情報やサービスが増え便利になってきたが、数が増えすぎたために人間が一つ一つを取り扱うことが困難になってきた。このため、システムが人間に代わって情報やサービスを取りまとめ、人間に分かりやすい形で提示することが重要になってきている。しかし、ネットワーク上で変化しつづける情報やサービスにシステムが自動的に対応するようなダイナミックな連携の実現のためには、従来技術だけでは不十分で、新たな技術開発が必要である。

本稿では、ダイナミック連携について、富士通研究所で取り組んでいるテーマについて紹介する。最初に、基礎となるWebサービスやセマンティックWebの概要を説明した後、パーソナルサービス連携、ポートレット連携、ユビキタスサービス連携、情報連携というそれぞれの立場からダイナミック連携について述べる。

Abstract

The amount of useful information and number of services on the Internet have reached the point where people cannot easily use or handle the information and services. As a result, it is now necessary to develop a system that can coordinate information and services on the Internet and present them to users in user-friendly forms. However, conventional technologies are not advanced enough to realize a system that automatically performs dynamic coordination of the Internet's ever-changing information and services. We should therefore develop new, more advanced technologies. Fujitsu Laboratories is working in several technology areas to develop a system for dynamic coordination of Internet information and services. This paper gives an overview of the Web services and the Semantic Web that will form the basis of dynamic coordination. Then, this paper describes dynamic coordination from various aspects such as personal service coordination, portlet coordination, ubiquitous coordination, and information coordination.



湯原雅信 (ゆはら まさのぶ)
ITコア研究所 所属
現在、サービス連携およびユビキタスミドルウェアの研究開発に従事。



松本達郎 (まつもと たつろう)
ITコア研究所Web&IPサービス研究部 所属
現在、ユビキタスミドルウェアの研究開発に従事。



井形伸之 (いがた のぶゆき)
ITメディア研究所言語処理研究部 所属
現在、セマンティックWeb技術を利用した知識管理システムの研究開発に従事。

ま え が き

様々な情報やサービスがネットワーク上で複雑にリンクし合うようになった。この様子はまさにWeb（クモの巣）という名前にふさわしい。Webの世界を成功させた基本概念は、すべてのことを処理する一つの巨大なサーバを用意するのではなく、独立に管理され常に化する多数のサーバを疎に結合することにより世界を構成するということであった。

このようなWebの考え方は今後も発展していくと思われるが、新たな問題が顕在化してきた。Web上のサービスや情報の数が増えすぎた結果、人間の手では十分に活用できなくなってきたのである。そこで、今後のシステムは、人間に代わって増えすぎたサービスや情報を連携し、統合する必要がある。

本稿では、変化するサービスや情報のダイナミックな連携を目指して進めている富士通研究所における研究開発を紹介する。最初に、ダイナミックな連携の基礎となるWebサービスとセマンティックWebについて説明する。続いて、パーソナライゼーションに重点を置いたWebサービス連携とポータルにおけるWebサービス連携について述べる。その後、サービスや情報と人間との間を結ぶユビキタスサービス連携について説明する。最後に、セマンティックWebを応用した情報連携について述べる。

ダイナミック連携を支える標準化技術

ダイナミックなサービス連携をねらった技術の一つにWebサービス^{(1),(2)}がある。従来のWWWは人間の目で見えるためのサービスであったのに対し、Webサービスは機械が利用するためのサービスである。システムは、Webサービスを組み合わせ高レベルなサービスを構築する。とくに、設計時に存在していなかったサービスが次々と出現しても、システムがそれらのサービスをダイナミックに連携させることにより、変化に適應できるような世界を目指している。Webサービスの基本的な標準は、サービスを登録・検索するためのUDDI（Universal Description, Discovery and Integration）⁽³⁾ サービスに関する記述を行うWSDL（Web Services Description Language）⁽⁴⁾ メッセージ交換プロトコルとしてのSOAP（Simple Object Access Protocol）⁽⁵⁾である。これらのうちダイナミック連

