

4 ビジネスグリッド^{*}関連技術動向 と標準化活動

特集

ビジネスグリッドコンピューティング

福井恵右 >>>> 富士通(株) : kfukui@labs.fujitsu.com

岸本光弘 >>>> 富士通(株) : hiro.kishimoto@jp.fujitsu.com

佐川暢俊 >>>> (株)日立製作所 : sagawa@sdl.hitachi.co.jp

中田登志之 >> 日本電気(株) : t-nakata@cw.jp.nec.com

森 拓也 >>>> 日本電気(株) : moritaku@bx.jp.nec.com

舘村純一 >>>> NEC ラボラトリーズ・アメリカ : tatemura@sv.nec-labs.com

グリッドの技術領域では、管理対象となる IT 機器の拡大や、投入する業務情報の相互運用性が重要となる。その中で、多様な IT 機器を組み合わせたシステム全体の複雑さを軽減するためには、IT リソースのモデル化／仮想化も重要である。また、業務とそれを実行する IT 機器の割付けを動的・柔軟に定義できるような枠組みは、日々変化するビジネス環境への IT システムの追従を容易にし、システム管理の複雑さを整理する。これらの技術の概要と動向、今後を、その歴史的経緯、主要な国際標準化団体である GGF^{☆1} と OASIS における標準化活動との関連をふまえて解説する。

グリッドと標準化動向、歴史的経緯

計算機がネットワークに接続され、IP などの標準的な通信プロトコルが整備されるにつれ、ある計算機上のサービスから他の計算機のサービス呼び出して利用したいという分散処理のニーズが高まった。このような要求を実現するための方法として、関数呼び出しのリモート化である RPC (Remote Procedure Call) が普及し、さらに、分散オブジェクトの概念と組み合わせるサービスのディスカバリやライフサイクル管理を可能とする CORBA の標準化が OMG (Object Management Group) を中心として行われてきた。これらは、分散コンピューティングの基本的機能として現在でも多くのアプリケーションで用いられている。

一方、1990 年代後半から、大学や研究機関を中心に IT リソースを相互に融通し利用効率を上げようとするグリッドの考え方が広まってきた。グリッドでは、サイエンス分野における余剰計算リソースの有効活用という実践的なニーズが先行したため、概念の整理や標準化に先立っていくつかのシステムが実装され、各々の地域や技術領域で利用されるにいたった。その代表例としては、Globus Alliance が開発している Globus Toolkit や

Unicore プロジェクトにて開発している Unicore などを挙げるができる。

グリッドの概念や各種システムが一定の普及を見るとそれらの間の概念整理やインタフェース統一、相互接続への要求が高まり、地域ごとにワークグループやフォーラムなどの情報交換組織が設立された。米国中心の Grid Forum、ヨーロッパ中心の European Grid、アジアパシフィック中心の APGrid などがその例である。1998 年暮れに、ハイパフォーマンスコンピューティング関連の著名学会である Supercomputing においてこれらグリッド関連団体を集めた会議が開催され、その結果を受けて 2001 年には世界的な普及促進を目指す標準化団体として GGF (Global Grid Forum) が組織されるにいたった。

初期の GGF は主に既存のグリッド団体やユーザ間の情報交換が行われたが、2002 年開催の会議 GGF4 にて Web サービスの考え方を基盤とした新しいグリッドアーキテクチャである OGSA (Open Grid Services Architecture) アーキテクチャが提案されると、以降の GGF はその詳細化と仕様策定が一つの大きなミッションとなっていった。この活動は、従来からのグリッドの主要ターゲットであったサイエンス分野に加え、期待の高まりつつあったビジネス分野へのグリッドの適用をスコープにいれており、ビジネスコンピューティングで多く用いられる J2EE (Java2 Enterprise Edition) や .NET とも親和性のよいアーキテクチャとなっていたため、広

☆1 後述するように、2006 年 7 月に GGF (Global Grid Forum) は OGF (Open Grid Forum) と改称したが、本稿では混乱を避けるため GGF と記述を統一する。

く IT 産業界の注目を集めるにいたった。本アーキテクチャは、インターネット上の IT サービス利用形態として標準的な位置を占める Web サービスの基盤の上に、グリッドサービスの配備、実行、相互接続、グリッドの利用する IT 機器 (IT リソース) の記述や制御、関連するセキュリティメカニズムなどを定義する網羅的な仕様を目指している。ビジネスグリッドプロジェクトもこの活動への参画と知見のインプットによる標準化推進を一つの主要なミッションと定め、主体的な活動を行ってきた。2004 年には、OGSA が GGF の「フラッグシップアーキテクチャ」であると宣言され、今日なおそのカバー範囲の拡大が精力的に進められている。なお GGF は、企業におけるグリッド関連ソリューションの開発および導入促進を目的とするコンソーシアム EGA (Enterprise Grid Alliance) と統合し、OGF (Open Grid Forum) として 2006 年 7 月に拡大再出発することとなっている。

