

WEBサービス技術を用いた広域分散システム 管理の構想

三浦 健次郎, 相浦 利治, 細川 武彦, 扇谷 篤志, 茂木 強

三菱電機株式会社 情報技術総合研究所

要旨

近年 Web サービスを利用した, サービス指向アーキテクチャに基づくシステム構築が進み始めている. Web サービスは, 利用者が直接利用する Web アプリケーションと異なり, システム間呼び出しとして利用されるサービスであり, その管理要件も Web アプリケーションの要件とは異なる. 本稿では, 現状十分に議論が行われていない WEB サービス管理に必要な要件を整理し, Web サービスさらにはネットワーク・サーバ機器管理を管理ゲートウェイにより統合管理する方式を提案する. また, 試作したサービス管理システムについて報告する.

A Distributed Management System based on Service Oriented Architecture.

Kenjiro Miura, Toshiharu Aiura, Takehiko Hosokawa,
Atsushi Ogiya, Tsuyoshi Motegi

Information Technology R & D Center, Mitsubishi Electric Corporation

Abstract

Recently, In the field of information system integration, Service-oriented architecture (SOA) technology gains in popularity. SOA is the system architecture concept in which a set of services are assembled and reused to rapidly adapt to business needs. But A management system for SOA is not well discussed. In this paper, we organize the requirements of SOA management and propose the integrated SOA management system by using SOA gateways.

1. はじめに

近年WEBサービス技術を利用したSOA(Service Oriented Architecture)に基づいた広域分散システムの構築が始まっている.

しかし, 現状はこのようなシステムのサービス化に対応した運用管理方法については十分に検討されているとはいえない.

そこで本稿では, WEBサービス技術を用いて構築された広域分散システムの管理手法について検討し, 試作したサービス管理システムについて報告する.

本稿の構成は次の通りである. 2章でWEBサービスを利用したシステムの概要を整理し, 3章でこ

れらの運用管理上の課題を検討・整理する. 4章では, これらの課題を解決するための提案を行うとともに, 著者らが行ったゲートウェイ方式によるプロトタイプ管理システムについて報告する. 最後に5章でまとめと今後の課題について述べる.

2. WEBサービスシステム 概要

本章では, まず管理対象となるWebサービスのシステム上の位置づけについて概要を整理する.

2. 1. WEBサービスの論理的な位置づけ

SOAシステムは論理モデルでは次ページ図1のような5階層のモデルで説明されている[1,2].

