The Grid-Idea and Its Evolution

Gregor von Laszewski
Argonne National Laboratory, Argonne, IL 60439, U.S.A.
gregor@mcs.anl.gov
Preprint number: ANL/MCS-P1253-0505*

Abstract

In this paper we review the essence of the *Grid-Idea*. Specifically, we explore the changing definition of the Grid and follow its evolution over the past decade. This evolution is motivated by the gradual expansion of management issues that must be addressed to make production Grids a reality and to meet user requirements for increased functionality. Additionally, we focus on the evolutionary path of the Globus Toolkit taken to address the increasing needs of the community. We also discuss the evolutionary inclusion of commodity technologies as illustrated by the Java Commodity Grid Project.

1 From Single Computer to the Grid

Grid computing has become a valuable asset in scientific and business communities for integrating distributed resources needed to match increased requirements in managing computations and data [41, 46].

We have seen the development of single computers, mainframes, vector computers, parallel computers, distributed computers, and metacomputers and, most recently, the emergence of the Grid infrastructure. With these developments several issues have arisen: (1) the ever increasing need of scientists to gain access to massive computing power; (2) the recognition that the most powerful computers, while useful for solving complex problems, may be too expensive for an individual computing center or organization to own; (3) the realization that connecting such resources as part of a persistent infrastructure is a challenging management issue; (4) the need for integrating noncomputational resources into the Grid; and (5) the importance of standards to promote interoperability among participants.

Most recently, these same issues – identified initially by the scientific user community – have been recognized also by the business community. Hence a considerable effort is now in place by both communities to drive the development of the next generation of Grid software.

This chapter is organized as follows. First, we present the terminology related to Grids and discuss its changes. Next, we follow the evolution of Grid standards and Grid technology, focusing on the Globus Toolkit, the CoG Kit, and emerging Web services. Finally, we summarize our findings and make predictions about future trends in Grid computing.

2 Evolution of Grid Terminology

The widespread use of the term Grid can be clearly attributed to the publication of the book *The Grid: Blueprint for a New Computing Infrastructure,* edited by Ian Foster and Carl Kesselman. Although the term was introduced in a proposal to NSF by a consortium of institutions including NCSA and Argonne in 1996, the term was not widely used until the book was published in 1999.

In the past five years, the definition of a Grid has undergone several evolutionary steps. We start by recreating these steps and then identify what the term Grid means today.

2.1 The Early Days

Grid computing has been envisioned since the very early days of computer science. Already in 1969 we find references that introduce the vision of a computational Grid infrastructure [33]:

> *We will probably see the spread of computer utilities, which, like present electric and telephone utilities, will service individual homes and offices across the country.*

This vision has been gradually implemented since then, in part in more traditional infrastructures. Because a computational infrastructure is expensive, the development of computing centers was an obvious way to create facilities that can be used and shared by many users. The problem with such centers, however, was that the users – at least in the early days – had to travel to these centers; moreover, only a few users were typically given access to the computers.

Five developments started to change this situation. First was the introduction of timesharing facilities and operating systems, which enabled multiple users to share the resources. Second was the introduction of networks and remote terminals, which allowed the centers to give users at remote locations access to the facilities. Third was the introduction of commodity hardware and distributed systems, which greatly reduced the cost of expensive resources. Fourth was the creation of standards through, for example, the IETF. And fifth was the creation of the Internet, which made it possible to easily link multiple resources. Sharing of hardware resources became well established by the mid-1990s, and a number of research efforts tried to exploit the new-found capabilities by providing software and tools for community use [13, 14, 6].

2.2 The Metacomputer

Some research scientists reached the stage where the compute infrastructure available at one center was not sufficient: either the available resources were oversubscribed, or more resources were needed to tackle the problem. Other scientists encountered a different challenge: the growing number of different platforms. For example, while a climatologist may have developed his code initially on a vector computer, he may have developed later parts of his code on parallel MIMD machines. Many scientists found it difficult to keep up with the rapid development of the compute infrastructure. Hence, some were enticed to run one part of the code on a newer machine and another on a "traditional" machine for which the code had been optimized. Many scientists simply did not have the time to translate their codes to other platforms. Thus, a new paradigm for computing was needed.

These factors all influenced the creation of what is termed a *metacomputer*. Around 1987 Larry Smarr used the term metacomputing to describe the concept of a connected, coherent computing environment [49, 50]:

> *The metacomputer is, simply put, a collection of computers held together by state-of-the-art technology and "balanced" so that, to the individual user, it looks and acts like a single computer. The constituent parts of the resulting metacomputer could be housed locally, or distributed between buildings, even continents.*

Here the emphasis is on *single computer*. This view is further emphasized by another quote from a presentation that Smarr published [50]:

> *Whether logging in from your laboratory or office PC, or via a wireless laptop, you'll tap into the computational might of several or many computers, possibly hundreds or thousands of miles away—just as easily as you reached this exhibit.*

2.3 The Grid

Influenced by research in parallel and distributed computing in the early 1990s and using metacomputing as a stepping stone, the term "Grid" gradually became popularized. The term was first formalized in 1999 in the book *The Grid: Blueprint for a New Computing Infrastructure* [16]:

> *A computational Grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.*

This definition focuses on a hardware and software infrastructure. Earlier that same year, however, von Laszewski had projected that the community was in need of an *extended infrastructure* including a "shared computing infrastructure of hardware, software, and knowledge resources." "Knowledge resources" encompass not only databases but also humans [65]. The implications of this distinction are important. It projects an integrated vision of an environment in which the human is integral part of the available resources. We believe strongly that future definitions of Grids will include this vision. Indeed, recent modifications to the original[16] definition are seen in [17], in which Foster refined the earlier definition of Grids to include resource sharing among a set of participating parties.