一方、OGSA の基盤となる Web サービスの機構についても、当初は OGSi (Open Grid Service Infrastructure) という名称のもと GGF にて標準化を推進する方向であった。しかし、この領域は W3C (World Wide Web Consortium) や OASIS (Organization for the Advancement of Structured Information Standards) などの団体が中心となっており、標準化を進めていることもあり、活動の重複を避け全体の効率を上げる意味から、それらの標準化団体との連携の下で標準化を進める方向が GGF13 にて決定された。W3C では主に Web サービスの通信方法や接続インタフェースなど下位レイヤの標準を定め、OASIS では Web サービスのメカニズムを用いた上位レベルサービスの標準化を進めている。OGSA では、Web サービスをリソース制御や高信頼通信、ジョブ制御のための基本機構として利用することから、特に OASIS との連携を重要視し、相互に仕様の確認やフィードバックを行いつつ整合性のとれた標準化を推進している。ビジネスグリッドプロジェクトにおいても、GGF に加えて OASIS にても積極的な役割を果たし、両者の活動紹介、役割分担などの橋渡しを行うほか、各種接続実験への参画により仕様の実証確認を行うなど標準化の加速に貢献している。

また、グリッドにおいては、利用される IT リソースが共通の方法でモデル化、記述されることが望ましい。IT システムの運用管理の観点から IT リソースの記述モデルの標準化を進めている団体として DMTF (Distributed Management Task Force) があり、ストレージ分野の標準化団体 SNIA (Storage Network Industry Association) は先行的にその成果をリソースの記述に活用している。OGSA はこれらの活動とも連携しつつ、IT リソース全体の整合性のとれた標準化作業の加速を図っ

ている。

標準化の位置付けとねらい

グリッドシステムの理想をかたちのある技術に展開し、相互にオープンな連携をしていくには国際標準仕様が不可欠である。IT リソースの管理技術はすでにベンダによる独自の開発が進んでいるが、標準仕様に基づいたリソースの仮想化を進めることにより、業務プログラムと IT リソースの関連付けをより柔軟に行い、マルチベンダ環境での IT リソースの有効活用をさらに進めることが期待される。この観点からビジネスグリッドプロジェクトでは散在するグリッドシステムの相互接続に必要なインタフェース部分の標準化に重点的に取り組み、また、この相互接続にかかわる実装を進めることで、仕様の普及を目指してきた。具体的には、全体アーキテクチャとして GGF のフラッグシップアーキテクチャである OGSA に準拠し、特に重要な以下の技術領域に重点をおいて活動してきた。

- 複数サイトにまたがったマルチベンダ環境での業務のリソース要求記述と業務実行依頼インタフェース・交渉の技術
- 複数サイトにまたがったマルチベンダ環境で業務の配備を可能とする技術
- マルチベンダ環境のリソースを统一的に管理・制御する技術

このほかにも、セキュリティや接続性の高信頼化の技術はビジネスには欠かせない要素である。これらを推進するにあたって GGF および OASIS において複数の作業部会に参加、もしくは作業部会創設時点の共同議長として貢献してきた (図 -1 参照^{☆2})。

この活動により、作業部会単位の個別な取り組みだけでなく、仕様相互の連携による全体アーキテクチャが形作られている。これらの仕様群は、グリッドシステムが究極的に目指す、管理組織の境界を越えた IT リソース活用への道筋を切り開くものであり、世界規模に分散する IT リソースを束ねる実用的標準仕様群が徐々に出現しつつあるといえる。ビジネスグリッドにかかわる標

☆☆☆
 ☆² OGSA : Open Grid Services Architecture
 JSDDL : Job Submission Description Language
 GRAAP : Grid Resource Allocation Agreement Protocol
 CDDLM : Configuration Description, Deployment, and Lifecycle Management
 ACS : Application Contents Service
 WSDM : Web Services Distributed Management
 WSRM : Web Services Reliable Messaging
 SEC : Security, RM : Resource Management
 WG : Working Group
 DT : Design Team
 TC : Technical Committee