携にとってもっとも本質的なのはWSDLである。サービスに関する情報をWSDLという形で電子的に入手できるようになって初めて、機械がサービスの仕様をダイナミックに理解できるようになる。

Webのもう一つの発展形として、セマンティックWeb⁽⁶⁾がある。セマンティックWebは、Webの発明者であるT. Berners-Leeが提唱するビジョンであり、Web上の情報を機械が理解できる形で表現することにより、ソフトウェアが意味的な（=セマンティックな）処理を自動的に行う世界を目指している。意味的な処理を行うには、例えば「本」と「Book」という文字列が同じ概念であるとか、「文庫」が「本」の一種であるというようなことをソフトウェアが理解できなければならない。W3Cが推進しているセマンティックWeb関連の標準化で、最も基礎的な標準がRDF（Resource Description Framework）⁽⁷⁾である。RDFは、リソース、属性、値の三つ組みにより、リソース（=データや概念）に関する情報や関係を表現する。

Webサービスについては、産業界が中心になりすぐ役に立つところから検討しているのに対し、セマンティックWebの方は、大学を含めて人工知能や知識表現など様々な分野から検討している。しかし、WebサービスとセマンティックWebはいろいろな面で相補的な関係にある。Webサービスを使って賢く連携するには、サービスがやり取りする情報やサービスに関する意味を機械が理解する必要があるし、セマンティックWebで情報を活用しようとするとき情報をやり取りするサービスのことを考える必要がある。生い立ちが違うため、しばらくの間、両者は刺激し合いつつも並行して進展しそうである。

パーソナルサービス連携

ダイナミック連携に向けた試みとして、Webサービスを利用したパーソナルサービス連携を最初に紹介する。例えば引越して住所を変更する場合、ユーザは市役所で転出、転入手続きを行わなければならないほか、自動車免許や車の登録、電話や電気、クレジットカードや定期購読雑誌の住所変更など、いろいろな手続きを行わなければならない。e-Japan構想が進展すると、これらの手続きはWeb上で行えるようになるはずである。しかし、単にWeb化しただけでは、ユーザは自分に必要な手続きを広大なイ

インターネットの中から探し出し、順番を考えながら、一つ一つ手続きを進めなければならない。そこで、パーソナルサービス連携は、ユーザが引越しのような特定の目的を達成したいときに、その目的に関係する多数のサービスの中から、そのユーザに必要なサービス群を選択し、それらを適切な順番で自動実行する。

著者らは、各手続きがWebサービスで提供されるようになることを想定し、住所変更手続きをワンストップで行うプロトタイプシステムを作成した(図-1)。

ここでは、ワンストップサービスをユーザに応じてパーソナライズするために、ユーザに関する情報と、サービス(手続き)に関する情報をダイナミックに参照し、そのユーザに必要なサービスだけを選択している。ユーザに関する情報は、ユーザプロフィールに蓄積されたデータと、その場で入力されるデータを利用する。WSDLの形で入手したサービスに関する情報は、信用できるものだけをサービスプロフィール(プライベートUDDI)に登録して利用する。さらに、WSDLが示す各サービスの入出力情報から、サービス間の依存関係を求め、手続きの実行順序を決めている。このようにして、全体の実行手順であるシナリオをそのユーザ専用

生成することができるようになった。

ポートレット連携

図-1のようなダイナミック連携を行う際の一つの問題は、末端のサービスが人間とのやり取りを必要としている場合には、パーソナルサービス連携の中で、やり取りのための画面をサービスに依存してあらかじめ実装しておかなければならないことである。新しいサービスが出現したときに実装の手間をかけずにダイナミックに連携させるためには、末端のサービスが画面を生成することを最初から想定してシステム全体を設計する必要がある。これには、ポートレットという考え方が役に立つ。

ポートレットとは、Webの部分画面を生成するサービス部品である。とくに図-2のようにWebサービスのインタフェースで実現されるものをリモートポートレットと呼び、OASISのWSRP(Web Services for Remote Portlets)®技術委員会で標準化中である。WSRPでは、Webサービスの仕様に準拠した上で、画面情報の交換方法を規定している。WSRPに関連して、ローカルなポートレットのAPIについても、Javaの標準化団体JCPのJSR-168®という専門委員会で標準化中である。著者らは、Fujitsu Laboratories of Americaとも協力してポー

