GES 2007 German e-Science

Available online at http://www.ges2007.de



# Grid Services Toolkit for Process Data Processing

T. Jejkal[1], T. Müller[2], R. Stotzka[1], M.Sutter[1], V. Hartmann[1] and H.Gemmeke[1]

[1] Institute for Data Processing and Electronics, Forschungszentrum Karlsruhe GmbH, Hermann-von-Helmholtz-Platz 1, 76344, Eggenstein-Leopoldshafen, Germany
[2] Institute for Nuclear and Energy Technologies, Forschungszentrum Karlsruhe GmbH, Hermann-von-Helmholtz-Platz 1, 76344, Eggenstein-Leopoldshafen, Germany
*email:* Thomas.Jejkal@ipe.fzk.de
*phone:* (+49 07247) 82 4042

**Abstract**

Grid is a rapidly growing new technology that will provide easy access to huge amounts of computer resources, both hardware and software. As these resources become available soon, more and more scientific users are interested in benefiting from them. At this time the main problem accessing Grid is that scientific users usually need to know a lot about Grid methods and technologies besides their own field of application. The Grid Services Toolkit (GST) is based on Web Services designed especially for the field of process data processing providing database access and management, common methods of statistical data analysis and project specific methods. The toolkit will fill to some extent the gap between high–level scientific Grid users and low–level functions in Grid environments, thus simplifying and accelerating the development of parallelized scientific Grid applications.

## 1 Introduction

Process data processing always had very high demands on computing resources. Nevertheless in the near future the demands of many process data processing projects will even grow further up to a point where data processing and data storage on local workstations will not be feasible any more. The best possibility to exhaustively fulfill the requirements of such projects in respect of standardization and stability is the use of oncoming Grid Computing technologies. Grid will finally provide access to today unimaginable amounts of computing resources. Therefore we naturally want to profit from these forthcoming resources by using Grid technologies in process data processing projects. At a first glance this sounds fairly simple. Ideally the Grid should be accessed from within any process data processing application by simply using an Application Programming Interface (API). However, in real life Grid Computing is still a scientific field of its very own. At this time, scientists willing to use the Grid are very often forced to learn job description languages for cluster computing or they are urged to implement their own services onto low–level interfaces additionally

to programming their own clients. Also they need a good deal of background knowledge about Grid Computing and the particular computing environment like eg. locally installed software packages and operating systems. This results in a large overhead and therefore in a low acceptance of Grid Computing in scientific communities, where it is not yet mandatory. At first of all knowledge about existing technologies realizing high–level Grid access is needed to avoid re–implementations or to find a general solution which can be taken as groundwork for the GST. There are some ambitions in this direction for example the Cactus Project [1]. Originally designed for parallel numerical relativity it became a more general platform as the code was rewritten for large scale computing. Finally a Grid–enabled version of Cactus was developed for the Globus Toolkit. Unfortunately Cactus provides only a few, very specialized mathematical methods and uses a proprietary interface definition. Therefore it does not qualify as basis for the GST but maybe can be integrated at a later date. The Grid application toolkit (GAT) as part of GridLab [2] is a modular plug–in prepared API. The main idea of GAT is to provide a consistent interface to access Grid Services of middlewares. As GAT abstracts from underlying middlewares it naturally does not provide all their features. Regardless the benefit of a consistent interface the advantages of implementing Grid Services with GT4 directly are even more valuable for us. The next chapter will deal with objectives which form the basis of GST. Following results are illustrated and finally a conclusion will be given.

## 2 Objectives of GST

The objective of the GST is as mentioned within the introduction providing a collection of high–performance Web Services which are feasible for Process Data Processing. The needed acceptance will be achieved by offering an intuitive API which hides anything concerning the underlaying Grid architecture Globus. The end–user should not have to know whether his process is running partly or completely on the Grid. There should not be any constraints by using the Grid. The GST API should be accessed like any other library to integrate GST seamlessly into new developments or even to replace parts of existing projects with a minimum of effort. Focussing existing implementations there should be also addressed the integration of legacy systems e.g. Matlab, R or Root. Special attention should lay on the domain of data access due to handling large amounts of data is a dominating topic in process data processing. Although such general services are useful most projects will have very special needs which can not be adapted to at least one other project. Doubtless there will be the need of specialized services which are completely fitted to single projects. To realize these objectives there should be used WSRF–compliant Web Services. The advantages of this technology are the easy–to–use high–level Grid access in a platform– and programming–language independent way. Furthermore WSRF represents an open standard and is therefore expected to be future–proof. Summarizing the final goal is to bundle high–level (scientific) Grid Services in a Grid Services Toolkit focussing especially on process data processing and high performance.

The next chapter will show the current achievements concerning the GST.

### 2.1 Results

As defined before GST consists of three major parts. The first one is responsible for solving statistical tasks and is called GST-Stat. This core of this part is a Web Service which is able to execute any kind of interpreter. The input and output of the respective interpreter can be on the one hand set or rather obtained by parameter and return type of a single Web Service operation. On the other hand the interpreter running within the Web Service can be controlled by secure streaming of data to and from the interpreter. Due to GST-Stat every project which is using an interpreter somehow or other can benefit from this part of the GST. The next topic which is addressed is the data handling done by GST-Dat. A realization of a general solution for this domain is a challenging task due to the big variation of used data management systems. For this purpose OGSA–DAI [3] was developed by the University of Edinburgh. OGSA–DAI offers a WSRF–compliant Web Service which abstracts a wide range of data resources by one uniform interface. Using this service the localization of the data resources does not play any longer a role. The focus in which the IPE wants to use OGSA-DAI lays on relational data management systems (RDBMS) like MySQL or Oracle. A drawback concerning RDBMS is that OGSA–DAI implements the access to such systems by raw SQL–queries but the SQL–syntax differs between different RDBMS. At this point GST-Dat will stand in due to it abstracts not only the localization of the resource by using OGSA-DAI but also the type of the RDBMS by wrapping specific SQL–types into general Java types. Furthermore this implementation revolutionizes database access at large even while using state–of–the–art access methods via JDBC. The last part of GST is called GST-CAD. CAD stands for Computer Aided Diagnosis and deals with project–specific tasks. Here are still missing most of the implementation but every oncoming development can and will use achievements of GST-Stat and GST-Dat as far as possible. Following the final chapter will give conclusions about the current results of GST.

## 3 Conclusions

Recapitulating it can be asserted that the current version of GST is very promising considering the needs of process data processing projects. Nevertheless there is remaining a lot of work which has to be done especially with the focus on rising the acceptance of Web Services and Grid technologies within the addressed scientific community. Finally the practical application will show if the needs of actual and oncoming projects can be continuously handled by the used technologies. If this is the case there is no doubt that there will be sooner or later the needed acceptance of the Web Services–based Grid and the advantages in comparison to accepted Grid or cluster environments.

## References

1. G. Allen, T. Dramlitsch, I. Foster, et al. "Supporting efficient execution in heterogeneous distributed computing environments with Cactus and Globus". In Proceedings of Supercomputing, 2001.
2. E. Seidel, G. Allen, AndrMerzky, and J. Nabrzyski. "GridLab - a grid application toolkit and testbed". Future Generation Computer Systems, 18:1143-1153, 2002.
3. M. Antonioletti, M. Atkinson, R. Baxter, et al. "The design and implementation of grid database services in ogsa–dai". Concurrency and Computation: Practice and Experience, pages 357–376, 2005.