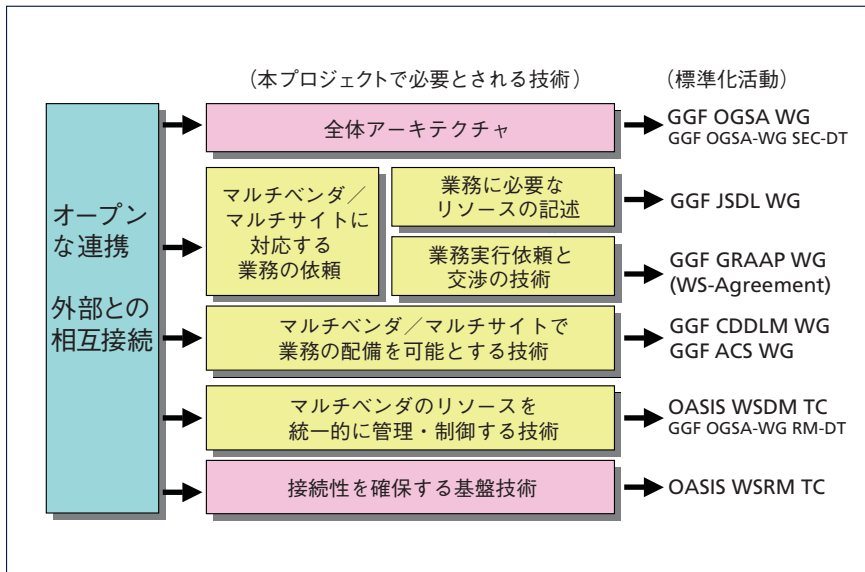


図-1 標準化分野

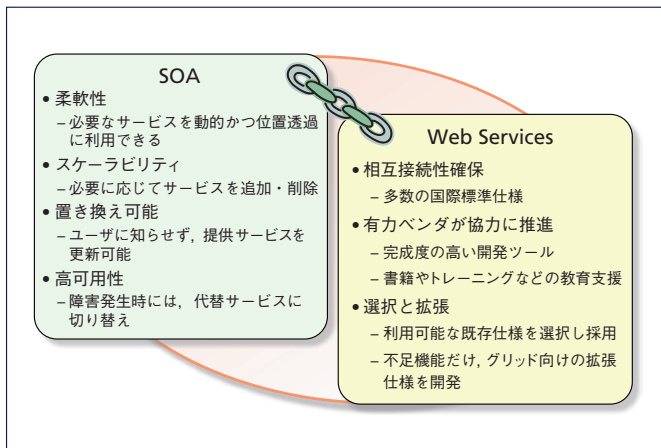


図-2 SOA と Web サービスの利点

準化はグリッドコンピューティングの究極像のみでなく、今日のビジネスにも貢献することで、長期・短期両面でITシステムの進歩を加速していくねらいがある。

標準化の各分野

本章では標準化の各分野ごとに仕様・技術の紹介とその動向、および、ビジネスグリッドプロジェクトでの取り組みを述べる。

❖ 全体アーキテクチャ

OGSA アーキテクチャ¹⁾は、次世代のグリッドコンピューティング・ミドルウェア・アーキテクチャを描く国際標準である。この標準は、SOA (Service Oriented Architecture) と、Web サービス (Web Services) の技術を採用している (図-2 参照)。SOA は、分散アプリ

ケーションの構築技術として知られているが、グリッドのミドルウェアに対しても複数の利点を提供する。まず、SOA が定義するサービスを、グリッドが必要とする機能の論理的なビューにとらえ、比較的大きな単位での機能分割と考えることができる。そして、これらサービスを再利用したり複数を組み合わせたりすることによって、さらに、新しい機能の提供が可能になる。さらにSOA は実際の機能の実装方法を利用者から隠蔽し、機能実装に用いたハードウェア、OS、言語に関係なく、利用者がサービスを利用できる。このようなSOA の特徴は、広域でかつ異機種分散環境というグリッドに

とって必須の性質である。

さらに、OGSA では、実際のインタフェース定義にあたっては、XML (Extensible Markup Language)、SOAP (Simple Object Access Protocol)、WSDL (Web Services Description Language)、Web Services Addressing などに代表される、Web サービス技術を広範に採用している。Web サービス技術は、現在、主要なITベンダが次世代の分散アプリケーションの構築技術として強力に推進しているもので、強力な開発ツールやさまざまな教育マテリアル等が容易に入手できる。さらに、分散システムの構築技術として、Web サービス技術とグリッド技術には共通点が多いので、OGSA ではグリッドコンピューティング用に似て非なる解決技術を開発するのではなく、Web サービス技術をなるべく利用することにしている。

一般的に、Web サービス仕様は柔軟性と拡張性を備えており、OGSA の要求するさまざまな用途に利用することができる。しかし一方、実際にグリッドシステム間の相互接続性を実現しようとする、これらの柔軟性と拡張性が逆に相互接続性の障害となる恐れがある。そこで、グリッドシステム間で相互接続性を確保するため、WSRF (Web Services Resource Framework) を含むWeb サービス標準仕様群をどのように使うかを厳密に規定するOGSA プロファイル (OGSA profile) を導入した。プロファイルの考え方は、WS-I (Web Service Interoperability)^{☆3}が考案したものであり、Web サービスにおいては、WS-I Basic Profile v1.1 とWS-I Basic Security Profile v1.0 が広く利用されている。OGSA プロ

