

# Grid Technology and Roadmap from UNICORE to OGSA

## グリッド技術とUNICOREからOGSAへのロードマップ

### Abstract

The development of Grid technology in Fujitsu dates back to the mid 1990s, long before the word "Grid" was used to describe the concept. Since 1996 Fujitsu Laboratories of Europe has led the development of the UNICORE Grid architecture and also created a realization of it through the Arcon implementation. UNICORE provides a comprehensive collection of "first generation" Grid functions. The Open Grid Services Architecture (OGSA) describes a framework for interoperable Grid systems, which provide additional "second generation" functionality. Unlike first generation Grid architectures, OGSA is built on top of the emerging Web Services technology, a shift that will make Grid computing accessible to the whole business community. This paper describes, through simple use cases, the key functions of both generations of Grid technology and outlines the roadmap from UNICORE to OGSA.

### あらまし

富士通におけるグリッド技術開発は、この概念が「グリッド」という言葉で表現されるよりずっと以前の90年代半ばに始まった。1996年以来、欧州富士通研究所はUNICOREグリッドアーキテクチャの開発を主導し、さらにArconという実現形態によりそのアーキテクチャを具現化してきた。UNICOREは第1世代のグリッド機能群を包括的に提供するものである。一方、オープングリッドサービスアーキテクチャ(OGSA)は、相互運用可能なグリッドシステムの枠組みを定義することで、第2世代のグリッド機能を提供するものである。第1世代のグリッドアーキテクチャとは異なり、OGSAは新たな先端Webサービス技術の上位に構築される。この変更によってビジネス社会全体がグリッドコンピューティングを利用できるようになるであろう。

本稿では、第1世代と第2世代のグリッド技術の主要機能を簡単な使用事例とともに説明し、UNICOREグリッドアーキテクチャからOGSAへの進展のロードマップの概要を述べる。



David Snelling  
Research Manager,  
Fujitsu Laboratories of Europe Ltd.



Sven van den Berghe  
Principal Researcher,  
Fujitsu Laboratories of Europe Ltd.

## 1. Introduction

The Grid has been described in many ways. The most popular view uses an analogy to the power grid, where resources (computational and data) are available “on demand”. This analogy works as long as the resources are free to all users. However, the resources on the Grid do not have a uniform pricing model, as does the power grid. Once people and institutions start to charge for the resource, a much more complicated negotiation process is required. This is a significant increase in the complexity of the problem, much greater than the power grid.

The most effective definition we have found to distinguish Grid computing from the Web, e-Commerce, and enterprise computing common today is as follows:

Grid computing is non-trivial distributed computing across multiple administrative domains.

The reference to distributed computing captures most of the technical complexity that constitutes the job of Grid middleware. Grid infrastructures, like traditional distributed computing environments (e.g. enterprise systems), need to deal with distributed coherency management, fault tolerance, synchronous and asynchronous communication patterns, static and dynamic deployment scenarios, heterogeneous hardware, heterogeneous software, object and non-object data systems, etc. There are many issues in traditional distributed computing that are still the subject of some research today.

To the basic complexity of distributed computing, we must add the complexities of multiple administrative domains over which this distributed system must be deployed. These key issues are what make Grid computing a challenge, e.g. security, site autonomy, corporate politics, protection of intellectual property, legal issues, etc. Many of the social aspects are not, and cannot be, solved by technology alone, but the technological infrastructure needs to be in place before we can attempt to address them.

The first and most critical step in this process is to establish standards that allow multiple Grid implementations to collaborate. Like the Web, Grid

computing will rely on these open standards to reach the critical mass necessary for wide spread adoption and hence profitable exploitation of the technology.

A team of academic and industrial Grid researchers, working within the Global Grid Forum (GGF: [www.ggf.org](http://www.ggf.org)) took one very important step along this road with the publication of the Open Grid Services Infrastructure (OGSI) specification.<sup>1)</sup> OGSI provides a basic set of functions, built on Web Services, that enable many of the extended functions needed for Grid computing. The Open Grid Services Architecture (OGSA) is another GGF activity that aims to build specifications for many of the important higher-level services needed in the near future.

**Figure 1** highlights the convergence of traditional Web Services, Grid Services, and more recent research in Web Services. The details of these different trends and how they are coming together are beyond the scope of this paper, however, the trends represented indicate a strong synergy within the community.

The remainder of this paper outlines the UNICORE Architecture and how Fujitsu is working together with its UNICORE partners to evolve UNICORE to this new emerging architecture for Grid computing.

## 2. Grids from Fujitsu

### 2.1 UNICORE and Arcon Architecture

The functions of UNICORE and its implementation, Arcon, have been described in Reference 2). This section is based on the discussion in this paper. Arcon is an implementation of the

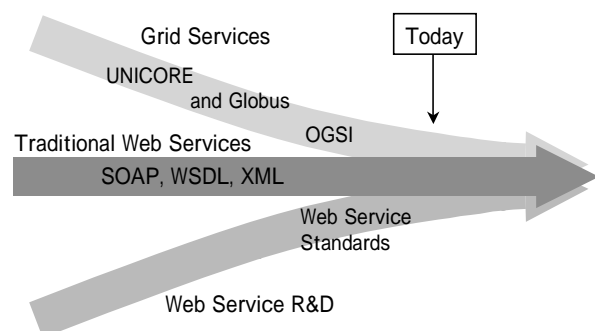


Figure 1  
Convergence of Grid and Web Services technology.



### 2.1.2 UNICORE Job model

UNICORE defines two important terms: Task and Job Group. A task is the atomic entity that is eventually incarnated into a real process (or batch job) configured for execution on the target system. A Job Group is recursively defined to consist of Job Groups and Tasks. The top level Job Group is called a UNICORE Job. It is the entity that represents the complete computational solution to the user's problem. The UNICORE Job is submitted as a whole to a UNICORE site (Usite).

### 2.1.3 Context of an Abstract Job Object

For the duration of a UNICORE Job, the NJS creates a temporary UNICORE workspace (Uspace) located in an installation specified file space. The Uspace holds temporary files created by NJS, for example the incarnated job scripts, consolidated standard output and standard error output, files transmitted from the client to the site, or data created by the executing jobs. UNICORE provides functions to transfer files between Uspaces and to import or export data from or to a permanent file space to which the user has access at various UNICORE sites.

The UNICORE Grid is a set of Usites (distinct administrative domains with firewalls etc.) Each Usite may contain several Vsites (Virtual Sites). Each Vsite typically represents a major computational resource, e.g. a large cluster, although other configurations are possible, e.g. multiple servers in one Vsite or overlapping Vsites.

### 2.1.4 Security

In Grid computing, security is of paramount importance and UNICORE is well known for its serious approach to security. Security issues with respect to user and site interaction can be characterized in terms of four concepts: authentication, authorization, integrity, and confidentiality. UNICORE (and the Arcon implementation) address these issues as follows.