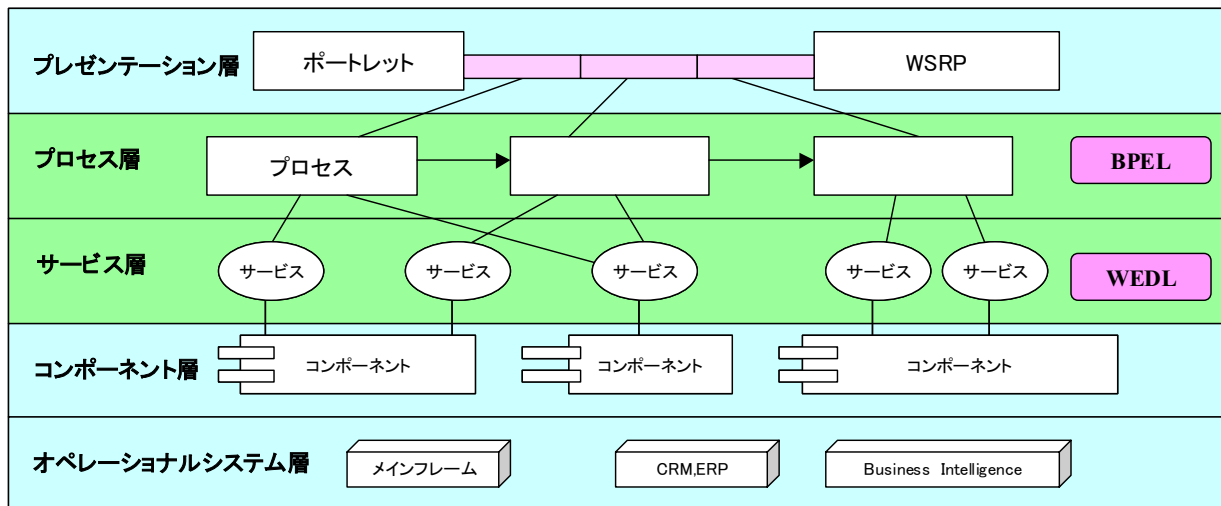


図1 SOAの層構造

Web サービスは SOA の層構造上、図1のサービス層に位置づけられる。実装されたビジネスロジックを広域ネットワーク（インターネット）上で呼び出すことができるように、サービスインターフェースを定義し、他システムに公開する。

サービスインターフェースはサービス記述言語 WSDL[3]で定義され、データは SOAP[4]メッセージとして交換される。転送プロトコルは、現状 HTTP[5]が使われていることが多いが、WSDL では Web サービスエンドポイント（Web サービスメッセージの宛先(URI)、以後「エンドポイント」と略）で実装されている機能と、HTTP等のトランスポートプロトコルは分離して定義する。従って WSDL 定義の binding パートを書き換えることにより、HTTP 以外のプロトコルに入れ替えることが可能である。

なお、公開したサービスを登録、検索する仕組みとして UDDI[6]が標準化されている。

図1のもうひとつのポイントはプロセス層である。サービス層で定義・公開された各ビジネスロジックをプロセス層で組み合わせてフローを形成し、業務システムを構築する考え方が SOA の特徴である。プロセス定義の標準化活動は WSBPEL[7]仕様の検討が OASIS で行われている。公開されたサービスを再利用してシステムを構築するのでシステムを迅速に構築でき、システムの変更は BPEL フローの変更で対応できるので、変化に強い新しいシステム構築手法として期待されている。

2. 2. WEBサービス物理的な位置づけ

SOAシステムの構成要素である個々のWEBサー

ビスは社内のみならず、社外で提供されている WEB サービスも利用することを想定している。社外のサービスとしては、例えば株価、天気、乗り換え経路表、書籍データ等の有意なデータの配信を主とするサービス、又は認証、決済、与信証明、検索などの専門性の高い機能を提供するサービス等が考えられる。

社外システムを利用する形態としては、既に ASP (Application Service Provider)の WEB アプリケーションを利用する形態が確立されている。ASP型の利用形態では完結した WEB アプリケーションを直接利用者が使用するのに対し、WEB サービスは業務プロセスの一機能としてシステムからサービスが呼び出されるという点が異なり、管理面でも考慮が必要である。

3. WEBサービスの運用管理の課題

次に WEB サービス技術を用いる SOA システムの運用管理上の課題を整理する。

3. 1. サービス層の管理の課題

まず、サービス層の管理について検討する。

①管理粒度についての検討

WEB アプリケーション管理手法として、クライアント型プローブ装置が、監視対象アプリケーションの URL に定期的に HTTP 要求を発行し、その応答の有無や応答速度により障害や性能を管理する手法がある。一般に WEB アプリケーションでは URL とアプリケーション機能が一対一に対応しているためこのような管理手法が有効である。

しかし、WEB サービスではサービスのエンドポイントで提供される機能 (wsdl:operation) は複数になりうるので、単なるエンドポイントへのポーリングでは十分な管理ができない。同一エンドポイントで提供される機能が複数である場合、呼び出した機能が違えば、特有のエラーの発生や応答性能の違いがでるからである。

具体的にはエンドポイントで提供されるサービスの操作(wsdl:operation)単位での管理と、関連する操作をひとまとめにした portType(wsdl:portType)単位での管理が必要になると考えられる。

②トランスポートプロトコルについての検討

前述の通り WSDL ではメッセージの交換に使うトランスポートプロトコルを HTTP に限定しているわけではない。従って、この点でもエンドポイントに対する HTTP のダミートランザクションに対する応答から障害や性能を判断することが難しいといえる。

③遅延許容度についての検討

WEB サービスはシステム間の呼び出しであり、利用者が直接利用する WEB アプリケーションに比べ、応答遅延が大きいメッセージでも正常と判断すべき場合がありえることを考慮する必要がある[8]。この点でも HTTP のダミートランザクションによる管理が難しいといえる。