☆3 <http://www.ws-i.org/>

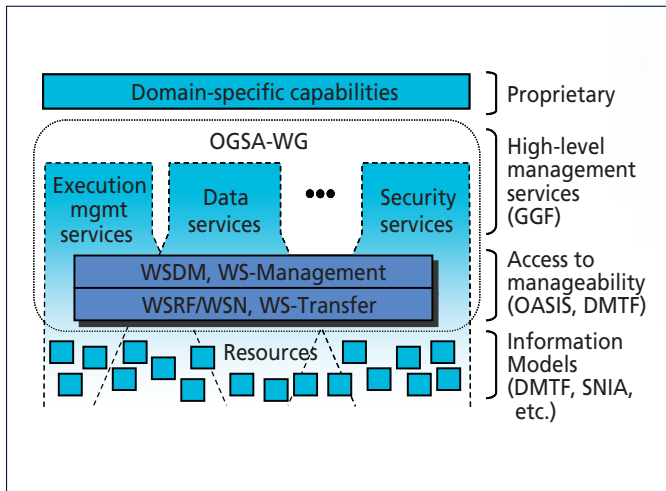


図-3 リソースの管理技術

ファイルは、この考え方に準拠し、プロファイルの対象とする仕様群の完成度および、普及度の定量化を行い、必要であれば、若干の機能追加を許容するものである。現在、WS-addressingとWSRFに基づいたOGSA WSRF Basic Profile 1.0と、セキュリティに関するOGSA Basic Security Profile^{2), 3)}の2つのプロファイルが作成されている。相互接続性の観点からはグリッドの基盤となるWebサービスの仕様群は1つであることが好ましいが、実際には複数のほぼ同等のWebサービス標準仕様群が同時に提案されることがある。そこで、OGSAでは、グリッドシステムのより広範囲な普及と、将来に向けた機能拡張の余地を残すため、OGSAプロファイルの定義に準拠していれば、複数のプロファイルの策定を許容している。さらに、複数のプロファイル間でなるべく同一の基盤機能を定義することにより、グリッドサービスの仕様開発を基盤となるWebサービスの発展と独立に進めることを可能としている。

グリッドが扱うリソースは多様であるが、それはシステムが複雑になることも意味する。この複雑さと、その結果としての管理コストをどのように軽減するかが、グリッドの主要課題の一つである。OGSAでは、この問題を、Webサービス技術を使ったリソースの仮想化(virtualization)により解決する。たとえば、ベンダごとや、同一ベンダであっても製造時期等の違いにより、個々のサーバが持つ機能や制御インターフェースは異なってしまう。しかし、あらかじめ基本的な共通インターフェースを定義することで、すべてのサーバを一定の範囲では同一に扱うことが可能になる。これをサーバの仮想化と呼ぶ。このとき、個々のサーバの特徴的な機能を活かすために、インターフェースの拡張余地を残した上で、すべてのサーバがサポートすべき共通インターフェースを定義している。さらに、サーバ・ストレージ・ネットワーク

といった異なる種別のリソース間でも、ライフサイクル管理などの基本的な機能は、複数のリソース種別に渡る共通なインターフェースとして定義する。

ビジネスグリッドプロジェクトのメンバが、OGSA-WGの共同議長として参加し、機能要件の提示やアーキテクチャに関する技術的提案を積極的に行っている。

❖ リソースの管理技術

グリッドシステムにおいては、それが利用するITリソース(計算機、ネットワーク、ストレージ、ミドルウェアなど)をどのようにモデル化し、その情報にいかアクセスするかということが大きな課題となる。グリッドにおけるITリソースは一般に複数のベンダやサプライヤから供給されるヘテロニアスなものを想定する必要があるので、これを共通的に取り扱うことができる標準が必要となる。

このようなITリソースの取り扱い方法を標準化する場として、OGSA内のサブグループとしてRM-DT(Resource Management Design Team)が設けられている。ビジネスグリッドプロジェクトは本DTにおいて、ITリソースのモデル/アクセス方法としてはすでにいくつかの標準化の動きが存在していることに留意し、GGFとして独自の仕様を開発するのではなく、既存の標準のうちのようなものがベースとしてOGSAのリソースモデルが構築できるかという観点で検討を進めるべく働きかけを行ってきた。これは、主に以下の2つの動きにまとめることができる。

- (1) ITリソースのモデル化や記述方法に関しては、関連標準化団体であるDMTFが開発しているCIM(Common Information Model)の汎用性に着目し、OGSAからCIMがどのように利用できるかというガイドラインの構築、およびGGFの観点からのDMTFに対する仕様のフィードバックを行っている。
- (2) ITリソース情報へのアクセスに関しては、OGSAと親和性のよいWebサービスの枠組みを用いたアクセスを行うための標準として、関連標準化団体であるOASISが定めているWSDMなどの標準を用いてITリソースの状態を管理、制御するための仕様の検討を行っている。