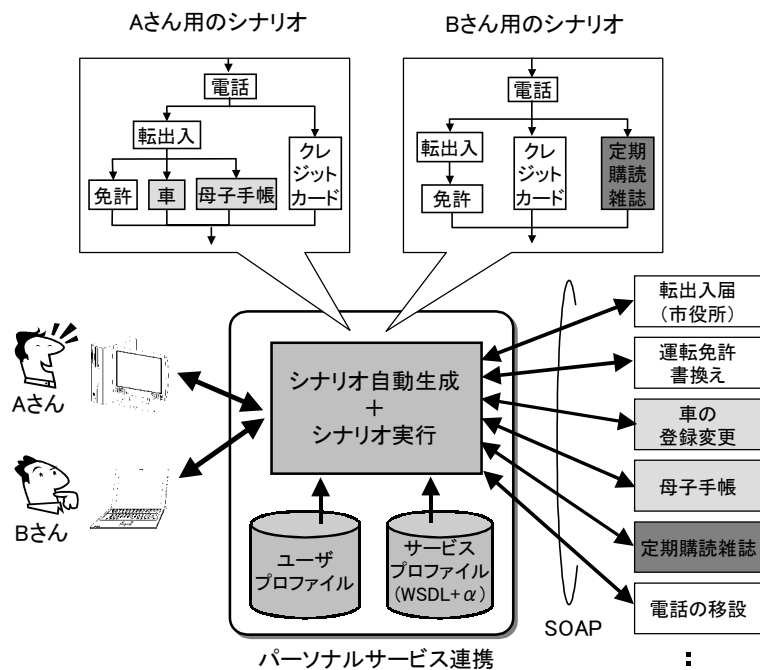


図-1 シナリオ自動生成
Fig.1-Automatic generation of scenarios.

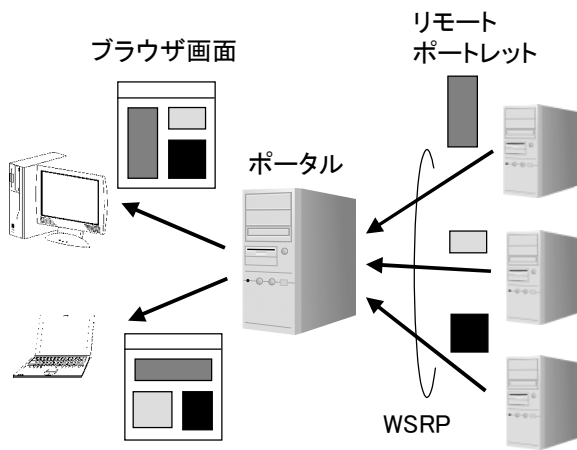


図-2 ポータルとリモートポートレット
Fig.2-Portal and remote portlets.

ポートレットの両方の標準化に参加し、検討中の仕様に基づいたプロトタイプを富士通のポータル製品であるInterstage Portalworks向けに開発した。

WSRPの第1版は、早期の標準化のため、ブラウザ画面内にポートレットの画面を埋め込む単純な機能に限定されるが、図-1のようなダイナミックな連携に必要な機能が今後検討されていくものと思われる。著者らは標準化議論に先行して、ブラウザ上でのユーザの指示（ドラッグ&ドロップなど）によりポートレット間で複合データをやり取りする方法を検討している。

ユビキタスサービス連携

ダイナミック連携に向けたもう一つの試みとして、ユビキタス環境における人とサービスのダイナミックな連携を紹介する。携帯電話、PDA（Personal Digital Assistant）、車載情報機器などの普及に加えて、定額制PHS、第3世代携帯電話、無線LANにより、モバイル環境における通信インフラのブロードバンド化、低料金化が進みつつある。これにより、様々な端末で、いつでも、どこでも、インターネットに接続し、あらゆるサービスを利用できるユビキタス環境が現実のものとなってきた。このような環境下で、有用なサービスを簡単に実現するためには、人が置かれた環境や状況を意識せず、人からサービスへ、サービスから人へのアクセスを可能にするミドルウェアが重要となってくる。

