

WebサービスとSOAシステム

文：柴田賀昭 Shibata Yoshiaki メタフロンティア代表兼チーフコンサルタント

今回は、FIMSの活動母体であるEBUやAMWAなる団体の紹介と、FIMS発足までの経緯に加え、SOAの特長としてのサービスの「疎結合」に基づくシステム構築やそれらを「オーケストレーション」することによる柔軟なアプリケーションの実現、そして作業進捗の「見える化」などを紹介。ただ、特に技術関係の読者からは、これらは概念としては理解できるものの具体的な実装イメージが全く湧かないといった声もある。そこで今回は、SOAのベースとなるWebサービスとSOAシステムの技術エッセンスを解説していただく。

はじめに

概念としてのSOAは比較的広く知られているものの、いわゆるSOAシステムと呼ばれているものが実際にどう実現されているかは、意外と知られていないのが実状ではなからうか。筆者も初めてSOAシステムの話を知った時、なぜそれが柔軟性や拡張容易性を実現できるのかをイメージできず、インターネットで調べたり、専門家に質問したりして、それなりの納得感を得るまでに相当苦労した記憶がある。

今回は筆者のこのような経験を踏まえ、FIMSないしは一般のSOA技術の根幹をなすWebサービスとSOAシステムについて、その技術エッセンスを紹介したい。なお、その趣旨に則り、専門用語の定義を含めた厳密性は問わないこととする。またサービスの議論においては、SOA的にはビジネス的視点などが問われるが、技術的には単に所定の機能を定められた入出力方法に従って提供しているに過ぎず、その意味で本連載ではサービスをソフトウェア工学におけるサブルーチンと同様に捉えていることに留意されたい。

Webサービス

(1) WWW動作のおさらい

Webサービスとは、一言でいえばWWW (World Wide Web) を、その要素技術を活用してさまざまなサービスを提供すべく「汎用化」したものである。既に日々の業務あるいは生活の一部になった感のあるWWWだが、改めてここで用いられている要素技術を紹介すべく、まずはWWW動作のおさらいから始めることとする。

WWWでは、IE (Internet Explorer) などのいわゆる汎用WWWブラウザ (以下、ブラウザと呼ぶ) において、そのアドレスバーに所定のURL (Uniform Resource Locator) を指定

することで、所望の情報がブラウザに画面表示される。

ここでURLが示すのは、ネットワークの「あちら側」に接続されたいわゆるWebサーバのホスト名 (および、それに格納された文書ファイル名) であり、ブラウザは接続先Webサーバに所望の文書ファイルを要求し、これを手元にダウンロードしている。そしてこの一連のやり取りを担う通信方法をHTTP (Hyper Text Transfer Protocol) と呼ぶ。また、入手した文書ファイルには所望の情報をどのようにブラウザに画面表示するかといった指示情報が記載されているが、そこで用いられる言語からこの文書のことをHTML (Hyper Text Markup Language) 文書と呼ぶ (例えばIE 9の場合、「表示 (V)」メニューの「ソース (C)」を選択すれば、ブラウザがWebサーバから実際に入手したHTML文書の中身を見ることができる)。

これよりWWW動作とは、結局は「所定のWebサーバ内に格納された所望のHTML文書をURLで指定し、これをHTTPにて手元にダウンロードし、ブラウザで画面表示させる」ということとなる。

(2) Webサービスとは

さて、このようなWWW動作において、Webサーバは単にそれが保有するHTML文書をブラウザからの要求に従って提供しているに過ぎない。これに対し、Webサーバにそれ以上の機能、例えば膨大なデータを管理させたり、大規模計算をさせたり、さらにはWebカメラのパンチルトのようにWebサーバ自体に物理的な動作をさせたいといった要求が自然発生した。これがWebサービスの始まりである。