以上の活動をまとめると、RM-DTの描くITリソースのモデル化/アクセスの全体像は図-3のように表すことができる。

ビジネスグリッドプロジェクトでは、RM-DTの活動と並行して、上記(2)の標準化を行うOASIS WSDM-TCにも積極的に参画してきた。WSDMのプロトタイプ実装と、それを用いた接続実験(プラグフェスタ)への参加を通じた仕様の検証により、WSDMの標準化を

加速すべく活動を推進し、OGSAのリソースアクセスのベースとなる WSDM1.0 および 1.1 の OASIS 標準としての採用を見ることができた。

❖ セキュリティ

ビジネスグリッドを始めとするグリッドシステムでは、ネットワークを介し計算機が接続された分散環境でのシステムの運用を前提としており、システムのセキュリティが本質的に重要となる。

GGF においては、当初 OGSA-Security-WG が OGSA のセキュリティアーキテクチャの検討を行っていたが、2004 年から OGSA-WG 内のサブグループである Security Design Team がその検討を引き継いだ。ビジネス的側面からの要求を OGSA セキュリティアーキテクチャの検討へ反映させるために、ビジネスグリッドプロジェクトのメンバは本 DT 設立当初から主要メンバとして議論に参画した。また、3 年間のプロジェクト期間中、主に OGSA アーキテクチャ文書制定と、セキュリティ関連の 2 件の OGSA プロファイルの執筆に貢献した。

OGSA アーキテクチャ文書に関しては、データセンタの広域連携に関する仮想組織のユースケースを提案した。これは OGSA 文書のセキュリティの章にセキュリティ機能が実現すべき仮想組織のユースケースとして採用された。このユースケースの提案にあたっては、データセンタの広域連携のユースケースを基に、これと同等の、より一般化した電子図書館のユースケースを考案し提案した。

ビジネスグリッドプロジェクトでは、データセンタの広域連携において、インターネットを介した WSRF、WSN (Web Services Notification) 通信を行っており、この通信の相互認証、秘匿、および完全性の保証といったセキュリティ要件を満たす必要があった。これらのセキュリティ要件を満たすため、OGSA Basic Security Profile と呼ばれる一群のプロファイルを規定する方針とした。まず第 1 弾として OGSA における TLS (Transport Layer Security) プロトコル通信の利用について定める OGSA Basic Security Profile 1.0 - Secure Channel (セキュア通信プロファイル) の仕様策定を主導し、標準化を推進した。またあわせて OGSA Basic Security Profile 1.0 - Core の仕様策定を主導し、標準化を推進した。このプロファイルは OGSA サービスとのセキュア通信に用いる鍵情報を OGSA サービスと結びつける方法について規定している。

今後は Basic Security Profile の策定だけでなく、セキュリティアーキテクチャの一貫として詳細化が必要な OGSA での認可モデルなどについても議論を行う必要がある。

❖ 業務実行依頼と交渉

グリッドシステムに、業務を投入する際に必要となる業務実行依頼の合意を取り決める文書ならびにプロトコルを規定するのが WS-Agreement 仕様⁵⁾である。この合意の交渉はグリッドサービスを提供するサービスプロバイダと、それを利用するサービスクライアント間で行われる。クライアントとプロバイダ間でやりとりされる、WS-Agreement 文書は、フレームワークを決める応用ドメインに非依存の部分と、各サービスに固有なドメイン依存部分からなる。典型的な利用シーケンスは以下のようなになる。

- (1) サービスプロバイダが、提供するサービスの機能などを、あらかじめ登録
- (2) サービスクライアントが、サービスプロバイダの機能を参照して、要求するサービスの機能 (CPU 数、メモリ容量など) と満たすべき SLA 要件をドメイン依存の記述形式で記述してサービスを依頼
- (3) サービスプロバイダが、現在保有しているリソースや、実行しているサービスなどを元に、クライアントの要求に応じられるか否かを判断して合意するか、辞退するかをサービスクライアントに通知

ビジネスグリッドプロジェクトでは、ドメイン依存部分に次章に述べる JSDL 記述を拡張したものを、いわば WS-Agreement + JSDL 拡張という組合せで、クライアント/プロバイダ間の交渉を行うようにしている。WS-Agreement の仕様に関しては 2004 年末に、最初のパブリックドラフトが提出された。このドラフトに関しては、ビジネスグリッドプロジェクトでの実装を鑑み、仕様のあいまいな点の明確化、ならびに、必要なプロトコル拡張 (非同期プロトコルへの拡張) の提案を以下のように行った。

〈非同期プロトコルへの拡張の提案〉

初期の提案では、Agreement が確定するのに時間はさほどかからないという仮定から同期プロトコルしか提案されていなかった。ビジネスグリッドのようにジョブを複数のサイトにまたがって投入する場合は (図-4 参照)、同期プロトコルを使った場合、タイムアウトなどの不具合が生じ、実用的ではなかった。この点に関する提案書をパブリックコメント期間内に作成し、提出した。この仕様は 2005 年 9 月に GGF エディタに提出された 2 回目のパブリックコメント仕様書に反映された。現在 (2006 年 6 月) は仕様書の GGF 勧告成立に向けて、最終確認を行っている。