We observe that the term Grid is often used interchangeably in the community to refer to an *idea* and a *physical instantiation* of a hardware infrastructure. To distinguish better between these two uses of the term Grid, von Laszewski defined the term "Grid approach" in 2000 [62, 71, 63]. Specifically, he observed that Grid computing promotes an approach to conducting collaborations that does not stop at the hardware and software but that enables a vision for a shared infrastructure including humans leading to the following definition[63]:

> *We define . . . the* Grid approach*, or paradigm, that represents a general concept and idea to promote a vision for sophisticated international scientific and business-oriented collaborations.*

We note that this definition is not limited to scientific use of the Grid, but includes business collaborations. An often overlooked fact is that, at about the same time that the Grid was gaining followers in the scientific community, the idea was being heavily discussed also in the business community. The natural consequence was that, today, Grids are interesting not only for scientific applications but also for business scenarios. A natural overlap exists.

One Grid or Many. During the early days of Grid computing it was often assumed that the infrastructure targeted a persistent management model. That is, a Grid would be built and users would become members of this Grid to conduct their tasks. Indeed, before 2000, some envisioned the creation of a single Grid. It soon became clear, however, that such a model was not sustainable. What was needed was the introduction of various communities managed through the creation of a virtual entry in an information service [64]. This new model was first demonstrated during Supercomputing 1998, where multiple user communities were managed through the Metacomputing Directory Service [64]. Foster's introduction of the term *virtual organization* [21, 24] in relationship to Grids was a much-needed abstraction to formulate issues related to building Grids more precisely.

We can now present the definition of a production Grid:

> *A production Grid is a shared computing infrastructure of hardware, software, and knowledge resources that allows the coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations to enable sophisticated international scientific and business-oriented collaborations.*

Ad Hoc Grids. One of the issues that will become especially important in future production Grids is the management of sporadic and ad hoc communications [67, 68]. In a Grid infrastructure, parts of the resources undergo patterns that are time-limited: that is, resources may be available for only a short time. Further, a production Grid and its associated virtual organizations need to be instantiated on short notice. Hence, future Grid approaches must not only provide services as part of a virtual organization but must address the creation and management of production Grids in an ad hoc fashion:

> *An ad hoc Grid provides a production Grid that addresses management issues related to sporadic, ad hoc, and time-limited interactions and collaborations including the instantiation and management of the production Grid itself.*

3 Evolution of Grid Standards

The nature of the Grid is to enable a shared infrastructure. Therefore, it is important that standards be developed to foster the contribution and integration of resources in a uniform fashion. Organizations such as IETF and OASIS have demonstrated how successful such efforts can be.

3.1 Workshops

The development of standards requires gathering information about community needs and available solutions. The first such coordinated effort to help gather this information for the Grid was a series of three workshops entitled *Desktop access to remote resources (Datorr)*. The goals of these workshops were to (1) collect the most recent information about the status of current projects accessing remote resources, (2) derive a strategy on how remote access to resources can be supported, and (3) bring the research community in this area together to initiate dialog and collaboration. The first meeting took place October 1998.

The Datorr workshops (which were later continued as the Grid Computing Environments research group in the Global Grid Forum) were later supplemented by other venues such as Birds of the Feather meetings at the annual Supercomputing conference. Such efforts were useful in building a community of users interested in Grid computing.

3.2 Global Grid Forum

To facilitate the creation of best practices and standards, a group of Grid users and developers from the United States formed the Grid Forum in June 1999. By the end of 2000 members from Europe and Asia joined the effort to form the Global Grid Forum (GGF) [26]. The first meeting was held March 2001. Subsequently, a strong influence by commercial companies elevated the role of the GGF beyond that of an academic community. Today the GGF includes users, developers, and vendors with the interest to define a global standardization effort for Grid computing. Thousands of individuals in industry and research, representing over 400 organizations in more than 50 countries, have participated in GGF activities. Three meetings are held annually, with recent meetings attracting over 500 participants (see Figure 1).

At present, community-initiated working groups are developing best practices and specifications in cooperation with other leading standards organizations, software vendors, and users. The GGF summarizes its mission with the following two items: (1) defining Grid specifications that lead to broadly adopted standards and interoperable software and (2) building an international community for the exchange of ideas, experiences, requirements, and best practices. The GGF has been successful in both categories.

The relationship to other standards bodies is an important part of the current GGF strategy, as it allows introducing standards to a larger community including the Web services community. Good examples for efforts that are under way are standards activities to promote the use of the Grid Security Infrastructure [7] as part of the IETF and the promotion of the WS-RF related standards as part of OASIS. One of the important achievements of the GGF is the definition of the Open Grid Service Architecture (OGSA) [23]. The initial efforts to also define the Open Grid Service Infrastructure shifted quickly to the definition of the Web services framework that offers the potential of a convergence between Web services and Grid services architectures. WS-RF has received backing from a number of academic and industrial partners. First implementations such as the Globus Toolkit version 4 are already based on it.

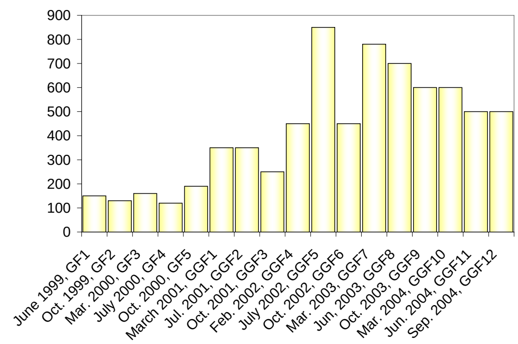

Figure 1: Number of participants at the various Global Grid Forums.

4 Evolution of Grid Software

The evolution of Grid software naturally mirrors the evolution of the Grid-idea and the standards-driving activities discussed earlier. Software for the Grid needs to deal with the various management challenges that the Grid offers. These challenges are often addressed through the introduction of a layered Grid architecture that spans from the Grid fabric to the application layer. The Grid fabric contains protocols, application interfaces, and toolkits to enable the development of services and components to access locally controlled resources, such as computers, storage resources, networks, and sensors. Recently, however, it became clear that the original vision of producing APIs or object models must be augmented with a service-oriented architecture in order to integrate with the commodity technologies developed by the Internet community. This reuse of commodity technology will have a big impact on the reusability and adaptability in other communities.