④サービス品質指標の標準化

各サービスの品質を評価するための指標値の標準化も必要になる。サービス品質評価の指標値がばらばらであると、同種のサービスの品質を比較することが難しくなるからである。

⑤他の管理プロトコルの利用

アプリケーションの管理は上記のように外部からのダミートランザクションによる管理方法の他に、例えば Java アプリケーションであれば JMX[9]を利用してシステムリソースやアプリケーションの管理属性を参照・設定する仕組みがある。このような既存の管理フレームワークを WEB サービスのフレームワークで利用可能にする仕組みを検討する必要がある。

3. 2. プロセス層の管理の課題

前述の通り SOA では Web サービスを組み合わせる新しい業務サービス (これを以下「複合 Web サービス」と呼ぶ) を構築することができる。

複合 Web サービスの状態は、各 Web サービスの管理ステータスの集約ステータスとして表現されるが、個々の Web サービスのステータスをどのように集約して表現するかは課題である。

3. 3. 責任分解点の明確化の課題

SOA によるシステム構築では、社外の Web サービスを取り込んで業務システムを構築する場合がある。

社外サービスを自システムの一部として取り込む場合の問題は、社外サービスの制御では制限事項が出てくるという点である。社外のサービスはサービス層のインタフェースが責任分解点であり、サービス提供者の内部障害に起因する障害については各サービス提供者で一次対応してもらうしかないからである。

また、障害のように判断基準が明確な場合は良いが、性能劣化のように基準が決まっていない問題の管理は予め明確な SLA の取り決めが必要である。

社外サービスについては採用前に評価を十分に行い、重要な社外サービスについては、サービス障害を迅速に切り分けられるようにする必要がある。さらに代替サービスへの系切替など設計上の考慮も重要となる。

一方社内の Web サービスについては、サービス障害や性能劣化時にコンポーネント層以下の部分を含めて、根本原因を解析し修正できるような仕組みが必要となる。

原因の追求には、単なる Web サービスの障害・性能監視だけではなく、Web サービスが動作しているサーバ・ネットワーク機器等の管理システムとの連携が重要となってくる。

3. 4. その他の課題、検討事項

多くの企業では、ネットワーク管理、システム管理等の運用管理システムを既に構築している。従って、これらの既存運用管理システムの有効利用も重要な検討事項である。

現状、運用管理システム間の連携はベンダー固有のソリューションによっているか、SNMP-TRAP 等による非常に弱い連携とになっている場合が多い。

ベンダー固有のソリューションでは、マルチドメインのマルチベンダー環境では適用範囲が限定されてしまうという欠点がある。

一方 SNMP-TRAP による連携も、ネットワークセキュリティ上 SNMP-TRAP が Firewall を通過する設定にすることは好ましくないため、マルチドメインの相互接続環境においては現実的な連携方式とはいえない。また、SNMP-TRAP 連携では、構造型のデータなど複雑なデータの扱いが煩雑になるという問題もある。

4. 広域分散運用管理システムの実装提案

本章では WEB サービスを管理する広域分散システムの標準化動向と実装提案を行い、著者らが行ったプロトタイプ実装を紹介する。

4. 1. Web サービス関連の管理仕様の標準化

WEB サービスを利用した運用管理システムとしては、OSMIC-MAXI[10]やWSDM[11]が提案・標準化されている。

MAXI は INTAP が推進している運用管理の仕様で、運用管理システム間のイベント通知と稼働監視情報交換の標準化を目指したものである。主に国内のベンダーが参加しており各社の運用管理システムを相互接続することに主眼がある。MAXI ではトランスポートには HTTP(又は SMTP)を想定し、オブジェクトのモデリングは DMTF で検討されている CIM[12]の利用を想定している。

WSDM は OASIS で検討されている分散運用管理の標準で、Web サービスの分散管理や Web サービスを利用した分散運用管理の標準を検討している。WEB サービスを利用した運用管理のための仕様 (MUWS: Management Using Web Services) と Web サービスを管理するための仕様 (MOWS: Management of Web Services) の 2 つの仕様から構成されている。本実装では主に WSDM 仕様を参考にして実装している。