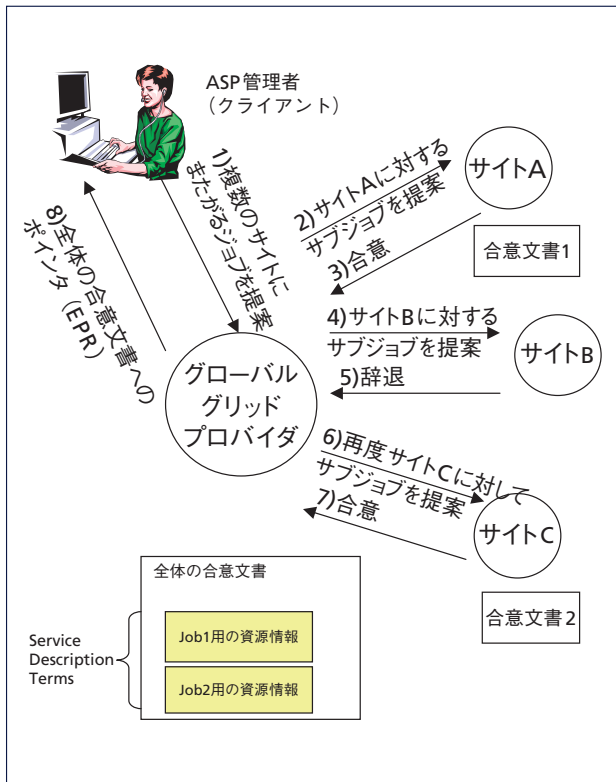


図-4 業務の実行依頼と交渉

❖ 業務に必要なリソースの記述

グリッドシステムでは、動的に投入される業務に対して、散在するITリソース群の中から最適なものを選び出し、割り付けることで、ITリソースの有効活用を図る。これに必要な技術が、JSDL仕様がカバーする、業務に必要なリソースの記述である。ビジネスグリッドプロジェクトではこれらの仕様検討の過程で、ビジネスアプリケーションに必要なとされる仕様の拡張案を継続的に検討してきた。

JSDL 1.0⁴⁾ は2005年11月に公開された。JSDL 1.0はジョブ識別子、アプリケーション、リソース、データステージング要求仕様の記述アトリビュートを含む。そのスキーマはアトリビュート間の関係と構造、値、選択肢の記述方法を定義する。

ジョブ識別子アトリビュート群：このアトリビュート群はジョブの補足説明と情報を提供するために用いられる。これらにはJobName、JobProject、JobAnnotationなどが含まれる。

アプリケーションアトリビュート群：このアトリビュート群は実行すべきアプリケーションとその引数、環境を記述する。アプリケーションはApplicationName、ApplicationVersionなどの属性を用いて抽象的な方法で記述される。特にApplication要素は拡張することが可能である。これにより、新しいアプリケーションに対するより具体的な要件を記述することができるように

するための拡張ポイントを提供する。科学技術計算では、POSIXアプリケーションが、広範に利用されている。JSDL 1.0はこのPOSIXアプリケーションの記述の標準仕様を、付録に収録している。ビジネスグリッドアプリケーションもその対象となる。

リソースアトリビュート：このアトリビュート群はジョブに必要なとされるリソースを記述する。これにより、ビジネスグリッドシナリオの重要な要素であるネットワークバンド幅の指定などを含む。

データステージングアトリビュート：このアトリビュート群はデータの移動（実行環境へのファイルのコピーと削除）を記述する。

❖ 業務の配備

一般に、ジョブを実行するには、必要なリソースを獲得するだけでなく、何らかのアプリケーションをインストールしたり、機器の設定をしたりする必要がある。これら各構成要素の設定には相互依存関係があり、一貫したシステムとして全体を構成するには特定の手順が必要になる。特に企業システムは多種多様な構成要素からなり、その手順は複雑なものとなり得る。このような処理自体をジョブの一部として手続き的に記述するのではなく、システム構成情報として宣言的に記述し、具体的な設定・配備処理をミドルウェアに任せられれば、保守性の向上が期待できる。CDDLWGでは、複数の構成要素（コンポーネント）からなるシステムを一貫性をもって設定・配備し、ジョブが実行可能な状態を管理するための標準の策定を行っている。CDDLの仕様⁶⁾は、設定記述言語（CDL）、構成要素のモデル（コンポーネントモデル）、Webサービスインタフェース（Deployment API）からなる。現在（2006年6月）、提出された上記3仕様の勧告成立のために、複数実装のインタオペラビリティ実験を行っている段階である。

コンポーネントは、システムの配備の状態を表す最小の単位であり、そのライフサイクル（状態遷移）や、必須のプロパティ（属性）、および配備時のコンポーネント間依存関係（フロー）の規定方法がコンポーネントモデルとして規定されている。