Grid software has four specific goals: (1) expose the Grid to users in such a fashion that the users may not be aware that they are using it, (2) develop software that allows the Grid application developer to easily integrate applications into the Grid, (3) develop software to allow the creation of a sophisticated Grid middleware infrastructure to be reused by the application developer, and (4) develop software that interfaces between commodity software and hardware that constitutes the Gird fabric and are exposed in easyily to the developers.

Looking at the current trend in developing Grid middleware, one might get the impression that all Grid software develops Grid services, as seen in the example of WS-RF enabled services. However, this is not true. While the development of such services is part of the Grid middleware, the services must still be implemented through common software engineering practices to expose the logic. Hence, in addition to Grid services there is a need to produce schemas, tools, APIs, object models, and programming frameworks that make the end-to-end application solution a reality (see Figure 2) [71].

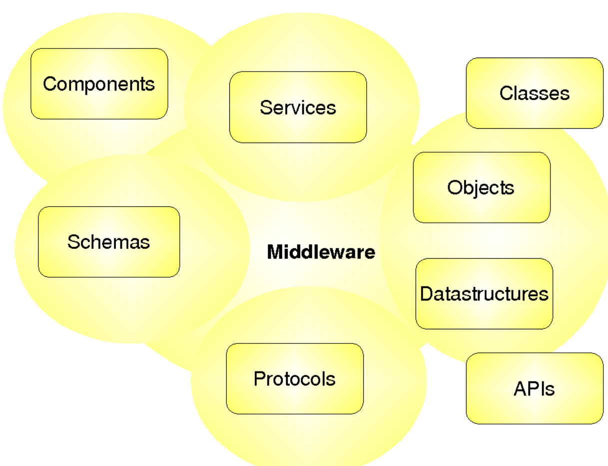

Figure 2: Grid middleware needs to support a variety of technologies in order to accommodate a wide variety of uses.

Some of the technologies being focused on at this time are the development of gateways, portals [40, 59], workflow systems, and Grid middleware systems [34, 9, 34, 52, 39]. In the following subsections we focus on the evolution of two Grid software systems: the Globus Toolkit and the Java CoG Kit. For a discussion of other Grid software, see [16, 17, 5, 42].

4.1 Globus and the CoG Kit

Figure 3 depicts the evolution of the Globus and Java CoG Kit needed to meet the requirements of Grid users. We discuss in this section how two toolkits have addressed these requirements.

4.1.1 Origins

The Globus Toolkit was originally addressing issues related to Metacomputers [15]. The origins of the Globus Toolkit can be traced back to the development of the Nexus multithreaded communication library as used within the I-Way [57, 20, 14]. Nexus has provided the communications module to the Globus Toolkit [19]. Nexus was intended to provide a portable communication library that could be easily integrated in compilers for parallel computing. It provided a global memory model, asynchronous events, and thread safety. Different communication protocols were hidden in the library and the protocol with potentially the best performance could be predefined. Hence, the initial philosophy within the Globus Project was to develop a set of standard libraries that allow communication, resource management, information management, and authentication. This philosophy was quickly augmented with services such as the Grid resource access manager and the Metacomputing Directory Service. More advanced libraries, such as the Dynamically-Updated Request Online Coallocator (DUROC)—which to this day allows coscheduling in MPICH-G [32]—relies on the Nexus API wrappers [10]. As there is no protocol compatible implementation available other than through calling the C-API, however, its functionality is not easily exposed as a service. Plans for doing so have been initiated with the latest Globus Toolkit version 4.

The Java CoG Kit [66] had its origin in a metacomputing system developed at Syracuse University that was able to submit a number of compute jobs to a set of compute resources including the North East Parallel Architectures Center at Syracuse University, Goddard Space Flight Center, and Maui Supercomputing Center [60, 69]. The system was created out of the need to run a parameter study as part of an integrated data analysis of a climate model research project. At the time, the queue wait time of jobs submitted to Goddard's compute facilities resulted in multiple hours of wait due to constant oversubscrip-