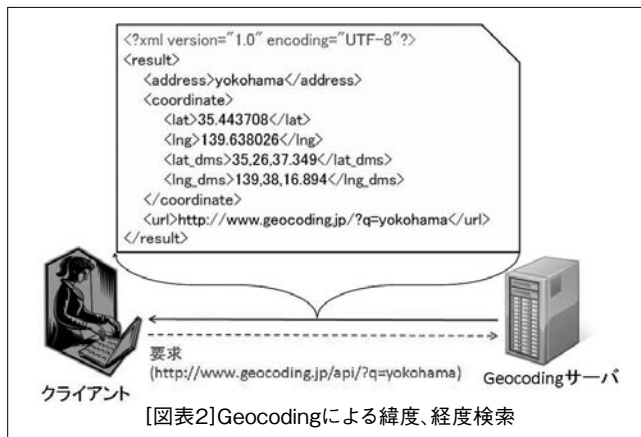
ここでURLやHTTPについては、WWWの爆発的な成功によって既にその安定性や拡張容易性が実証済みなので、これを転用することとした。ただHTMLは、あくまでブラウザでの画面表示を目的とした言語であるから、これに替わるものとして新たにXML (eXtensible Markup Language) を開発し、Webサーバ側の動作結果をXML文書で返すこととした。また、ほとんどの機能はその動作において入力情報を必

要とするが、これにはURLの一部に入力情報を埋め込む方法 (REST) や、あるいはクライアント側にて入力情報を含むXML文書を作成し、これをWebサーバ側へアップロードする方法 (SOAP) などが開発された。[図表1]にWWWとWebサービスの比較を示す。

	WWW	Webサービス
目的	情報の閲覧	検索、データ処理、制御、他
要求方法	URL	URL (REST)、あるいはURL+XML文書 (SOAP)
結果 (戻り値)	HTML文書	XML文書

[図表1]wwwとwebサービスの比較

具体例として、住所や都市名からその緯度、経度が検索できるGeocoding (<http://www.geocoding.jp/>) なるWebサービスを紹介する。[図表2]は、Geocodingを用いて横浜 (Yokohama) の緯度、経度を検索した様子を模式的に示したものである。ここでクライアントがGeocodingサーバに対し、「q=yokohama」なる入力情報を含めたURLにてサービス提供を要求すると、その結果がXML文書として返される (IE9の場合、このXML文書がそのままブラウザに表示される)。そしてその中には、横浜の緯度、経度情報がそれぞれ「<lat>…</lat>」及び「<lng>…</lng>」なるかたちで格納されている。すなわち、横浜は北緯35.443708度、東経139.638026度に存在することがわかる。



ところで、実際のアプリケーション (以下、アプリと称す) では、ブラウザはこうにして得られた情報を、例えば別途入手した地図データ上の横浜の位置に旗を立てるなどといったかたちで活用する。これらの処理は、ブラウザに前もってプラグインされた補助アプリで実行される他、Webサーバがその命令コード自体をHTML文書に埋め込んでブラウザに提供し、ブラウザがそれを解釈、実行するJavaScriptのような方式もある。

(3) Webサービスの使い方

さて、上記の説明からGeocoding自体は、住所や都市名を入力すればその緯度、経度を戻すサブルーチンの類とみなすことができる。ただ、このままだとCやJavaでいうサブルーチンに比べてどうも使い勝手が悪い。そこでWebサービスでは、後述するような対策が施されている。

そもそもネットワークの「あちら側」に能動的で高性能な処理系を共有し、手元のクライアントから作業を依頼するといっ

た動作は、古くはメインフレームをタイムシェアリングした時代からの永遠の課題である。そしてその開発のポイントは、いかにしてそれを楽に使えるようにするか、ということであった。

ここで改めてWebサービスの動作をみると、まずは手元で入力情報をURL埋め込み、あるいはXML文書として準備し、次にこれをHTTPにてWebサーバに受け渡し、そして当該Webサーバが所定の作業の結果出力するXML文書を取り込むといった流れとなる。これより、やり取りするメッセージの構造 (スキーマ)、メッセージ交換パターン (基本はRequest/Response)、利用するプロトコル (原則HTTP)、そして対象Webサーバのホスト名といった情報が与えられれば、そのWebサービスを使うためのプログラミングが始められる。