ユビキタス環境下では、人は様々な場所で、様々な通信メディア・端末を用いてインターネットへ接

続する。そのため、その場所、通信端末、ユーザ状況に応じたタイムリでパーソナライズされたコンテンツが提供されることが望まれる。

また、人がサービス側の変化を常に監視するのは不可能であるため、サービス側が自律的にユーザ状況に応じてサービスをPushすることが必要である。

これらを実現するには、第一に、サービス側が人の状態（利用中の/利用可能な通信メディア・端末、Busy/運転中、位置情報など）をリアルタイムで把握できなければならない。この場合、個々のサービスが毎回状態を問い合わせるのではなく、ある条件に合致する状態変化があったときにリアルタイムでサービスに状態変化を通知する仕組みが重要となる。

第二に、サービスがユーザ状態変化を契機として何らかのコンテンツを提供する場合、ユーザが利用可能な通信メディア・端末に応じた最適なコンテンツを提供する必要がある。この場合、サービス側が各ユーザの端末を意識してサービスを記述することは、コンテンツ作成のコスト増大を招くため、サービス側には通信メディア・端末を意識させない仕組みが重要となる。

そこで、これら二つの機構を兼ね備え、ユビキタス環境下での人とサービスのインタラクションを容易に実現する枠組みをユビキタスミドルと名付け、先に述べたパーソナルサービス連携と組み合わせたシステムを試作した。

ユビキタスミドルの構成を図-3に示す。以下に、ユビキタスミドルの詳細について述べる。ユビキタスミドルは、以下の三つのモジュールから構成される。

(1) メディアGW（Gateway）

通信メディアの違いを吸収し、ユーザ状態を抽象化する。また、Push依頼を各通信メディアに配送する。

(2) ユーザ状態管理

ユーザ状態の収集、状態の通知先・通知条件の管理、状態通知を行うモジュールで構成する。ユーザ状態はXML形式で管理し、通知予約は、XPath⁽¹⁰⁾に準拠した形式で行う。

(3) Push制御

サービスからのPush依頼を受け、ユーザ状態に応じて、メディア選択、コンテンツ変換を行う。

サービスは、通知条件を登録しておくことでユー

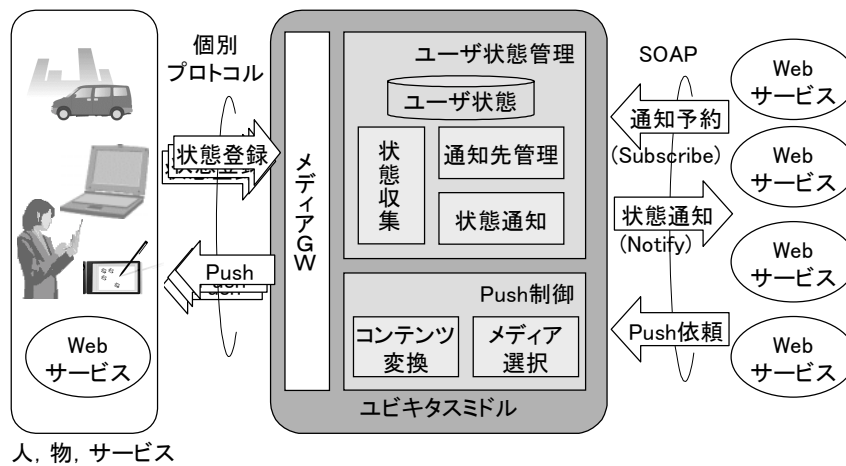


図-3 ユビキタスマドル
Fig.3-Ubiquitous Middleware.