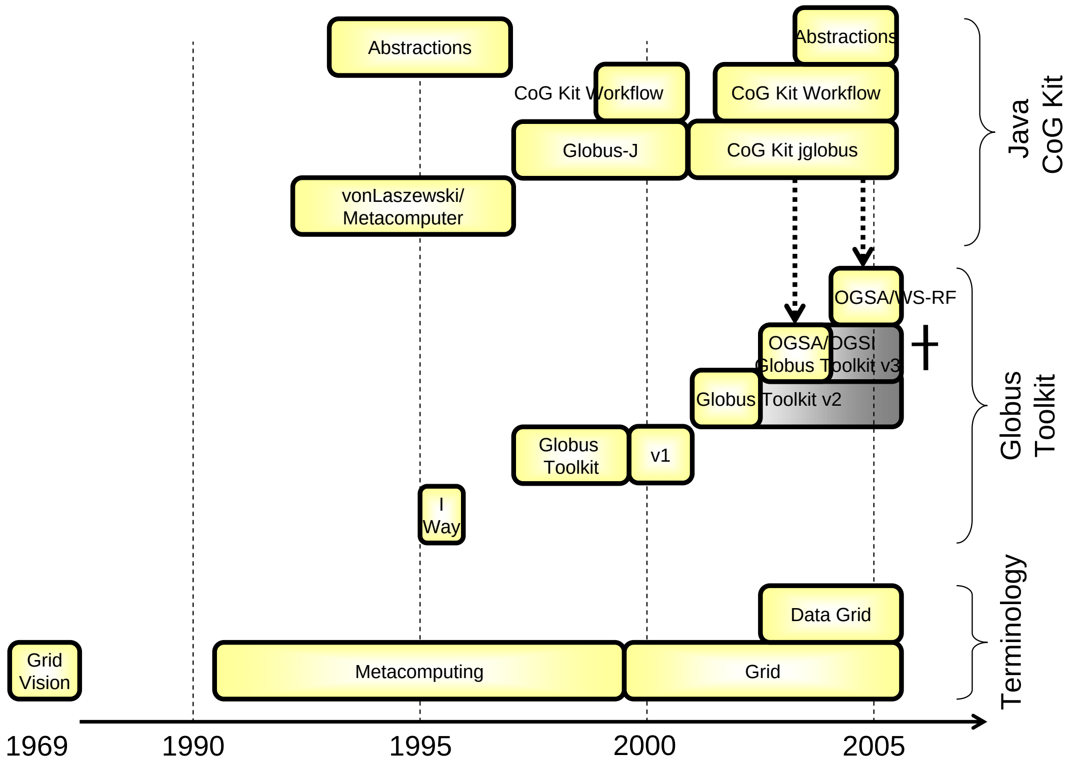

Figure 3: General History

tion. The goal was to reduce the wait time dramatically by increasing the number of available resources. The system to coordinate the resources implemented a dataflow engine and allowed visual workflows with dependencies to be executed and monitored at the different resources. An important factor was an abstraction mechanism that was introduced to interface with the diverse queuing systems. Additionally, it was possible to ingest application notifications that allowed the monitoring of the progress of the calculation. A prediction service was able to select resources based on previous experience [61].

In 1996, the author of the precursor to the Java CoG Kit joined the Globus project two months after the Globus project was initiated. The development of the metacomputing system was suspended.

The software engineering design philosopy of the Globus Toolkit and that of the metacomputer were different. While Globus proposed to develop from the bottom up, the metacomputer system provided a high-level abstraction inspired by a top-down design. The top-down design philosophy and the abstractions did subsequently influence the development of the Java CoG Kit, which was in turn partially integrated in the Globus Toolkit (version 3 and 4).

4.1.2 Protocols and Services

The various activities within the Globus Project—such as the creation of an information service that can be accessed not only with a C API, the development of a security protocol [7], the development of a low overhead heartbeat monitor [53], and an execution service—were all major influences leading to the realization that many of the Globus components must also be expressed as services and not only as APIs. This fact was even more evident after native Java wrappers to the C toolkit were implemented. Based on an analysis and the communication with multiple developers in the community, the members forming later the Java CoG kit team concluded that just providing a wrapper resulted in the loss of significant functionality of the Java framework. The code was cumbersome to write for Java programmers and provided no improvement over the C implementation [35]. The team believed this was also true for the Java implementation of Nexus [25] although it was used by former I-Way community members. From the point of view of a Java programmer, it was clear what needed to be done: future developments within the project should be formulated as services with associated protocols; these services should be included in the most natural way into the Java framework.

This refocusing on protocols and services had another advantage. As the Java group reverse-engineered the Globus protocols, the group was able to testing against an independent implementation. The result improved the overall implementation quality of the C toolkit.

The result of this development were two products: the C-based Globus Toolkit (version 2), and the Java-based Java CoG Kit. The term CoG Kit is short for *Commodity Grid Kit* indicating the fact that reuse of commodity technology together with Grid concepts lead to a development software kit; just like the C-based Globus toolkit did for C. While the C-based toolkit provided the de facto standard for Grid middleware, the Java CoG Kit demonstrated that such middleware can be included in other frameworks and hence added significant value in the combination of the Grid functionality. It became the de facto standard for accessing Globus in Java. During this time other CoG Kits were developed for Perl [55], CORBA [43], Python [45], and Matlab. The CORBA and Matlab ports used the Java CoG Kit as a bridge. A large number of projects used both the Globus Toolkit and the Java CoG Kit to develop more advanced Grid-based software.

4.1.3 Production Grids

The availability of a software base made it possible to develop higher-level applications and frameworks. In order to prove the validity of the software, testbeds were set up. One of the initial testbeds was GUSTO [29], which won the best of show award at Supercomputing 1998 while demonstrating several Grid applications. The focus application was the execution of a microtomography experiment that showed convincingly that Globus technology can be used to enable remote collaborations. In this experiment a number of users communicated via advanced display devices distributed at Argonne and Florida (the conference site) to analyze data gathered in real time from Argonne's Advanced Photon Source. Real-time visualizations were produced of data manipulated on supercomputers and graphics pipelines at Argonne's computing center. The requirements identified during this experiment [70] continue to drive many of the developments of the Globus Toolkit today; see, for example, [63] for a discussion of challenging management issues.

The next evolutionary step in the development of testbeds and applications was the definition of Grids for large user communities as projected by the NASA Information Power Grid [30] and the high energy physics experiments conducted at CERN [11] and elsewhere [44, 31, 28]. The latter motivated the introduction of replica location services [3] that allow the duplication of data as a set of hierarchical organized information services. Additional projects in Asia and Europe resulted in the creation of national and international Grid efforts [4, 36].

A key step in the development of ambitious production Grids was the design of the TeraGrid [54]. This NSF-funded project has as its goals to build and deploy the world's largest, most comprehensive, distributed infrastructure for open scientific research. Currently, nine institutions are combining their expertise to make such a challenging effort a success.

The introduction of the service model into the Grid and its standardization effort within the Global Grid Forum led to the inclusion of the business community and the development of tools that specially targeted the issue of stability and sustainability of such an environment. Therefore, as part of the Globus toolkit development at Argonne and elsewhere, great emphasis was placed on performance and stability improvements of the core services. The result is the Globus Toolkit 4. The next generation of production Grids will benefit from these improvements. A selected number of well known production girds are listed in Figure 4 [37, 47, 30, 11, 48, 56, 12, 36, 27].

4.1.4 Applications and Portals

The original applications demonstrating the feasibility of the Grid were small case studies. With the increased availability of usable Grid middleware projects, however, more ambitious Grid applications were targeted (for example, by the HEP community [27]). One of the issues that arose was the difficulty in developing or integrating these services into a Grid application. At the same time the Internet was firmly established, and scientists expected to expose their data and applications through portals.