4. 2. 広域分散管理システムの提案と課題解決

3章で整理した WEB サービスの運用管理上の課題をふまえ、図2に示すように、Web サービスが動作するドメイン内部でのみ収集可能な管理情報をドメイン外部に管理用 Web サービスとして公開する、

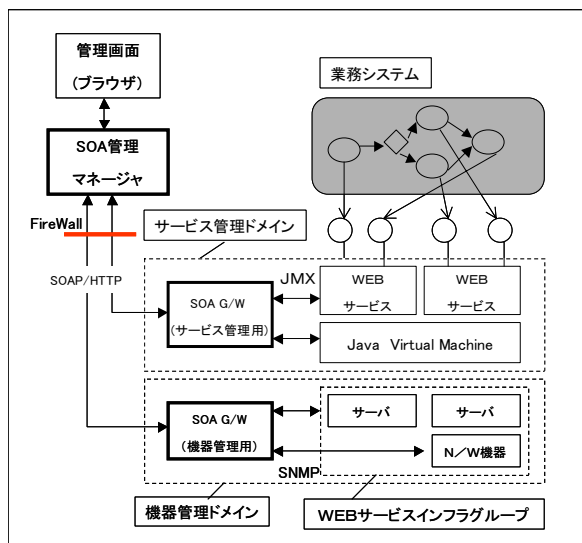


図2 ゲートウェイ方式による管理

表1 管理データ例

管理属性	説明
Operational status	ステータス Available, Partially Available, Unavailable, Unknown の4状態
NumberOfRequests	要求総数
NumberOfFailedRequests	Fault 応答が返った要求数
NumberOfSuccessfulRequests	Fault 応答以外が返った要求数
ServiceTime	サービス継続時間
MaxResponseTime	最大応答時間
LastResponseTime	前回リクエスト時の応答時間

管理ゲートウェイ(以下「SOA G/W」)によって広域分散管理する方式を提案する。

本管理システムは、全体を統合管理する「SOA 管理マネージャ」、SOA G/W として WEB サービスを管理する「SOA G/W (サービス管理用)」とネットワーク機器やサーバ等の機器を管理する「SOA G/W (機器管理用)」の2種類のゲートウェイで構成する。

各 SOA G/W は、SOA 管理マネージャから管理情報収集要求を受信すると SOAP/XML メッセージを分析し、SOA G/W 内に蓄積したドメイン内部管理用の管理データベース又はドメイン内の管理プロトコルを利用して指定された情報を収集し、SOAP メッセージの応答を返す。各 SOA G/W は各ドメイン内部の管理マネージャの機能も持っている。

4. 3. プロトタイプ実装画面例

(1) サービス管理用 SOA G/W

図3にサービス管理用 SOA G/W 用の管理画面例を示す。SOA 管理マネージャが提供する管理



図3 サービス管理画面例

画面用の URL をブラウザでアクセスすることでこの画面が表示される。

左ペインのツリーの最上位には管理ゲートウェイの一覧が表示される。ゲートウェイを選択すると、選択したゲートウェイを管理用エンドポイントとする Web サービスの一覧が表示される。表示された管理対象 Web サービスの中から管理したい Web サービスを選択すると、表 1 に示したステータス情報 (Operational Status) や性能指標 (Metrics) が右ペインに表示される。

なお管理対象の Web サービスは Java で実装されていることを想定しており、SOA G/W (サービス管理用) は内部的には Java アプリケーション管理用フレームワーク JMX を使って管理情報を収集している。同様に JMX でサービスが動作している Java Virtual Machine(JVM)の管理情報も収集可能としており、ツリーをドリルダウンすることで Web サービスが動作する JVM のバージョン、ヒープメモリ、スレッド数等が参照可能となっている (図 4)。

(2) 機器管理用 SOA G/W

サーバやネットワーク機器等の管理を行う SOA G/W (機器管理用) は開発済みの SNMP マネージャ [13] を利用して実装した。既存の管理マネージャに、Web サービスアクセス用の機器管理用インタフェースを追加実装している。