User authentication is managed in UNICORE using standard X.509 based certificates. There are a

number of Public Key Infrastructure (PKI) models that can be used for handling the authentication process. These range for fairly loose processes based on an exchange of e-mails, to more formal processes involving physical checking of identity, e.g. passport checks. It is important to note that UNICORE is the only open Grid environment that requires un-extended X.509 certificates. This means that existing, commercial PKIs can be used for user authentication.

The UNICORE architects have been very careful to separate authentication from authorization in the design of the security model. The users are authenticated by an authentication process (PKI) agreed to by all members (sites) of a Grid community. This level of agreement poses significantly less conflict than full agreement of all aspects of user management. Therefore, by allowing sites total autonomy over user policy, it is much easier to obtain all the agreements needed to establish a Grid community, than would be the case if authentication and authorization were not separated.

The remaining security properties of integrity and confidentiality are ensured by using SSL based connections between all agents (user clients and servers).

On other aspect of security often overlooked, is that of the underlying trust model implied by a Grid system. The authorization settings within a site define the trust relationships between individual users and the sites that authorize them to use site facilities. However, the trust relationship between two sites is more complicated to define and relate to systems administrators. If two sites totally distrust each other, then no Grid based collaboration is possible. So some trust is required, but on the other hand, a site is unlikely to completely trust another site's systems administrator. UNICORE provides a trust model that restricts the privileges of a remote systems administrator to the minimum necessary to operate a Grid. This amounts to sending and retrieving sub-jobs that have been already signed and authorized by an end user. This means that, except for temporary storage of these sub-jobs, all trust issues devolve to

user-site trust model established by the authorization relationship between the site and the end user who signed the sub-job.

### 2.2 Globus based services

Fujitsu is also a leader in providing Globus based Grid technology in Japan. Globus, unlike UNICORE, does not provide the full vertical environment for Grid computing. Rather Globus is a tool kit that allows developers to create complex and flexible applications in a Grid environment.

Globus has a similar three-tier structure to UNICORE, i.e. client, gatekeeper, and the GRAM (job manager). However, there are differences worth noting, UNICORE has better support for seamless computing and provides client based workflow support missing in Globus. The underlying trust model of Globus is less strict than the UNICORE model.

In UNICORE seamless computing means that an abstract job created to run on machine X at site A will also run, without modification, on machine Y at site B. This is implemented by a piece of software called the Incarnation Database, which creates a machine and site specific incarnation (or rendering) of the UNICORE abstract representation in terms of the specifics of the given machine and site.

A user, creating a complex interaction (workflow) between servers at a number of sites, faces two very different programming models between UNICORE and Globus. In UNICORE, a client based visual editor is used to create complex workflows easily. The dependencies between job steps are visually provided to the user and automatically managed by the UNICORE server (NJS). The corresponding process in Globus requires the user to create, at the end of one job step another job as the last task in the script. In addition to the complexity of creating these scripts within a script, the ability to deal with non-sequential job steps is very complicated, although it is possible. UNICORE also provides built-in graphical support for conditional execution and loops within a job.

Although the security interactions between users

and the sites on the Grid are very similar between UNICORE and Globus, the implementation in Globus results in a more relaxed trust model between sites. In short, the Globus implementation requires sites to trust the systems administrators at remote sites. This creates the potential for a rogue system administrator at one site to assume the identity of a user at another site. This is not possible with UNICORE. An analysis of this issue and a detailed comparison is given elsewhere.<sup>3)</sup>

## 3. Fujitsu's OGSA roadmap

The capabilities of Grid systems in general, and Fujitsu's in particular, have increased over the past few years, but the promise of the future is also significant. The next section describes some of the use cases that exist today in UNICORE enabled Grid environment. Fujitsu's roadmap to OGSA will provide continuous support for all these services and incrementally add the new features enabled through OGSA described in Section 3.2.

### 3.1 What can Grids do today?

The Grids that are available today are some steps along the journey towards the vision of on-demand computational resources. Although there is a long way to go, the current Grid systems provide many new capabilities that change the way that people use computational resources.

One of the basic functions that are available in current Grids is the ability to run batch jobs on high-value resources such as supercomputers or very large data center clusters. The batch jobs can be submitted from remote locations, and the Grid systems will take care of the authentication and authorisation of non-local users. UNICORE also alleviates the problems of running jobs on different sites by abstracting many of the site-specific details, such as executable and file locations. Grids for batch jobs allow job submission, monitoring and control.

In order to support batch jobs, Grids must also handle file transfer. Files are needed for the job scripts, input data and output results. A range of

different options is available to support the transfer of a wide range of file sizes, which can be up to multi-gigabyte files for input and results. The transfer of data can be integrated into the workflow, as in UNICORE, or it can be handled by separate Grid-enabled applications, such as GridFTP built on top of Globus.

The two basic functionalities of batch jobs and data transfer can be combined to provide novel but essentially simple new ways of making use of computational resources. The first is the obvious application of providing convenient access to resources no matter where the user is located. A more sophisticated application is to make resources available to users that can flexibly cover wide ranges in demand. It is possible nowadays for an economist to run a sophisticated model of an economy in order to evaluate the effects of events or changes to policy. In the run up to significant economic events, such as announcements of national budgets, groups of economists will run a large number of simulations to test the effects of possible options and to establish sensitivities. The surge in demand is large but temporary (the need dies away once the budget is set) with tight deadlines. Maintaining sufficient resources just for a short peak season is not economic. However, if these simulations are run using a Grid system, then excess demand can be securely and seamlessly migrated to remote systems.

The data transfer supported by Grids is more sophisticated than simple file transfer. Some scientific experiments can generate a large set of results, which are then analysed by a community of researchers based in many different institutions. There are Grid systems available to handle the distribution of this data, making sure that local copies are used where possible and that remote copies are used if not. These systems also ensure that the data is kept consistent, with changes, updates and corrections, automatically made to all the distributed copies.

Current Grid systems do more than execute a single-step job; they can also execute a workflow, where jobs consist of a number of steps that must be

executed in a certain order. The results of one step can be fed into another and it is possible to modify the execution sequence, with branch or loop constructs, depending on the results of previous steps. Grid based work-flow allows each step to be executed on a different system and it is possible to adjust where a step runs if conditions change after a (Grid) job has been started.

Since Grid systems allow a user to access resources from any location, it is possible now for a user to become much more directly involved with the progress of a simulation by fetching interim results to monitor the progress of a simulation as it is running and even to send data to a running simulation to change the progress of the simulation. Using the Grid a researcher can steer an application as it is executing on one system while using a second system to visualise the (interim) output data.

### 3.2 What will OGSI and OGSA add?

From the above it is clear that Grid computing can already provide remote service execution, resource sharing, off loading excess workload, data distribution, and application mobility across multiple administrative domains. The question is then what else is possible with the development of OGSA? The answer to this question comes in two parts. First, we describe the extended basic functionality provided by OGSI,<sup>1)</sup> and second we discuss the higher level capabilities envisioned in OGSA,<sup>4)</sup> built on top of OGSI.