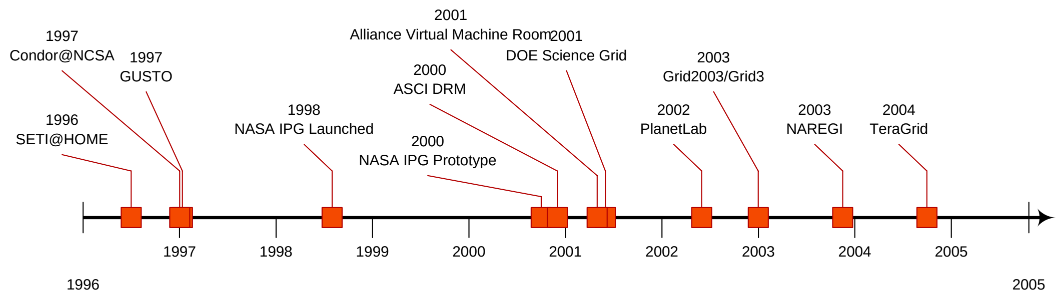

Figure 4: An approximate number of a subset of influential production Grids.

Because a lot of the work in the Internet community was driven by Java, it was natural to explore the development of Java portals to the Grid [40]. The Java CoG Kit provided a convenient mechanism for the development of Grid portals; in fact, a number of portal users [38] did not even know that they were using the CoG Kit to access the Grid. Most of these portals initially focused on the development of management tools to administer general job queues, as was common for compute centers. Once it was clear that application developers could develop their own portals, more application-specific portals were developed.

One of the significant problems application developers faced was the evolving Grid software. Since the APIs, the protocols, and their services wer not yet standardized and since significant changes in Grid software were being made, the application community faced a problem: the effort to upgrade between different software versions was extremely high. One of the solutions to this problem was again provided by the Java CoG Kit. To date, together with the community, the Java CoG Kit is defining a set of simple abstractions that fulfills the basic need of many Grid application users to execute a task remotely. Some of the users of this software were able to switch between Globus Toolkit versions, without changing the interfaces in their application code over multiple years.

Today, this concept has been enhanced to include an extensible XML-based workflow language that allows application users to integrate self-defined scripts into an extensible workflow. Moreover, through the availability of a functional representation of the workflows and the availability of high level abstractions in form of Java interface, no knowledge about XML is necessary to use the Grid. Prototype tools such as a Grid desktop and the introduction of the concept of Gridfaces, that allow to abstract views between a portal and a java application, will provide significant practical and conceptual methodologies to elevate the use of Grids within a computer desktop. Additionally, the simple integration of the CoG Kit into Matlab can provide an easy bridge between analysis tools used as part of the scientific discovery process and the Grid.

4.2 Grid and Web Services

The introduction of the services model in the Globus Toolkit was further discussed within the project the summer of 2001, while the creation of a Web services-based commodity Grid toolkit was proposed by the Java CoG Kit team. As a precursor to this activity, a system called InfoGram was developed that projected a merger of information services and job submission services. The idea came from the observation that standard Web services issued queries in the form of URL requests to HTTP servers. Hence, the CoG group concluded that a job submission, a filetransfer, and an information query should not be treated differently and that they present a *task*, or job, to be executed by a service. The task is invoked in all cases through a query to a service. The prototype implementation was not based on Web services but on an enhanced Java CoG Kit job submission server that allowed backwards compatibility with the Globus GRAM protocol. The prototype was successful, demonstrating that the architecture had merit and that Java was a viable player to implement the next generation of prototype Grid software.

Shortly thereafter, others in the project focused on the definition of the concept of stateful Web services, called Grid services.

4.2.1 OGSA

As part of the evolutionary process, an Open Grid Services Architecture (OGSA) was proposed that defines the basics of a Grid application structure applied to any grid system [23]. OGSA defines and recommends what Grid services are, what they should be capable of, and what technologies they be based on. However, OGSA does not make explicit recommendations to the technicalities of the implementation of the specification. It is used to classify the ingredients of a Grid system.

4.2.2 OGSI

The concepts presented within OGSA are formally specified by the Open Grid Services Infrastructure (OGSI) [58]. It defines how work is managed and distributed and how service providers and Grid services are described [18]. It specifies a Grid service to be a Web service that conforms to a set of conventions expressed as interfaces and behaviors that define how a client interacts with a Grid service. By using extended WSDL [8] and XML Schema [1], OGSI defines an implicit component model. It is centered on the concepts of stateful Web service instances, common metadata and inspection, asynchronous notification of state changes, references to instances of services, collections of service instances, and service state data declaration. It was hoped that OGSI would be adopted not only by the Grid community but also by the commercial and Web communities. However, because of issues with the use of schemas and its conceptual differences from other Web services-related specifications, it was not widely adopted [22]. Although projects still use OGSI today, its development has virtually halted in favor of a new specification.

4.2.3 WS-RF

In January 2004, a new proposed specification was published that replaces OGSI as part of the lowest layer of the Open Grid Services Architecture (OGSA) specification [72]. The specification was a re-factoring of OGSI resulting in a set of Web Services specifications that is much more closely related to existing Web services and allows integratation of web services directly into the Grid fabric.

The consequences from this development were summarized in [51] as "bumps in the road that led to a much better starting point with respect to obtaining wider community acceptance as well as rapid development of usable tools and Grid applications." such as the Java CoG Kit have been successful in hiding these "bumps" to their user communities due to their higher level interfaces. By now the WS-RF specification has been drafted and a number of other related specifications followed suit. We expect that as part of the standard process, changes still could happen and that the intermediate use of higher level tools, as projected through portals, workflows, and commodity Grid Kits, will promote the gradual replacement of the older technologies while at the same time not worrying about eventual additional changes to the standard.