サービスとの関連付けは WEB サービスの動作に関連する機器をグループ化し、作成したグループと対応する WEB サービスを XML 定義で関連付けることにより行う。サービス管理用 SOA G/W の管理画面から機器管理用 SOA G/W の管理画面をリンク参照できるように実装しており、サービス管理用 G/W でのサービス障害検知を契機として、関連付けられた機器管理用の画面で機器の状態を



図 4 サービスインフラ管理画面例

確認することができる。

企業内に既に導入されている同種の管理マネージャも本手法のように管理用 Web サービスインタフェースを追加開発することで、他のドメインで必要な情報を効率良く公開・参照できるようになると考えられる。

5. まとめ

本稿では、Web サービスを利用して構築するシステムの運用管理上の課題を整理した。

また、課題を解決する方式として管理ゲートウェイ方式を提案し、各管理ドメインの管理情報を Web サービスで疎結合統合し、Web サービスやそれを動作させるインフラの管理情報を統合管理する試作システムを紹介した。

この試作システムによって、ゲートウェイ方式でも WSDM 標準に準拠した形で実装可能なこと、既存の管理マネージャを有効利用した管理用の Web サービスを実装可能なことを示すことができた。

本稿では、3章で整理した課題のうちプロセス層の管理について具体的な解決方法を示せていない。これについては今後検討をしていく予定である。また、管理リソース間の関連付けや依存関係管理の管理、SOA G/W 間の関係管理についてはもう少し整理して統一的な定義方式を検討する必要があると考えている。

また、相互接続性を高めるには各種 WebServices 標準に準拠する [11,14-16] が必要だが完全に準拠した実装になっているわけではない。またメッセージの暗号化等セキュリティなどについての考慮も必要であろう。

今後これらの課題を継続検討し、本システムの改良を行っていく予定である。

参考文献

- 1) Ali arsanjani, "Service-oriented modeling and architecture", <http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/>
- 2) 牧野友紀, "ビジネス環境と実装システムを繋ぐ BPM と SOA", 情報処理, 46 巻 1 号 p63, 2005 年 1 月
- 3) E.Christensen, F.Curbera, G.Meredith, S.Weerawarana, "Web Services Description Language(WSDL)1.1", <http://www.w3.org/TR/wsdl>, March 2001.
- 4) Simple Object Access Protocol(SOAP),

- <http://www.w3.org/TR/soap/>
- 5) Hypertext Transfer Protocol(HTTP),
<http://www.w3.org/Protocols/>
 - 6) Universal Description, Discovery, and
Integration(UDDI), <http://www.uddi.org/>
 - 7) OASIS Web Services Business Process
Execution Language (WSBPEL) TC,
[http://www.oasis-open.org/committees/tc_home.
php?wg_abbrev=wsbpel](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel)
 - 8) 青木,柴山 他,”WEB サービスコンピューテー
ング9章”,電子情報通信学会,2005年2月
 - 9) Sun Microsystems, Java Management
Extensions(JMX),[http://java.sun.com/products/
JavaManagement/](http://java.sun.com/products/JavaManagement/)
 - 10) INTAP,OSMIC-MAXI,
[http://www.net.intap.or.jp/INTAP/osmic/busine
ss/MAXI212.pdf](http://www.net.intap.or.jp/INTAP/osmic/business/MAXI212.pdf)
 - 11) OASIS Web Services Distributed
Management (WSDM) TC,
[http://www.oasis-open.org/committees/tc_home.
php?wg_abbrev=wsdm](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm)
 - 12) DMTF, Common Information Model(CIM)
Standards, <http://www.dmtf.org/standards/cim/>
 - 13) 三浦, 扇谷, 虎渡,”1CD Linux を利用したネッ
トワークエッジ監視装の開発”, 情報処理学会, 第
122回マルチメディア通信と分散処理研究会,
2005-DPS-122(51), 2005
 - 14) W3C, Web Service Addressing,
<http://www.w3.org/Submission/ws-addressing/>
 - 15) OASIS Web Services Resource Framework
(WSRF)TC,[http://www.oasis-open.org/committe
es/tc_home.php?wg_abbrev=wsrf](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf)
 - 16) Web Services Interoperability
Organization,<http://www.ws-i.org>