#### 3.2.1 *Open Grid Services Infrastructure (OGSI)*

As its name implies OGSI provides underlying capabilities on which other services can be built. The services provided by OGSI can be combined in various ways and then extended to provide application specific capabilities. The key services supported by OGSI are global service naming, introspection, notification, dynamic service creation, life cycle management, and service aggregation.

1) Global service naming

Grid Services exist for a variable length of time

and as a result may outlive the platform on which they run. The name of such a service needs to be valid longer than the address of the server running the service is valid. Thus the “endpoint” capability provided by Web Services is insufficient. Furthermore, a Grid Service may be accessible through several paths, e.g. across a secure connection over the Internet or via a LAN within a corporate office. To support these requirements, OGSI provides a two level naming scheme. When a service is created a Grid Service Handle (GSH) is created, which refers to that service. This GSH is valid for the lifetime of the service and will never refer to any other service, e.g. GSHs cannot be recycled. Once in possession of a GSH a client may request a “resolver” (also defined by OGSI) to provide a Grid Service Reference (GSR) for the service referred to by the GSH. The GSR contains all addressing information and may expire, for example if the service is relocated to a new platform or server.

### 2) Introspection

Once a service exists it needs to advertise itself to potential clients. The approach with Web Services is to register the service with a UDDI (Universal Description, Discovery and Integration) registry. Since a Grid Service is a Web Service, it can be registered in a UUDI registry, but many of the important properties of a Grid Service are dynamic. UDDI has little or no capability for handling dynamic descriptions of services. OGSI provides a concept called Service Data. Properties associated with a service, like its current load for example, are exposed as Service Data and the client provided with a standardized OGSI interface for performing simple and complex queries against these properties.

### 3) Notification

In a further extension of the Service Data concept, Grid Services can push notification of changes in Service Data to interested clients. The client registers an interest in a property (or a query against a collection of properties) using a subscription interface, also included in OGSI. Notification subscriptions can be registered with longest and

shortest times between messages so that the client is not overloaded with message traffic, while ensuring that it receives assurances of the service’s continued existence, even when the service property of interest is not changing regularly. Subscriptions also have a lifetime, which can be extended, associated with them. In this way, forgotten services eventually stop sending messages and reduce unnecessary network traffic without the need for explicit service management.

### 4) Dynamic service creation

Unlike Web Services, Grid Services aim to be customizable. Clients can provide a set of properties that fine-tune the functionality of a service they would like to use. These properties serve as input to a Grid Service factory, which then creates and deploys the service on behalf of the client.

### 5) Life cycle management

Like subscriptions above, all Grid Services have a lifetime. This allows the Grid to perform soft state destruction on services that no longer are of interest to clients. Thus a service, once deployed, could be the subject of a complex management system, and in fact for efficient operation this might be desirable, but it is not required. By setting sensible values for the lifetime of a service, a service centre manager need not deploy full management infrastructures for all services under her control. In time, unused services release their resources and terminate automatically.

## 3.2.2 *Open Grid Services Architecture (OGSA)*

OGSA is the vision of all services, based on the OGSI, that are needed to address a collection of use cases collected by the GGF.<sup>4)</sup> The complete service collection is quite extensive, and includes the OGSI as the lowest layer of the architecture. The list of services envisioned is as follows:

- Open Grid Services Infrastructure
- Virtual organizations
- Discovery
- Choreography, orchestration and workflow
- Transactions
- Metering
- Rating

- Accounting
- Billing and payment
- Installation, deployment, and provisioning
- Fault management
- Problem determination
- Distributed logging
- Messaging and queuing
- Event
- Policy
- Job management
- Basic data services
- Advanced data services
- Generic agreement
- Reservation agreement
- Data access agreement
- Job queuing
- Service management

Rather than attempt to detail any of these many services, we present a few use cases that motivate the work of the OGSA working group.

### 1) UNICORE service

The existing UNICORE Grid environment can be realized easily as a collection of services from OGSA. A simple “discovery service” would expose the availability of UNICORE Vsites, a “job management service” would support the creation, monitoring, and control of UNICORE jobs, and a “basic data service” could be used to effect data and file transfer between sites. This is not to imply that UNICORE is a simple concept to implement as Grid Services, since like Grid Services the bulk of the technology is in the underlying fabric of the Grid. Once that is in place, the provision of services on top of it is a relatively small task.

### 2) Distributed query processing

In a specialized example of the factory pattern, a service can be created as the result of a complex distributed query, where the initial query and list of distributed data sources serve as input to the factory create process. The service thus created can then be used by the client for further processing, or shared with collaborators or partners.

### 3) Dynamic virtual organizations

One of the strong motivations for the Grid is the ability to create new business models. The idea of a “virtual organization” (VO) provides one of the basic building blocks for these new models. A VO is a collection of individuals or companies that enter into an agreement for the purpose of solving a problem or creating a business opportunity. These VOs can be short lived and have very limited scope, and therefore, it is imperative that they are lightweight. This requires a predefined collection of terms under which members will agree to join a VO. The negotiation takes place automatically via the “generic agreement service”, contract is established, the work done, and the VO dissolved all dynamically, possibly in the space of a few seconds.

Some of the business models enabled by this concept include:

- An Internet or Grid Service Bondsman for business-to-business transactions, in which the service guarantees that a service does what it claims to do.
  - An intellectual property clearing house or broker, where for example drug companies might be able to explore available technologies from other companies without revealing their own.
  - A new level in the out-sourcing, in which a company’s out-sourcing is funded by selling their own services or resources to others.
  - Provision of electronic “Chinese Wall” services within or across companies.<sup>note)</sup>
  - A virtual organization “dating” service, in which potential members pay to discover each other’s existence and the terms under which they do business. This could also be modelled as a “business man’s club” instead of a dating service.
  - A broker for dynamically arranged contracts.
- ### 4) IT infrastructure management

The OGSA “service management services” will

---

note) From [www.investorwords.com](http://www.investorwords.com), “A term used to describe procedures enforced within a securities firm that separate the firm’s departments to restrict access to non-public, material information, in order to avoid the illegal use of inside information.”

provide standardized interfaces for managing, provisioning, and controlling services. If a corporation's IT infrastructure is represented as Grid Services (something very natural with Grid Services, but not so for Web Services), these interfaces provide the framework for uniform management of highly diverse and heterogeneous resources and services within an organization. This is particularly important in large corporations affected by mergers and buy outs and in IT service companies, who need to manage the many diverse choices made by their clients in terms of IT infrastructure.

### 4. Conclusions

At the time of printing, the OGSA based version of UNICORE provided all the "first generation" interface. Some extended "second generation" Grid functions through a Grid/Web Services interface. Some extended "second generation" functions, such as interoperable broker services, notification capabilities, and basic support for virtual organizations, were also in advanced stages of development.

The following are good places at which to start an initial search for further information on Grid computing in general and UNICORE in particular.