5 Evolution of Collaboration

Collaboration is an essential part of human interaction. Grids can be viewed as a tool to help in improving collaboration through shared virtual organizations. In order to make this happen agreements must be put in place that govern the extent to which sharing of resources is allowed. Such sharing is not limited to the instantiation of a program: considerable human interaction is necessary to agree on the policies under which such automated systems operate. Once these policies are in place and systems on the Grid are developed and deployed, scientists hope to enable what we term the *Zweistein effect*, a play on the name Einstein. Instead of having a single scientist working alone in his laboratory, we strive to develop an infrastructure that allows the unique capabilities of a number of scientists to be integrated. What if we could create a Zweistein effect, a Dreistein, or even an N-stein, by having N researchers communicate with each other?[1]

Technologically projects such as the Access Grid [2] and commodity tools for ubiquitous communication will improve this situation. We as scientists must, however, learn how to manage these communications and build better tools as part of the Grid environment to support this quest.

6 From Evolution to Revolution

We have analyzed the evolution of the term Grid and identified its underpinning ideas. The Grid is a natu-

[1] Eins, zwei, drei are the German numbers for one, two, three.

Hardware trends

Single computer
⇓
Multiple computers
⇓
Vector computer
⇓
Parallel computer
⇓
Distributed computer
⇓
Metacomputer computer
⇓
Production Grid*
⇓
On-demand resources and services

Software trends

Sequential programming
⇓
Batch processing
⇓
Vector languages
⇓
Parallel Processes
⇓
Message passing
⇓
Parallel datastructures
⇓
Parallel objects
⇓
Metacomputing software
⇓
Grid middleware*
⇓
Grid upperware

Operational trends

Single computer in laboratory
⇓
Compute center
⇓
Volunteer computing
⇓
Shared metacompute center
⇓
Virtual organization* and Peers
⇓
Adhoc Grids

Figure 5: Several trends have in parallel contributed to the development Grid computing. Items marked with * indicate items related to Grid computing today.

ral evolution originating from several trends in computer science (see Figure 5). We identified a number of significant projects that influenced the definition of the term that was popularized in 1999. Since then, numerous activities have been undertaken to build software and hardware infrastructures that constitute a Grid.

These prototypic efforts quickly matured and have currently reached a state that allows reuse even by business companies. However, this is not the end in the development of Grids. We, as a research group, feel that we are in the initial stages, just as the Internet was a decade ago. We believe that the Grid-idea of enabling collaborative environments with shared resources is fostering an evolution not just in software and Grid middleware development but also in how state-of-the-art scientific discovery and business oriented management processes are handled. Our hope is that the availability of such an infrastructure will enable revolutionary quantum leaps based on the integration of scientific and other communities.

Acknowledgments

This work was supported by the Mathematical, Information, and Computational Science Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-Eng-38. DARPA, DOE, and NSF support Globus Project research and development. The Java CoG Kit is supported by DOE SciDAC and NSF NMI.

References

[1] XML Schema, Primer 0 - 3, 2001. Available from: `http://www.w3.org/XML/Schema`.

[2] The Access Grid. Web Page. Available from: `http://www-fp.mcs.anl.gov/fl/accessgrid/`.

[3] Bill Allcock, Joe Bester, John Bresnahan, Ann L. Chervenak, Carl Kesselman, Sam Meder, Veronika Nefedova, Darcy Quesnel, Steven Tuecke, and Ian Foster. Secure, Efficient Data Transport and Replica Management for High-Performance Data-Intensive Computing. In *MSS '01: Proceedings of the Eighteenth IEEE Symposium on Mass Storage Systems and Technologies*, page 13, Washington, DC, USA, 2001. IEEE Computer Society.

[4] ApGrid: Partnership for Grid Computing in the Asia Pacific Region. Web Page, 2001. Available from: `http://www.apgrid.org/`.

[5] Fran Berman, Anthony Hey, and Geoffrey Fox, editors. *Grid Computing: Making the Global Infrastructure a Reality*. Number ISBN:0-470-85319-0. John Wiley, 2003.

[6] D. Bhatia, V. Burzevski, M. Camuseva, G. C. Fox, W. Furmanski, and G. Premchandran. WebFlow - a visual programming paradigm for Web/Java based coarse grain distributed computing. *Concurrency: Practice and Experience*, 9(6):555–577, 1997.

[7] R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, and V. Welch. A National-Scale Authentication Infrastructure. *IEEE Computer*, 33(12):60–66, 2000.

[8] Roberto Chinnici, Martin Gudgin, Jean-Jacques Moreau, and Sanjiva Weerawarana. Web Services Description Language Version 1.2, July 2002. W3C Working Draft 9. Available from: `http://www.w3.org/TR/2002/WD-wsdl12-20020709/`.

[9] Condor: High Throughput Computing. Web Page. Available from: http://www.cs.wisc.edu/condor/.

[10] K. Czajkowski, I. Foster, and C. Kesselman. Co-allocation Services for Computational Grids. In *Proceedings of the 8th IEEE Symposium on High Performance Distributed Computing*, 1999.

[11] The DataGrid Project, 2000. Available from: http://www.eu-datagrid.org/.

[12] Doe Science Grid. Web Page. Available from: http://www.doesciencegrid.org/.

[13] FAFNER. Web Page, Syracuse University. Available from: http://www.npac.syr.edu/factoring.html.

[14] I. Foster, J. Geisler, W. Nickless, W. Smith, and S. Tuecke. Software Infrastructure for the I-WAY High Performance Distributed Computing Experiment. In *Proc. 5th IEEE Symposium on High Performance Distributed Computing*, pages 562–571, 1997. Available from: ftp://ftp.globus.org/pub/globus/papers/isoft.pdf.

[15] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, 11(2):115–128, 1997.

[16] I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, July 1998.

[17] I. Foster and C. Kesselman, editors. *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, 2004. second eddition.

[18] I. Foster, C. Kesselman, et al. The Physiology of the Grid:An Open Grid Services Architecture for Distributed Systems Integration. Technical report, Argonne National Laboratory, Chicago, January 2002.

[19] I. Foster, C. Kesselman, and S. Tuecke. The nexus approach to integrating multithreading and communication. *Journal Parallel and Distributed Computing*, 37:70–82, 1996.

[20] Ian Foster. Personal communication.