そしてすべてのWebサービスでは、そのような利用のための詳細情報がWSDL (Web Service Description Language) 文書として提供され、さらにそれを入力として当該Webサービスを利用するためのソースコードを自動合成するWSDLコンパイラなるツールも準備されている。

これよりWebサービスの利用においては、それに併せて提供されるWSDL文書を手し、それを利用するためのソースコードをWSDLコンパイラで合成すれば、あとはあたかもCやJavaにおけるサブルーチンのように、そのWebサービスを利用することができる。このようにネットワークの「あちら側」の機能があたかも従来のサブルーチンのように利用可能な開発環境が整っていることも、Webサービスのもう一つの特長と言えよう。

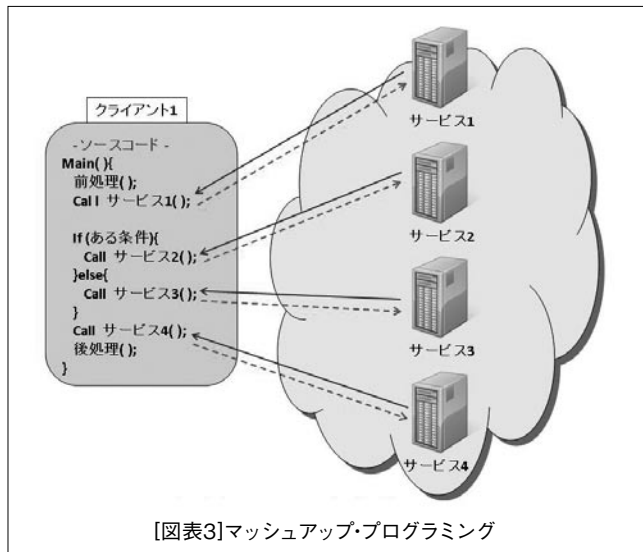
SOA システム

(1) SOAシステムとは

次にSOAシステムの技術エッセンスを紹介する。ここでの説明はいわゆる「SOAとは何か?」といった概念論ではなく、具体的なアプリを実現する「実装手段」としてのSOAシステムであることに注意されたい。

さて、前項でWebサービスを紹介したが、実はインターネット上には既に膨大な数のさまざまなWebサービスが存在しており、これらを連携させて所望のアプリを開発するマッシュアップ・プログラミングと呼ばれる手法が広まっている。先述したように、Webサービスの特長の一つは、それがあたかも従来のサブルーチンのように使えることであるから、その基本的な方法は、目的を達成するためのWebサービスの連携をソースコードとして記述するということになる。

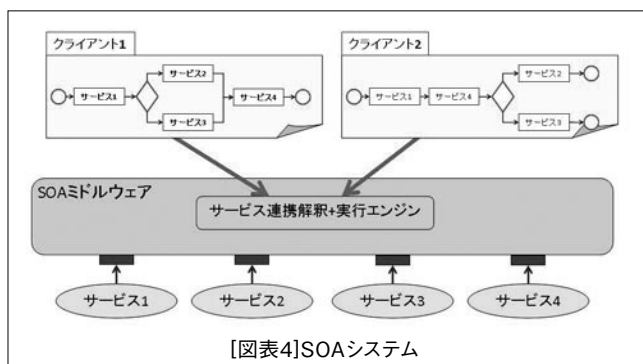
いま、[図表3]のような基本的なアプリ開発方法の問題点を考えてみる。そのようなアプリの開発では、単にソースコードを書くだけでなく、これをコンピュータ上で実行可能な形式 (オブジェクトコード) へ変換 (コンパイル) の上、利用可能なように準備 (デプロイ) する必要もある。そして、当該アプリが所望の動作をしなかった場合は、その旨ソースコードを修正し、コンパイル、デプロイといった作業を繰り返すこととなる。