The UNICORE Forum ([www.unicore.org](http://www.unicore.org)) is an organization, which acts as custodian and promoter of the UNICORE protocols and standards.

Intel in Germany, in collaboration with Fujitsu,

provides an industrial strength implementation of the UNICORE protocols that is available for free trial use and a demonstration Grid at ([www.unicorepro.com](http://www.unicorepro.com)).

There are two major EU research projects (EuroGrid and GRIP) that have just finished, which based their work on UNICORE, and a newly started project in Japan (NAREGI), [www.eurogrid.org](http://www.eurogrid.org), [www.grid-interoperability.org](http://www.grid-interoperability.org), [www.naregi.org](http://www.naregi.org).

The Global Grid Forum [www.ggf.org](http://www.ggf.org) is a combination of standards body and research forum. Thus both cutting edge concepts and industrial strength ideas are shared in an open forum.

The Web Services specification activity can be found at [www.w3c.org](http://www.w3c.org) and [www.oasis-open.org](http://www.oasis-open.org).

### References

- 1) S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, K. Kesselman, T. Maquire, T. Sandholm, D. Snelling, and P. Vanderbilt: Open Grid Services Infrastructure (OGSI) v. 1.0.
- 2) D. Erwin and D. Snelling: UNICORE: A Grid Computing Environment. LNCS 2150, Euro-Par 2001, Springer, 2001.
- 3) T. Goss-Walter, R. Letz, T. Kentemich, H-C. Hoppe, and P. Wieder: An Analysis of the UNICORE Security Model.  
<http://forge.gridforum.org/projects/ggf-editor/document/GFD.19/en/1>
- 4) OGSA working group.  
<http://forge.gridforum.org/projects/ogsa-wg/>

まえがき

グリッドという概念は様々に説明されてきた。一般的には、電力供給網に例えて、グリッドはリソース(コンピュータやデータ)をオンデマンドで使える資源供給網であると説明されている。ただし、この例えはすべてのユーザがリソースを自由に使える場合に限って意味がある。電力供給網と違って、グリッド上の様々のリソースは均一の価格モデルを採用しているわけではない。リソースを有する人々や機関がそれぞれに使用料を請求し始めたら、込み入った交渉プロセスが必要になる。このことが電力網の場合よりリソースの問題をかなり複雑にしている。

現在普及しているWebサービス、電子商取引および企業内コンピューティングとグリッドコンピューティングを区別する最も有効な定義は、つぎのとおりである。

グリッドコンピューティングは、複数の管理ドメインにまたがって意味ある処理を行う分散コンピューティングである。

分散コンピューティングを引き合いに出せば、グリッドミドルウェアの機能を実現する技術の複雑さが理解できる。従来の分散コンピューティング環境(例えば企業システム)と同様に、グリッド基盤は、分散整合性管理、耐障害性、同期・非同期通信形態、静的・動的な展開シナリオ、多種多様なハードウェア・ソフトウェアなどに対応する必要がある。従来の分散コンピューティングには、今日でも研究課題となっている様々な問題がある。

分散コンピューティングの基本的な複雑さに加えて、分散システムが展開される複数管理ドメインの複雑さも考えなければならない。グリッドコンピューティングにとっての主要な課題は、セキュリティ、サイトの自律性、企業の戦略、知的財産の保護、法律的問題などである。社会的側面を持つこれらの課題は技術のみで解決することはできないが、これらの課題に取り組む前に技術基盤は整備されなければならない。

この技術基盤の整備プロセスの最初の重要なステップは、複数のグリッド実施計画を共同して推進できるような標準規格を制定することである。Webの場合と同様に、グリッドコンピューティングを広

範囲に導入して有益に活用できるような臨界点に到達させるには、オープンスタンダードとしての標準規格の制定が欠かせないであろう。

このオープンスタンダード制定の重要な一歩として、グローバルグリッドフォーラム(GGF: Global Grid Forum)(www.ggf.org)の学術機関および産業界のグリッド研究者のチームがオープングリッドサービスインフラストラクチャ(OGSI: Open Grid Services Infrastructure)仕様<sup>(1)</sup>を発表した。OGSIは、Webサービス上に構築される基本機能セットを提供し、グリッドコンピューティングに必要な数多くの拡張機能を実現する。オープングリッドサービスアーキテクチャ(OGSA: Open Grid Services Architecture)はGGFの研究活動の一つで、近い将来必要になる数多くの重要な高度サービス仕様の確立を目的としている。

図-1は、従来のWebサービス、グリッドサービスおよび最近のWebサービス研究の収束状況を示している。これらのトレンドの詳細およびどのように収束するかはこの論文の範囲外であるが、これらのトレンドが共同体の中で強い相乗効果を持っているということが言える。

以下に、UNICORE(Uniform Interface to Computing Resources)アーキテクチャの概要を述べ、さらに富士通が開発パートナーとともにUNICOREをどのようにグリッドコンピューティングのための新しいアーキテクチャに進化させてきたかをまとめる。

富士通が提供するグリッド技術

UNICOREとArconアーキテクチャ

UNICOREとその実現形態であるArconの機能は

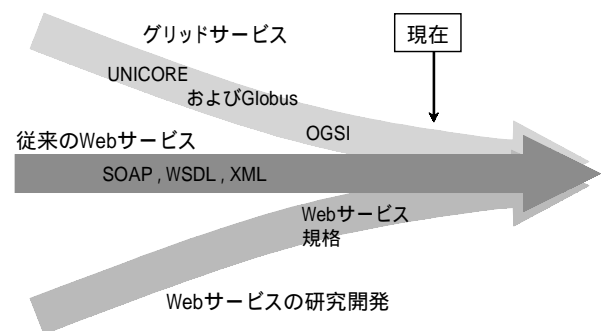


図-1 グリッドとWebサービス技術の収束  
Fig.1-Convergence of Grid and Web Services technology.

参考文献(2)に記述されている。本節の内容はその文献内の議論に基づいたものである。図-2に示したように、ArconというのはUNICOREアーキテクチャの一つの実現形態である。以下に、UNICOREに採用されている3層モデル、UNICOREジョブモデル、UNICOREデータモデルおよびリソース発見メカニズムを説明する。

【UNICORE3層モデル】

UNICOREクライアントは、ジョブを完了するために一つ以上のUNICOREサイトにある複数のシステムを要求する複雑な同期・非同期ジョブに関して生成・操作・管理をサポートする。ユーザが定義するジョブとアクションは、Javaクラスであるアブストラクトジョブオブジェクト(AJO: Abstract Job Object)群として表現され、UNICOREの構成要素間で転送される際に直列化されて署名される。UNICOREのサーバレベルは、UNICOREサイトへの安全なエントリポイントであるゲートウェイで構成されている。UNICOREクライアントからのリクエストはそのゲートウェイでX.509証明書に基づいて認証され、ネットワークジョブスーパーバイザ(NJS)に渡されて処理される。NJSは、AJOで表