[21] Ian Foster. The Grid: A New Infrastructure for 21st Century Science. *Physics Today*, 55(22):42, 2002. Available from: http://www.aip.org/pt/vol-55/iss-2/p42.html.

[22] Ian Foster, Jeffrey Frey, Steve Graham, Steve Tuecke, Karl Czajkowski, Don Ferguson, Frank Leymann, Martin Nally, Igor Sedukhin, David Snelling, Tony Storey, William Vambenepe, and Sanjiva Weerawarana. Modeling Stateful Resources with Web Services. Technical report, IBM Developerworks, 5 March 2004. Version 1.1. Available from: http://www-128.ibm.com/developerworks/library/ws-resource/ws-modelingresources.pdf.

[23] Ian Foster, Carl Kesselman, Jeffrey M. Nick, and Steven Tuecke. *Grid Computing: Making the Global Infrastructure a Reality*, chapter The Physiology of the Grid, pages 217–249. Wiley, 2003. Available from: http://www.globus.org/research/papers/ogsa.pdf.

[24] Ian Foster, Carl Kesselman, and Steve Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of Supercomputing Applications*, 15(3), 2002. Available from: http://www.globus.org/research/papers/anatomy.pdf.

[25] Ian Foster, George Thiruvathukal, and Steve Tuecke. Technologies for Ubiquitous Supercomputing: A Java Interface to the Nexus Communication system. *Concurrency: Practice and Experience*, June 1997.

[26] The Global Grid Forum Web Page. Web Page. Available from: http://www.gridforum.org.

[27] Grid3: An Application Grid Laboratory for Science. Web Page. Available from: http://www.ivdgl.org/grid2003/.

[28] GriPhyN - Grid Physics Network. Web page. Available from: http://www.griphyn.org/index.php.

[29] Globus Ubiquitous Supercomputing Testbed Organization (GUSTO). Web page, 1998. Available from: http://www-fp.globus.org/research/testbeds.html.

[30] Information Power Grid Engeneering and Research Site. Web Page, 2001. Available from: http://www.ipg.nasa.gov/.

[31] The International Virtual Data Grid Laboratory. Web Page. Available from: http://www.ivdgl.org/.

[32] N. Karonis, B. Toonen, and I. Foster. MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface. *Journal of Parallel and Distributed Computing (JPDC)*, to appear, 2003. Available from: ftp://ftp.cs.niu.edu/pub/karonis/papers/JPDC_G2/JPDC_G2.pdf.

[33] Leonard Kleinrock. UCLA to Build The First Station in Nationwide Computer Network. Press Release, 1969. Available from: http://www.lk.cs.ucla.edu/LK/Bib/REPORT/press.html.

[34] The Legion Project. Web Page. Available from: http://legion.virginia.edu.

[35] Robert McMillan. Grid guru: An interview with argonne's steve tuecke. *IBM Developerworks*, 2003. Available from: http://www-106.ibm.com/developerworks/java/library/j-tuecke.html?dwzone=java.

[36] National Research Grid Initiative. Web Page. Available from: http://www.naregi.org/.

[37] NSF Middleware Initiative. Web Page. Available from: http://www.nsf-middleware.org/Middleware/.

[38] J. Novotny. The Grid Portal Development Kit, 2001. Available from: http://dast.nlanr.net/Projects/GridPortal/.

[39] Network Weather Service. Web page. Available from: http://nws.cs.ucsb.edu/.

[40] Open Grid Computing Environments. Web Page. Available from: http://www.ogce.org.

[41] Manish Parashar and Craig A. Lee. Scanning the Issue: Special Isssue on Grid-Computing. *Proceedings of the IEEE*, 93(3):479–484, March 2005. Available from: http://www.caip.rutgers.edu/TASSL/Papers/proc-ieee-intro-04.pdf.

[42] Manish Parashar and Craig A. Lee. Special Isssue on Grid-Computing. *Proceedings of the IEEE*, 93(3), March 2005. Available from: http://ieeexplore.ieee.org/xpl/tocresult.jsp?isNumber=30407&puNumber=5.

[43] Manish Parashar, Gregor von Laszewski, Snigdha Verma, Jarek Gawor, and Kate Keahey. A CORBA Commodity Grid Kit. *Concurrency and Computation: Practice and Experience*, 14:1057–1074, 2002. Available from: http://www.mcs.anl.gov/~gregor/papers/corbacog-ccpe-gce01-final.pdf.

[44] Particle Physics Data Grid. Web Page, 2001. Available from: http://www.ppdg.net/.

[45] pyGlobus Commodity Grid Kit. Web Page. Available from: http://dsd.lbl.gov/gtg/projects/pyGlobus/.

[46] David De Roure, Mark A. Baker, Nicholas R Jennings, and Nigel R. Schadbolt. *Grid Computing: Making the Global Infrastructure a Reality*, chapter The Evolution of the Grid, pages 65–100. Number ISBN:0-470-85319-0. John Wiley, 2003.

[47] Scientific Discovery through Advanced Computing (SciDAC). Web Page, 2001. Available from: http://scidac.org/.

[48] SETI@Home. Web Page, February 2002. Available from: http://setiathome.ssl.berekeley.edu/.

[49] L. Smarr and C.E. Catlett. Metacomputing. *Communications of the ACM*, 35(6):44–52, 1992.

[50] Larry Smarr. Multimedia online expo, "science for the millennium.". Online, 1995. NCSA. Available from: http://archive.ncsa.uiuc.edu/Cyberia/MetaComp/MetaHome.html.

[51] David F. Snelling. Web Services Resource Framework: Impacton OGSA and the Grid Computing Roadmap. 2(1):1, 2004.

[52] Storage Resource Broker (SRB). Web Page, 2001. Available from: http://www.npaci.edu/DICE/SRB/.

[53] P. Stelling, C. DeMatteis, I. Foster, C. Kesselman, C. Lee, and G. von Laszewski. A Fault Detection Service for Wide Area Distributed Computations. *Cluster Computing*, 2(2):117–128, 1999. Available from: http://www.globus.org/research/papers/stelling--hbm.pdf.