CDL（Configuration Description Language）はグリッド環境で重要なオンデマンドの構成管理を実現するために、テンプレート（プロトタイプ）の継承、リファレンス解決の機能を提供するXMLベースの言語である。コンポーネントを組み合わせたシステムの構成や、コンポーネントのプロパティはXMLで表現される。各コンポーネントについて、設定すべきプロパティの組はテンプレートとして定義される。コンポーネントを組み合わせるシステムを構成するにはこのテンプレートを継承し

て新たな XML 文書をつくり、そこに必要な値を記述する。プロパティに設定される値間の依存関係を表現するために、CDL では値を参照するためのリファレンスを用意している。プロパティの中には、実行時（配備時）にその値が決定されるものも存在するため、実行時に値を解決するリファレンスの記述も可能である。

サービスインタフェースとしては、システムを生成するポータル、各システムの配備管理を行うシステム、および各コンポーネントを WSDM の枠組みで管理する（状態監視など）ためのインタフェースが規定されている。

❖ 業務のレポジトリ

ビジネスアプリケーションの実行にはアプリケーションバイナリ以外にもさまざまなファイルが必要である。また、グリッド環境への投入を実現するためには、その配付・配備や実行を指定するさまざまなメタ情報も必要となる。これらを誤りなく、容易にシステムへ投入するためには、その投入形式やインタフェースの標準化に基づいた自動化が有効である。ACS（Application Contents Service）はグリッドシステムへのジョブの投入点におけるサービスの一つであり、業務のレポジトリの標準インタフェースを定義することにより、さまざまな実装に対して、相互接続性を提供する仕様である。また、OGSA WSRF Basic Profile および各種 WS 標準と組み合わせられて利用されることを想定している。

ACS 仕様書 1.0⁷⁾ はアーカイブデスクリプタ形式（AAD：Application Archive Descriptor）を規定する。AAD はグリッドジョブの構成ファイル群に対するアーカイブを記述する XML 文書であり、対応する XML Schema が仕様書に含まれている。ACS に関する想定ユースケースでは、業務システム開発者があらかじめ開発完了した業務パッケージを、それを構成するファイル群とともに「アプリケーションアーカイブ（AA）」として、ACS レポジトリに登録する。その後、業務管理者が登録済み AA への参照情報とともに業務投入を依頼することにより、複雑な業務を容易に実行することが可能になる。また、ジョブ投入パラメータを変えることにより、同一 AA を再利用することが可能になるため、業務の反復の実行も容易になる。この AAD は登録先レポジトリ内での識別子となる AAID の他にアーカイブの作成者情報、説明記述、アクセス制御情報、署名情報、アーカイブ内容一覧を含む。アクセス制御情報や署名情報については独自の定義を行うのではなく、それぞれの XML 標準を参照している。アーカイブ内容には CDL 等のデプロイメント記述を始めとする GGF 標準仕様に基づいたファイルを含めることが想定されている。ACS 仕様はまた、レポジトリインタフェース（ARI：Application

Repository Interface）も規定する。これは OGSA Basic Profile 準拠のサービスとして ACS レポジトリを実装するためのインタフェース仕様を記述するものである。

ACS-WG はその成果として ACS 仕様書 1.0 を策定し、2006 年 6 月に公開された。この仕様書は、規約と補足情報の文書記述、および仕様として規定する XML スキーマ文書から構成されている。ビジネスグリッドプロジェクトのメンバは、ACS-WG の創設メンバとして仕様の策定に主導的な役割を果たしてきた。ACS 仕様書 1.0 に準拠した実装については、ビジネスグリッドプロジェクトによる参考実装の他にも NAREGI グリッド PSE において開発が進行中である。

❖ 通信接続性の確保

Web サービス向けの通信プロトコルとして、SOAP が広く普及している。標準化団体 W3C が SOAP を W3C Note として公開し、さらにベンダによる実装や相互接続テストも進んだため、Web サービス同士の相互接続性は SOAP によって実現できるようになった。

しかし単純に SOAP を用いるだけでは、システムの信頼性を確保することはできない。たとえば、通信中にシステムダウンが発生したときに通信途中のデータが失われてしまったり、システムが復旧したときに同一のデータが重複して送信されてしまったりする可能性がある。

この課題の解決方法を考える際には、インターネット特有の要件について注意が必要である。インターネットではさまざまなハードウェアやソフトウェアを利用したシステムが、それぞれの組織の定めたルールの下で運営されている。そのため、システムごとの独立性や自律性が失われないう、Web サービス同士の連携は緩やかで柔軟な方法で実現しなければならない。