現された抽象的リクエストをターゲットシステムで実行される具体的ジョブやアクションにマッピングする。このプロセスを「具体化」と呼ぶ。異なるサイトで実行する必要のあるサブジョブは、当該サイトのゲートウェイに転送されて同等なNJSによって処理される。NJSのそのほかの機能には、ユーザが指定した依存関係を持つジョブの同期化、ジョブ実行に必要なUNICOREサイト間のデータファイル自動転送、ジョブ実行結果(特に、標準出力と標準エラー出力)の収集、UNICOREジョブ作業空間間のデータファイルのインポートとエクスポート、ターゲットサイトで使用可能なファイルシステムとアーカイブ、そしてクライアントワークステーションなどがある。

このアーキテクチャの3番目の層は、具体化されたユーザジョブを実行するターゲットホストである。ターゲットホスト上にはターゲットシステムインタフェース(TSI: Target System Interface)と呼ばれる小規模なデーモンが常駐していて、ユーザに代わってローカルバッチシステム(または、ほかのプロセス管理ソフトウェア)との交信を行う。NJSとTSIの間の通信にはステートレスプロトコルが使

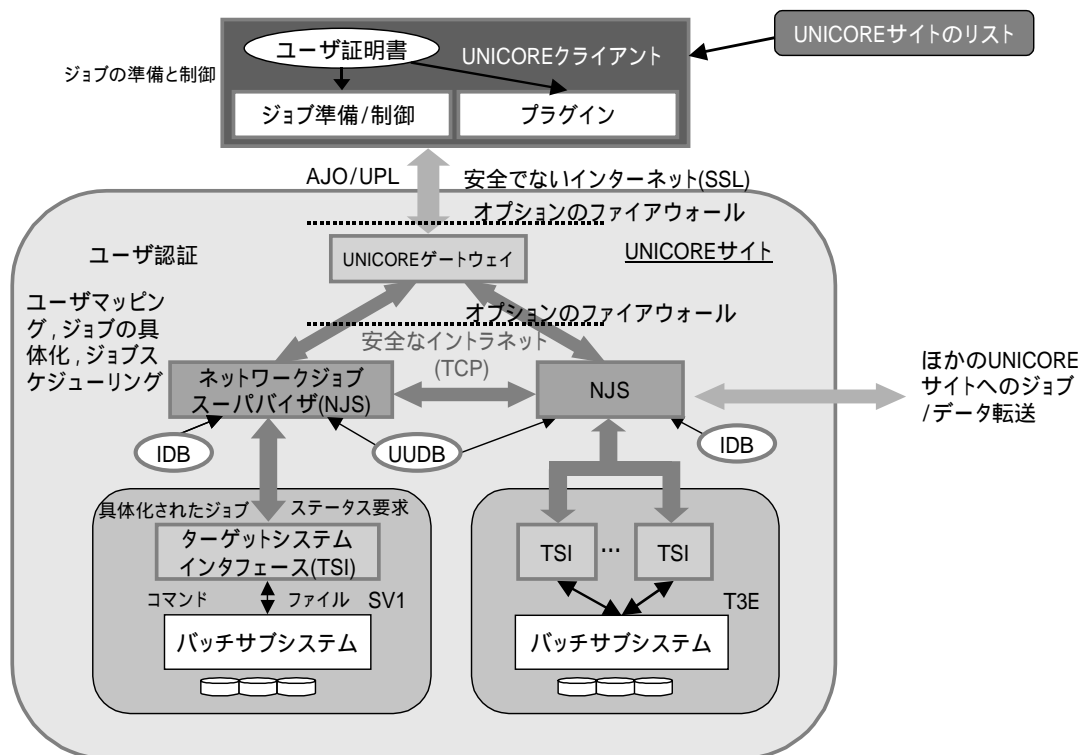


図-2 UNICOREアーキテクチャの概要  
Fig.2-UNICORE architectural overview.

用される。また、性能向上のために単一ホスト上で複数のTSIを動作させることもできる。

#### 【UNICOREジョブモデル】

UNICOREでは「タスク」および「ジョブグループ」という二つの重要な用語を次のように定義している。一つ目のタスクとは、ターゲットシステムで実行できる形の実プロセス（またはバッチジョブ）に最終的に具体化される極小の構成要素である。もう一つのジョブグループは、複数のジョブグループおよびタスクで構成されるように再定義される。最上位のジョブグループはUNICOREジョブと呼ばれる。UNICOREジョブとは、ユーザの問題に対する完結したコンピューティングソリューションを表わす構成要素である。UNICOREジョブ全体が一つのUNICOREサイト（Uサイト）に投入される。

#### 【アブストラクトジョブオブジェクトの文脈】

UNICOREジョブ実行中に、NJSは導入時に指定されたファイル空間に一時的UNICORE作業空間（U空間）を生成する。このU空間にはNJSが生成した一次ファイル群が保持される。これら一次ファイルの内容は、例えば、具体化されたジョブスクリプト、統合された標準出力と標準エラー出力、クライアントからサイトに伝送されたファイル、または実行中のジョブが作成したデータなどである。UNICOREには、U空間の間でファイルを転送する機能や、様々なUNICOREサイトからユーザがアクセスする常駐ファイル空間と一次ファイル空間との間でデータをインポート/エクスポートする機能がある。

UNICOREグリッドはUサイト（例えば個別のファイアウォール付きの管理ドメイン）の集合である。個々のUサイトはいくつかのバーチャルサイト（Vサイト）を含む。各Vサイトは通常大型クラスタなどの主軸となるコンピューティングリソースを有する。さらに、単一のまたは重複するVサイトに複数サーバが存在するような構成も可能である。

#### 【セキュリティ】

グリッドコンピューティングではセキュリティが決定的に重要であるが、UNICOREがセキュリティを重視していることはよく知られている。ユーザとサイト間のインタラクションに関するセキュリティの課題は、認証、認可、保全および機密保持という4種類の概念で特徴付けることができる。

UNICOREでのユーザ認証は、X.509規格に基づいた証明書を使って管理される。認証プロセスを処理するための公開鍵インフラストラクチャ（PKI：Public Key Infrastructure）モデルは数多くある。Eメール交換による公平で厳しくない認証プロセス処理から、パスポート確認などの物理的な検査を含む公式の認証プロセスまで様々である。ここで指摘しておきたい重要なことは、UNICOREが、非拡張X.509規格に準拠した証明書を要する唯一のオープンなグリッド環境であるということである。言い換えれば、現行の商用PKIがユーザ認証に使用可能であるということである。

UNICOREの設計者は、セキュリティモデルの設計に際して認証と認可をとて注意深く区別してきた。あるグリッドのユーザは、当該グリッド共同体の全メンバー（全サイト）によって合意された認証プロセス（PKI）を使って認証される。このレベルの合意によるユーザ認証は、ユーザ管理のすべての側面を包括した合意を前提とする場合に比べて、明らかに不一致を生ずる度合いが低い。それ故、認証と認可を区別しない場合と比べて、サイトをユーザポリシーから完全に自立させることで、グリッド共同体の開設に必要なすべての合意を得ることが容易になるのである。