また、そもそもこのような一連の作業は素人が簡単にできることではなく、それに必要なプログラミングの知識と開発スキルをもったソフトウェア技術者の仕事となる。

さらに[図表3]の「クライアント1」は、例えば「サービス3」を用いているが、それ以外の他のクライアントもまた「サービス3」を利用することも想定される。そして、それらのクライアントは互いの動作を知らないことから、例えばある瞬間において「サービス3」へのアクセスが集中し、それをダウンさせてしまうといった事態を引き起こす可能性もある。

さて、まずは後者の対応であるが、それにはクライアントからの要求と実際のサービスの実行をうまく調停する仲介者を両者の間に立ててやればよい。SOAシステムは、まさにこのような考え方に則ってアプリを実現するものである。



すなわち[図表4]に示すように、SOAシステムでは、クライアントとサービスとの間にSOAミドルウェアと呼ばれる仲介者を設置し、クライアントはこのSOAミドルウェアを介してサービスを要求、実行することで先の問題を解決している。このように、クライアントとサービスを直結せずSOAミドルウェアを介して間接的に結び付ける方法を「疎結合」と呼ぶ。

(2) SOAシステムの動作

以下、[図表4]に従ってSOAシステムの動作を説明する。まずSOAシステムは、それにて利用可能なWebサービスをSOAミドルウェアに登録することから始まる。具体的には先述した

WSDL文書をSOAミドルウェアに入力することで、SOAミドルウェアがその必要に応じて所望のWebサービスを実行できるような状態にする。

次にクライアントであるが、各クライアントは目的を達成するためのWebサービスの連携情報(これはしばしば「ワークフロー」と称される)を記述し、これをSOAミドルウェアに提供する。ここでSOAシステムとして特長的なのは、このWebサービスの連携情報を、先述したソースコードでなく、[図表4]中に示したフローチャートとして記述できることにある。

そしてSOAミドルウェアは、与えられたWebサービス連携情報に基づいて所定のWebサービスを実行し、その結果を次のWebサービスへと引き渡し、最後のWebサービスで得られた結果を依頼元のクライアントに返すことでアプリとしての一連の処理を完了する。このようにSOAシステムにおいてWebサービスを連携させることを、特に「オーケストレーション」と呼ぶ。

ここでWebサービス間の入出力は、基本的にはXML文書のやり取りで実現されるが、各サービスがその入出力にどのようなXML文書を求めるかは、それが提供するWSDL文書で規定されている。これよりSOAミドルウェアは、これらXML文書のWebサービス間の受け渡し制御に加え、前のWebサービスが出力したXML文書を次のWebサービスでうまく取り込めるよう、タグ名などXML文書の変換処理も同時に実施している。

ところで、先述したフローチャートによるサービス連携情報の記述は、その直感的な表現力のおかげで、ソフトウェア技術者でなくとも対応可能である。実際、エンタープライズシステム分野においてこれを準備するのは、いわゆるビジネスアナリストと呼ばれるワークフロー分析の専門家である。もちろん、それに基づきSOAシステム上で実現されるアプリが所望の動作をしなかった場合は、入力であるフローチャートをその旨、修正することとなる。ただ、従来の方法で求められたコンパイル、デプロイといった作業が不要になるので、アプリの修正は非常に簡単に実施することができる。そしてこれらもまた、柔軟性や拡張容易性といったSOAシステムの特長に大きく寄与している。

(3) SOAシステムの「見える化」

最後にSOAシステムの「見える化」について説明したい。

[図表4]から推測されるように、登録済みWebサービスの利用頻度や稼働状況、またすべてのクライアントからのサービスの連携情報(アプリ)や、各アプリの作業進捗など、SOAシステム上で実行されるアプリに関するすべての情報はSOAミドルウェアに集約されている。SOAシステムがサポートする「ダッシュボード」とは、まさにそういった情報を必要に応じてリアルタイムで表示する機能であり、これによってシステム上の全アプリの作業進捗や各サービスの稼働状況がリアルタイムで把握できる。

これより一連の作業の進捗管理やボトルネックの早期検出とその対応などといったシステムの全体最適化が図れるが、これもまたSOAシステムのもう一つの重要な特長と言えよう。