OASIS WSRM（Web Services Reliable Messaging）TC は、このような要件を考慮して、Web サービス間の通信における信頼性を保証するための高信頼メッセージの仕様である WS-Reliability を作成する TC である。ビジネスグリッドプロジェクトのメンバは WSRM TC の発足当時から、積極的に本 TC に参加してきた。WSRM TC は、その成果として、WS-Reliability V1.1⁸⁾ を作成した。この仕様は 2004 年 11 月に OASIS 標準として採用された。ビジネスグリッドミドルウェアでは、重要な通信部分に高信頼メッセージを用いるため、本 TC における標準化は、相互接続性を確保するために不可欠であった。WS-Reliability は次のような特長を持つ。

標準化団体による標準化：標準化団体 OASIS の WSRM TC で作成された、ロイヤリティフリーのオープンな仕様であるため、誰もが自由にこの仕様を実装できる。

高信頼性：高信頼メッセージの要件である、配送保証・

重複防止・順序保証の3つの通信品質（QoS：Quality of Service）を備えている。このため、アプリケーションレベルでは通信障害に関する処理を行う必要がなくなり、アプリケーション本来の処理に開発リソースをより集中できるようになる。

非同期通信と同期通信のサポート：高信頼メッセージの要件である非同期通信方式に加え、同期通信方式もサポートしている。このため、通信レイヤに MOM（Message Oriented Middleware）を前提とする非同期に動作するアプリケーションにも、通信レイヤに RPC（Remote Procedure Call）を前提とする同期的に動作するアプリケーションにも、対応することができる。

SOAP に準拠：SOAP 1.1 および SOAP 1.2 仕様に準拠しつつ、上記の高信頼性機能と非同期・同期通信方式を規定している。したがって、本仕様を実装する際には、既存の SOAP エンジンを下位の通信レイヤに流用することができるため、あらためて下位レイヤまで開発する必要がない。

他の Web サービス仕様との組合せ：セキュリティ機能など、他の機能のための Web サービス仕様と組み合わせられた利用が可能である。

IT業界の発展に向けて

GGF における OGSA を中心とするグリッド関連の標準化は、現在も精力的に継続されている。すでにグリッド全体を俯瞰するアーキテクチャの策定は終了しており、具体的なプロトコル、インタフェースの策定が機能ごとに詳細化されている段階である。このような標準化が完成すると、それに準拠した各種グリッドサービスの相互連携や活用が現実となり、データセンタの内部はもちろんのこと、データセンタをまたがった広域なビジネス向けグリッドの構築が柔軟に行えるようになってくる。このような状況は、「IT の所有から利用へ」の流れを加速し、

ユーティリティコンピューティング時代の本格化を促す可能性が高い。ビジネスの分野においても、必要なときに、必要なだけのリソースを、最適なコストで、安心かつ安全に利用できる環境が整うことになる。

このような環境においては、IT の利用者は、処理要求の急速な拡大や、不慮の災害などに対しても、標準的な枠組みを用いて頑強なシステムを構築することが可能となる。また、従来に比べて低廉なコストで豊富な IT リソースやサービスを楽しむことができるようになるため、余剰となった投資や人的、時間的リソースをより高度なサービスの開拓や拡充にあてることが可能となる。これによって、新たな IT 利用の地平が広がり、ひいては経済全体のさらなる活性化に寄与することが期待される。

ビジネスグリッドプロジェクトでは、当初からこのような効果をもたらす標準化活動の重要性を視野に入れ、ビジネスグリッドの実装を行うのと並行してその成果や知見を関連標準化団体に提案、フィードバックしてきた。その成果は GGF を中心とする各種標準に広く組み入れられ、今後のグリッドの発展に対して大きな役割を果たすものと考えられる。

参考文献

- 1) Foster, I., Kishimoto, H., Savva, A. et al. : The Open Grid Services Architecture, Version 1.0 (29 Jan. 2005).
- 2) Mori, T. and Siebenlist, F. : OGSA Basic Security Profile 1.0 - Core (9 Feb. 2006).
- 3) Mori, T. and Siebenlist, F. : OGSA Basic Security Profile 1.0 - Secure Channel (9 Feb. 2006).
- 4) Anjomshoaa, A., Brisard, F., Drescher, M., Fellows, D., Ly, A., McGough, S., Pulsipher, D. and Savva, A. : Job Submission Description Language (JSDL) Specification, Version 1.0 (7 Aug. 2005).
- 5) Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S. and Xu, M. : Web Services Agreement Specification (WS-Agreement) (20 Sep. 2005).
- 6) Bell, D., Kojo, T., Goldsack, P., Loughran, S., Milojicic, D., Schaefer, S., Tatemura, J. and Toft, P. : Configuration Description, Deployment, and Lifecycle Management (CDDL) Foundation Document (7 Aug. 2005).
- 7) Fukui, K. : Application Contents Service Specification 1.0 (8 May 2006).
- 8) Iwasa, K. et al. : WS-Reliability 1.1 (15 Nov. 2004).

(平成 18 年 7 月 28 日受付)