[54] TeraGrid. Web Page, 2001. Available from: http://www.teragrid.org/.

[55] Mary Thomas, Steve Mock, and Gregor von Laszewski. A Perl Commodity Grid Kit. *Concurrency and Computation: Practice and Experience*, 14:1085–1095, 2002. Available from: http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--perl-cog.pdf.

[56] John Towns. The Alliance Virtual Machine Room. Web Page, 2001. Available from: http://archive.ncsa.uiuc.edu/SCD/Alliance/VMR/.

[57] Steve Tuecke. Personal communication.

[58] Steven Tuecke, Karl Czajkowski, Ian Foster, Jeffrey Frey, Steve Graham, and Carl Kesselman. Grid Service Specification (Draft 2). Web page, June 2002. Available from: http://www.gridforum.org/ogsi-wg/drafts/GS_Spec_draft02_2002-06-13.pdf.

[59] Unicore. Web Page. Available from: http://www.unicore.de/.

[60] Gregor von Laszewski. An Interactive Parallel Programming Environment Applied in Atmospheric Science. In G.-R. Hoffman and N. Kreitz, editors, *Making Its Mark, Proceedings of the 6th Workshop on the Use of Parallel Processors in Meteorology*, pages 311–325, Reading, UK, 2-6 December 1996. European Centre for Medium Weather Forecast, World Scientific. Available from: http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--ecwmf-interactive.pdf.

[61] Gregor von Laszewski. A Loosely Coupled Metacomputer: Cooperating Job Submissions Across Multiple Supercomputing Sites. *Concurrency, Experience, and Practice*, 11(5):933–948, December 1999. The initial version of this paper was available in 1996. Available from: http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--CooperatingJobs.ps.

[62] Gregor von Laszewski. Grid Computing: Enabling a Vision for Collaborative Research. In Juha Fagerholm, Juha Haataja, Jari Järvinen, Mikko Lyly, Peter Raback, and Ville Savolainen, editors, *The Sixth International Conference on Applied Parallel Computing*, volume 2367 of *Lecture Notes in Computer Science*, pages 37–52, Espoo, Finland, 15-18 June 2002. Springer. *(Invited Talk)*. Available from: http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--para4.pdf.

[63] Gregor von Laszewski and Kaizar Amin. *Grid Middleware*, chapter Middleware for Commnications, pages 109–130. Wiley, 2004. Available from: http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--grid-middleware.pdf.

[64] Gregor von Laszewski, Steve Fitzgerald, Ian Foster, Carl Kesselman, Warren Smith, and Steve Tuecke. A Directory Service for Configuring High-Performance Distributed Computations. In *Proceedings of the 6th IEEE Symposium on High-Performance Distributed Computing*, pages 365–375, Portland, OR, 5-8 August 1997. Available from: http://www.mcs.anl.gov/~gregor/papers/fitzgerald--hpdc97.pdf.

[65] Gregor von Laszewski, Ian Foster, George K. Thiruvathukal, and Brian Toonen. A Computational Framework for Telemedicine. *Journal of Future Generation Computer Systems*, 14:10–123, 1998. Available from: http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--telemed.pdf.

[66] Gregor von Laszewski, Jarek Gawor, Sriram Krishnan, and Keith Jackson. *Grid Computing: Making the Global Infrastructure a Reality*, chapter Commodity Grid Kits - Middleware for Building Grid Computing Environments, pages 639–656. Communications Networking and Distributed Systems. Wiley, 2003. Available from: http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--grid2002book.pdf.

[67] Gregor von Laszewski, Jarek Gawor, Carlos J. Peña, and Ian Foster. InfoGram: A Peer-to-Peer Information and Job Submission Service. In *Proceedings of the 11th Symposium on High Performance Distributed Computing*, pages 333–342, Edinbrough, U.K., 24-26 July 2002. Available from: http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--infogram.ps.

[68] Gregor von Laszewski, Branko Ruscic, Kaizar Amin, Patrick Wagstrom, Sriram Krishnan, and Sandeep Nijsure. A Framework for Building Scientific Knowledge Grids Applied to Thermochemical Tables. *The International Journal of High Performance Computing Applications*, 17(4):431–447, Winter 2003. Available from: http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--knowledge-grid.pdf.

[69] Gregor von Laszewski, Mike Seablom, Milo Makivic, Peter Lyster, and Sanya Ranka. Design Issues for the Parallelization of an Optimal Interpolation Algorithm. In G.-R. Hoffman and N. Kreitz, editors, *Coming of Age,*

Proceedings of the 4th Workshop on the Use of Parallel Processing in Atmospheric Science, pages 290–302, Reading, UK, 21-25 November 1994. European Centre for Medium Weather Forecast, World Scientific. Available from: `http://www.mcs.anl.gov/~gregor/papers/vonLaszewski94-4dda-design.pdf`.

[70] Gregor von Laszewski, Mei-Hui Su, Joseph A. Insley, Ian Foster, John Bresnahan, Carl Kesselman, Marcus Thiebaux, Mark L. Rivers, Steve Wang, Brian Tieman, and Ian McNulty. Real-Time Analysis, Visualization, and Steering of Microtomography Experiments at Photon Sources. In *Ninth SIAM Conference on Parallel Processing for Scientific Computing*, San Antonio, TX, 22-24 March 1999. Available from: `http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--siamCmt99.pdf`.

[71] Gregor von Laszewski and Patrick Wagstrom. *Tools and Environments for Parallel and Distributed Computing*, chapter Gestalt of the Grid, pages 149–187. Parallel and Distributed Computing. Wiley, 2004. Available from: `http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--gestalt.pdf`.

[72] Web Services Resource Framework (WSRF). Web Page. Available from: `http://www.globus.org/wsrf`.

A final version of this paper is available as publication in the journal **Information Technology**, Oldenbourg Wissenschaftsverlag, München, Vol. 47, No. 6 (2005), pp. 319-329.