残りのセキュリティ属性である保全と機密保持に関しては、すべてのエージェント間（ユーザクライアントおよびサーバ間）の接続にSSL仕様のプロトコルを用いることで安全性を確保している。

見逃されがちなセキュリティの一側面として、グリッドシステムが包含する基礎的なトラストモデル（信頼規範）というものがある。各サイト内では、個々のユーザと、ユーザにサイト機能の使用を認可するサイト群との間の信頼関係を、認可設定によって定義している。しかし、例えば二つのサイト間の信頼関係を定義してシステム管理者間で折合いを付けることは、複雑で困難を伴うことである。もし二つのサイト間に信頼関係が全くなければ、グリッド上の共同作業は不可能である。だから、ある程度の信頼関係が必要である。しかし、一方、あるサイトがほかのサイトのシステム管理者を完全に信頼するという可能性も低い。そこでUNICOREでは、遠隔サイトのシステム管理者の権限を、グリッドを操作するのに必要な最小限の範囲に制限するトラストモ

デルを採用している。要するに、システム管理者の権限を、エンドユーザによって署名され認可されたサブジョブを送出して回収するという操作に制限するのである。つまり、サブジョブの一時的保管を除けば、すべての信頼に関する課題は、各サイトとサブジョブに署名したエンドユーザの間での認可関係によって確立されたユーザサイトのトラストモデルに依存するということになる。

Globusに基づくサービス

富士通はまた、Globusに準拠したグリッド技術も日本で主導的に提供している。UNICOREとは異なり、Globusはグリッドコンピューティングのための完全な包括的環境を提供するものではない。Globusはむしろ、開発者が複雑で柔軟性のあるアプリケーションをグリッド環境内で作成するためのツールキットのようなものである。

Globusは、クライアント、ゲートキーパおよびGRAM(ジョブマネージャ)から成る、UNICOREと似たような3層構造を有している。注目に値する違いは、UNICOREがGlobusと比べてよりシームレスなコンピューティングをサポートすることと、Globusにはないクライアントに準じたワークフローをサポートすることである。

UNICOREがサポートするシームレスなコンピューティングというのは、サイトAにあるマシンX上で動作するように作成されたある抽象ジョブが、変更を加えることなくサイトBにあるマシンY上で動作できるということである。これが具体化データベース(Incarnation Database)と呼ばれるソフトウェアによって実現される。このソフトウェアは、UNICOREの抽象表現を、個別のマシンとサイトの仕様に特有の表現で、個別のマシンとサイトに固有な形に具体化(または翻訳)するのである。

多数サイトのサーバ間の複雑なインタラクション(ワークフロー)を作成する場合、ユーザはUNICOREとGlobusとの間で2種類の大きく異なるプログラミングモデルに直面する。UNICOREでは、クライアント上のビジュアルエディタを使って複雑なワークフローを簡単に作成することができる。ジョブステップ間の依存関係がユーザに視覚的に提供され、UNICOREサーバ(NJS)によって自動的に管理される。一方、Globusでの当該プロセスでは、ユーザは一つのジョブの最後にそのスクリプ

トの最終タスクとして別のジョブを作成しなければならない。あるスクリプト内にこれらの複数のスクリプトを作成するという複雑さに加えて、不連続のジョブステップの取扱いは可能ではあるがとても面倒である。UNICOREはまた、ジョブの条件付き実行とジョブ内ループのための内蔵グラフィカルサポート機能も備えている。

グリッド上の複数ユーザおよびサイト間のセキュリティに関するインタラクションはUNICOREとGlobusで類似しているが、Globus準拠のグリッドでは、サイト間のトラストモデルはUNICOREのものよりも厳しくないものになる。つまり、Globusでは個々のサイトが遠隔サイトのシステム管理者を信頼する必要があるのである。その状況では、あるサイトのシステム管理者が悪意を持ってほかのサイトのユーザを装う可能性が生じる。これはUNICOREでは起こり得ないことである。この問題に関する分析と詳細な比較は参考文献(3)で述べられている。

### 富士通のOGSAへのロードマップ

グリッドシステム全般に、とりわけ富士通のシステムが、ここ数年の間で能力を増加させてきた。さらに、グリッドシステムの能力が将来飛躍的に伸びることも確実である。この章では、現行のUNICOREで可能になったグリッド環境の使用事例を説明する。富士通は、OGSAへのロードマップに従ってこれらの現行サービスを継続的にサポートし、さらにOGSAが実現する新機能を後述するように徐々に追加していくであろう。

現行のグリッドでできることは

現在使用可能なグリッドは、オンデマンドのコンピューティングリソースという未来のビジョンに向かう道程を数歩進んだ程度のものである。まだ道は遠いが、現行のグリッドシステムも、コンピュータリソースの使用方法を変える多くの新しい能力を提供している。

現行のグリッド上の基本機能の一つは、スーパーコンピュータまたは大規模データセンタクラスのような高価値リソース上でバッチジョブを実行する能力である。バッチジョブは遠隔地から投入可能で、グリッドシステムが遠隔ユーザの認証と認可の処理をする。また、UNICOREは、実行ファイルの配置などのサイト固有の詳細項目の多くを抽出すること

で、異なるサイト上でのジョブ走行に関する問題を軽減している。バッチジョブ用のグリッドでは、ジョブの投入、監視および制御が可能である。

グリッドがバッチジョブをサポートするためには、ファイル転送も扱わなければならない。ジョブスクリプト、入力データおよび出力結果にはファイルが必要である。広範囲にわたるサイズのファイルの転送をサポートするために、様々なオプションが用意され、入力データや出力結果の数ギガバイトのファイルまで転送することができるようになっている。UNICOREの場合と同様に、データ転送はワークフローに組み込むことができるほか、Globusの最上位に構築されたGridFTPなどの個別のグリッド上アプリケーションでデータ転送を処理することもできる。

バッチジョブとデータ転送という二つの基本機能を組み合わせることで、斬新だが基本的に単純なコンピューティングリソースの利用方法を提供できる。まず、ユーザがどこに居てもリソースに便利にアクセスできる明白な利用方法である。さらに、少し高度な利用方法は、ユーザが機動的に広範囲の需要に対応できるリソースを使用できるようにすることである。今日では、一人のエコノミストが複雑な経済モデルを使ってイベントや政策変更の経済への影響を評価できる。国家予算の発表など重大な経済イベントの前段階で、エコノミストグループが数々のシミュレーションを行い可能な選択肢の影響を検証することで、それらに対する反応を確認することも可能となる。需要の変化は大きくしかも一時的であり、その期間が限られている（予算が決定されたとたんにシミュレーションは不要になってしまう）。短い需要のピーク時期だけのために多くのリソースを維持しておくのは経済的ではない。上記のようなシミュレーションをグリッドシステムで行えば、過剰な需要を複数の遠隔システムに安全にしかもシームレスに移行できる。