ザ状態管理から状態の更新通知を受け、ユーザ状態に応じたサービスの提供を行う。また、ユーザ状態の変化を契機としてサービス主導でユーザへサービスを提供する場合は、Push制御に対してコンテンツ配信依頼を行うことで、ユーザの端末種別を意識することなくユーザへのサービス提供が可能になる。これらの機能を利用して、以下のようなサービスへの適用が考えられる。

(1) エリア情報Push

店舗近くにいる人という条件で通知予約を行い、条件にマッチした人にセール情報をPushする。

(2) シームレスなナビゲーション

自動車運転中は車載端末にナビゲーション情報をPushし、降車後は携帯機器へ継続的にナビゲーション情報をPushする。

(3) マルチモーダルコンテンツ提供

アクセスしてきた端末能力に応じて、コンテンツを音声、画像、テキストやそれぞれを組み合わせたマルチモーダルコンテンツに変換し提供する。

情報連携

Webサービスがネットワーク上の様々なサービスを連携するのに対し、セマンティックWebは、ネットワーク上の様々な情報(リソース)を連携する試みと言える。バラバラに存在している情報を連携することで、個々に見ては分からない関係も見えるようになる。また、情報をダイナミックに連携することで、激変するネットワーク上の情報に対しても、常に最新の情報を提供することが可能に

なる。

本章では、セマンティックWebの要素技術をイントラネット環境に適用した情報連携アプリケーションの一例として、特定の技術分野に詳しい専門家を検索するシステム(以下、KnowWhoシステム)を紹介する。

技術分野の知識は、一人の専門家だけでなく、複数の専門家によって構築されることがある。また、専門家の持つ知識は日々変化するものであり、同時に専門家同士のつながりも変化する。このような情報をいかに収集し、情報の日々の変化に追従するかがKnowWhoシステムのポイントとなる。

今回試作したKnowWhoシステムでは、セマンティックWebの要素技術であるRDFを用いて、異種アプリケーションにおける個人や個人に関係するリソースのメタデータを記述し、それらを相互連携することで、KnowWhoシステムの問題点を解決している。具体的には、富士通研究所内の研究員データベース(LDAPサーバ)、ネットワーク上で公開されている技術文書(Webページ、Word、PowerPointなど)、グループウェア(スケジュール管理ソフト)の情報をRDFにより意味的に連携し、個人の所有知識から所属部署の所有知識を構築したり、所属部署を越えた専門家同士のつながりを検出したりしている。

本KnowWhoシステムで用いているメタデータの例を図-4に示す。同図では、研究員「木村卓也」を中心に、3種類のメタデータを連携している。それぞれのメタデータは、情報抽出技術により自動生成

の分野での人間関係を推測している。

以上，KnowWhoシステムを用いて情報連携アプリケーションの一例を紹介した。このような情報連携技術は，既存アプリケーションの情報をシームレスに連携するEAI（Enterprise Application Integration）にも応用することができる。

む す び

本稿では，WebサービスやセマンティックWeb技術を活用したダイナミック連携について，パーソナルサービス連携，ポートレット連携，ユビキタスサービス連携，情報連携というそれぞれの観点から，富士通研究所で研究開発中の技術について紹介した。ネットワーク上の情報やサービスはこれからも増え続けるため，それらをダイナミックに連携させる技術は今後ますます重要になる。著者らは，本稿で述べた技術の実用化を図りつつ，さらに先を見た研究開発を進めていく予定である。

参 考 文 献

- (1) 飯田一朗：XMLの新しい潮流 - パーソナライゼーション技術 - . *FUJITSU*, Vol.53, No.3, p.250-254 (2002).
- (2) Webサービス (W3C).
<http://www.w3.org/2002/ws/>
- (3) UDDI (OASIS).
<http://www.uddi.org/>
- (4) WSDL (W3C).
<http://www.w3.org/2002/ws/desc/>
- (5) SOAP (W3C).
<http://www.w3.org/2000/xp/Group/>
- (6) セマンティックWeb (W3C).
<http://www.w3.org/2001/sw/>
- (7) RDF (W3C).
<http://www.w3.org/RDF>
- (8) WSRP (OASIS).
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp
- (9) JSR-168 (JCP).
<http://www.jcp.org/en/jsr/detail?id=168>
- (10) XPath . (W3C).
<http://www.w3.org/TR/xpath>