グリッドがサポートするデータ転送は単純なファイル転送ではなく複雑なものである。科学実験で大量の結果が生み出され、実験結果のデータが多くの異なる機関に属する研究者によって分析される場合を想定してみよう。グリッドシステムをこの実験結果のデータの配布に使用すれば、データのローカルコピーが使える所ではそれを使い、使えない所では

リモートコピーを使用するということが確実にできる。さらに、このグリッドシステムで、データの更新や修正などの変更を自動的に配布したすべてのコピーに反映することでデータの一貫性を確保できる。

現行のグリッドシステムでも、単一ステップのジョブを実行する以上のことができる。特定の順序で実行すべき複数ステップのジョブで構成されたワークフローの実行も可能である。各ジョブステップの結果が別のジョブステップに入力でき、先のジョブステップの結果に従って分岐やループを行うことで、ジョブステップの実行順序を変えることもできる。グリッド上のワークフローの各ジョブステップが異なるシステム上で実行でき、グリッドジョブが開始された後で条件が変われば、どのシステムでステップを実行するかを調整し直すことも可能である。

グリッドシステム上では、ユーザはどの場所からでもリソースにアクセスすることができるので、シミュレーションの進行過程にこれまでよりもっと直接的にかかわることができる。例えば、シミュレーションの進行中に中間結果を取り出して進捗状況をモニタしたり、進行中のシミュレーションにデータを送ってその進捗状況を変更したりできる。グリッドを使うと、一人の研究者が、あるシステムでアプリケーションを実行するように操作しながら、一方では別のシステムで（中間）出力データを視覚的に表示することもできるのである。

OGSIとOGSAによって可能になること

前述したように、遠隔でのサービス実行、リソース共有、過剰な作業負荷の除去、データ配布および複数管理ドメインにわたるアプリケーションの移動などがグリッドコンピューティングで既に可能になっている。そこで、OGSAの開発によって何が可能になるのかという疑問が生じる。この疑問に対する答えを二つの観点から述べる。まず、OGSI<sup>(1)</sup>で提供される拡張された基本機能を説明し、つぎにOGSIの上位に構築されるOGSA<sup>(4)</sup>の高水準の能力について検討する。

【オープングリッドサービスインフラストラクチャ (OGSI)】

OGSIは、その名称が示すように、上位に他サービスの構築を可能にする基礎的能力を提供する。OGSIが提供するサービス群は様々に組合せ可能で、各アプリケーションに固有の能力を供与するために

拡張することもできる。OGSIがサポートする主要サービスは、グローバルサービスネーミング、イントロスペクション(内観)、通知、動的サービス生成、ライフサイクル管理、サービス集約などである。

#### (1) グローバルサービスネーミング

グリッドサービスの存続期間の長さは可変であり、その結果として、プラットフォーム上で動作するあるグリッドサービスがそのプラットフォームよりも長く存在し続けることもある。このようなグリッドサービスの名前は、そのサービスを走行させているサーバのアドレスよりも長い間有効でなければならない。したがって、Webサービスが提供する「エンドポイント」能力ではこの要件を十分満たすことができない。さらに、グリッドサービスは、インターネット上の安全な接続パスや企業オフィス内のLANなどいく種類かのパスを経由してアクセス可能なものである。これらの要件をサポートするために、OGSIでは2段階のネーミング体系が提供される。あるサービスが作成されると、そのサービスを示すグリッドサービスハンドル(GSH: Grid Service Handle)が一つ作成される。そのGSHは当該サービスの存続期間中は有効であり、またほかのサービスを指し示すことはない(言い換えれば、GSHは再利用されない)。あるGSHを所有するクライアントは、そのGSHが示すサービス用にグリッドサービスレファレンス(GSR: Grid Service Reference)を作成するために、OGSIが定義する「リゾルバ」を要求することができる。このGSRはすべてのアドレス情報を含み、例えば当該サービスが新たなプラットフォームまたはサーバに移動された場合には有効期限が切れることもある。

#### (2) イントロスペクション(内観)

サービスが作成されると、そのサービス自体を潜在的なクライアントに向けて広告する必要が生じる。Webサービスの場合には、UDDI(Universal Description, Discovery and Integration)レジストリに当該サービスを登録するという広告方法がある。グリッドサービスもWebサービスの一つであるから、UDDIレジストリに登録できる。しかし、グリッドサービスの主要な属性は動的であるのに対して、UDDIレジストリには動的なサービス属性の記述を扱える能力がほとんどない(または全くない)。OGSIでは「サービスデータ」と呼ばれる概念が用

意されていて、例えば現在の負荷などのサービス属性がサービスデータとして公表される。クライアントは標準化されたOGSIインタフェースによって、これらの属性についての複雑な照会処理を簡単に行うことができる。

#### (3) 通知

サービスデータという概念の拡張機能として、グリッドサービスのサービスデータの変更通知を、そのサービスに関心を持つクライアントに送ることができる。クライアントはOGSIに含まれているサブスクリプションインタフェース(通知申込みインタフェース)を使ってサービス属性への興味の対象(または一群の属性への照会)を登録する。通知受信を申し込む際には希望する通知メッセージ受信の間隔を長くも短くも登録できるので、クライアントはメッセージトラフィックの過負荷に悩まされることはなく、当該属性が定期的に変化しない場合でも、当該サービスが存続していることを(通知メッセージによって)確認することができる。通知受信にもその種類に応じた有効期限があり、その期間は延長も可能である。使われなくなったサービスなどは最終的にメッセージ送信を停止して、特別なサービス管理の必要もなく、不要なネットワークトラフィックを減らすことができる。

#### (4) 動的サービス生成

Webサービスと異なり、グリッドサービスはカスタマイズできるサービスとなることを目指している。クライアントは、使いたいサービスの機能を微調整するために一群の属性をサービスに対して提供することができる。提供された属性はグリッドサービスファクトリへの入力となり、そのファクトリがクライアントの代わりにそのサービスを生成して展開する。

#### (5) ライフサイクル管理

すべてのグリッドサービスは、上述した通知サービスのように存続期限を有する。存続期限があることで、グリッド上で、クライアントがもはや関心をなくしたサービスの穏やかな自然消滅が可能になる。展開されたサービスを複雑な管理システムの支配下に置くことも可能だろうし、そのような管理はサービスを効率良く運用するために実際には望ましいことであるかもしれない。しかし、グリッドサービスではその必要はない。サービスの有効期限を理にか

なった長さに設定することで、サービスセンタ管理者は、管理下にあるすべてのサービスのために全面的な管理インフラストラクチャを配備する必要がなくなる。時がたてば、使用されないサービスは自動的にリソースを開放して終了する。

#### 【オープングリッドサービスアーキテクチャ (OGSA)】

OGSAは、GGF<sup>(4)</sup>が収集した使用事例に対応するために必要な、OGSI準拠のあらゆるサービスを包括する概念である。これらのサービス群の総体はかなり広範囲にわたり、またOGSIを最下層のアーキテクチャとして包含する。想定されているサービス群は下記のとおりである。

- ・ オープングリッドサービスインフラストラクチャ
- ・ 仮想組織
- ・ 発見
- ・ コレオグラフィ (振り付け), オーケストレーション (組織化) およびワークフロー
- ・ トランザクション (取引)
- ・ 計量
- ・ 格付け
- ・ 会計
- ・ 請求および支払
- ・ 導入, 展開および供給
- ・ 障害管理
- ・ 問題判別
- ・ 分散ロギング
- ・ メッセージングおよびキューイング
- ・ イベント
- ・ ポリシー
- ・ ジョブ管理
- ・ 基本データサービス
- ・ 高度データサービス
- ・ 包括的合意
- ・ 留保合意
- ・ データアクセス合意
- ・ ジョブキューイング
- ・ サービス管理

ここでは、各サービスの詳細を述べることはせずに、OGSA作業部会の仕事を動機付けているいくつかの使用事例を紹介する。

#### (1) UNICOREサービス

現行のUNICOREグリッド環境は、OGSAに基づくサービスの集合体として容易に実現できる。単純

な「発見サービス」でUNICOREのVサイトの可用性を公開し、「ジョブ管理サービス」でUNICOREジョブの作成、監視および制御をサポートし、「基本データサービス」でサイト間のファイル転送を実行することができる。このことは、UNICOREがグリッドサービスを実現するための単純な概念であるということの意味するのではなく、グリッドサービスのような多くの技術がグリッドの基礎構造の中に含まれているということの意味する。その基礎構造が配備されれば、その上位でサービスを提供することは比較的簡単なことなのである。

#### (2) 分散照会処理

特殊な生成パターンの事例では、複雑な分散照会処理の結果として一つのサービスが作成される場合もある。その場合、最初の照会と分散したデータソースのリストがファクトリでのサービス作成プロセスへの入力となる。クライアントは、このようにして作成されたサービスをその後の処理に使うこともできるし、協力者またはパートナーと共有することもできる。

#### (3) 動的仮想組織

グリッドには新しいビジネスモデルを創造する能力がある。それがグリッド実現への強い動機の一つとなっている。仮想組織 (VO: virtual organization) という概念が、新ビジネスモデルのための基礎的構造となる。一つのVOは、ある問題の解決またはビジネス機会の創出の目的で合意した個人または企業の集合である。このようなVOは存続期間も短く、適用範囲も非常に限られ、必然的に小規模のものになる。このことを可能にするには、そのVOに参加するメンバーが合意するための、事前に定義された一連の規約が必要となる。「包括的合意」サービスによって自動的に交渉が行われ、契約が締結され、作業が行われ、その後そのVOは解散する。これらすべてのプロセスが動的に、おそらく数秒間の内に実行される。

この概念で可能になるビジネスモデルには次のようなものがある：

- ・ 企業間取引用インターネットサービスまたはグリッドサービスが公言するサービス内容を実行することを保証する保証人
- ・ 例えば製薬会社が利用可能な他社の技術を自身の技術を明かさずに調査するための知的財産情報セ

ンタまたは仲介人

- ・ある企業が自身のサービスやリソースを他企業に売ることによって自身の業務外注資金を調達する、新たなレベルのアウトソーシング
- ・企業内または複数企業にわたる電子的「チャイニーズウォール」サービスの提供<sup>(注)</sup>
- ・メンバが有料で互いの存在と互いに取引を行う条件を発見するための仮想組織の「交流」サービス。「交流」サービスではなく、「ビジネスマンクラブ」としてモデル化することもできるだろう。
- ・動的に取り決められた契約の仲介人

#### (4) ITインフラストラクチャ管理

OGSAの「サービス管理サービス」はサービス群を管理、供与および制御するための標準化されたインタフェースを提供する。ある企業のITインフラストラクチャがグリッドサービスである場合(またはそのインフラストラクチャがWebサービスよりもグリッドサービスに適した能力を備えている場合)、上記インタフェースによってその組織内で様々な異種リソースおよびサービスの均一管理のための枠組みを得ることができる。合併や買収の影響を受ける大企業や、ITインフラストラクチャに関してクライアントが要求する多様な選択を管理する必要のあるITサービス会社にとっては、この管理サービスはとりわけ重要である。

### む す び

本論文執筆時点で、OGSA準拠バージョンのUNICOREはすべての第1世代グリッド機能をグリッド/Webサービスインタフェースによって提供している。相互運用可能なブローカーサービス、通知機能および基本的な仮想組織のサポートなどのいくつかの拡張第2世代機能についても、開発が進められている。

グリッドコンピューティング全般およびUNICOREについての詳細情報の検索を開始するのに適した組織およびプロジェクトは、下記のとおりである。

UNICOREフォーラム ([www.unicore.org](http://www.unicore.org)) は、

UNICOREプロトコルと規格の管理、および推進機構である。

ドイツIntelは、富士通と共同で、産業対応の無料試用可能なUNICOREプロトコルとデモンストレーション用グリッドを ([www.unicorepro.com](http://www.unicorepro.com)) で提供している。

EuroGridおよびGRIPというEUの2大研究プロジェクトは、UNICOREに基礎を置く研究を最近終了し、日本で新たな研究プロジェクト(NAREGI)が開始された。 ([www.eurogrid.org](http://www.eurogrid.org), [www.grid-interoperability.org](http://www.grid-interoperability.org), [www.naregi.org](http://www.naregi.org))

GGF ([www.ggf.org](http://www.ggf.org)) は、標準化機関と標準研究フォーラムを組み合わせた組織であり、そこでは最先端の概念と産業対応の発想が公開フォーラムの形で共有されている。

Webサービス仕様に関する活動の情報は ([www.w3s.org](http://www.w3s.org)) および ([www.oasis-open.org](http://www.oasis-open.org)) で参照することができる。

#### 参考文献

- (1) S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, K. Kesselman, T. Maquire, T. Sandholm, D. Snelling, and P. Vanderbilt : Open Grid Services Infrastructure (OGSI) v. 1.0 .
- (2) D. Erwin and D. Snelling : UNICORE: A Grid Computing Environment . LNCS 2150 , Euro-Par 2001 , Springer , 2001 .
- (3) T. Goss-Walter, R. Letz, T. Kentemich, H-C. Hoppe, and P. Wieder : An Analysis of the UNICORE Security Model .  
<http://forge.gridforum.org/projects/ggf-editor/document/GFD.19/en/1>
- (4) OGSA working group .  
<http://forge.gridforum.org/projects/ogsa-wg/>

(注) [www.investorwords.com](http://www.investorwords.com)によれば、「証券会社において内部情報の違法使用を防ぐ目的で非公開の重要情報へのアクセスを制限するために部門を強制的に分離するための手段」を意味する言葉。