GEMSS: Grid-infrastructure for Medical Service Provision

Authors

Siegfried Benkner, Guntram Berti, Gerhard Engelbrecht, Jochen Fingberg,
Greg Kohring, Stuart E. Middleton, Rainer Schmidt

Affiliations

Siegfried Benkner, Gerhard Engelbrecht, Rainer Schmidt:
Institute for Software Science, University of Vienna, Vienna, Austria

Guntram Berti, Jochen Fingberg, Greg Kohring:
C&C Research Laboratories, NEC Europe Ltd., Sankt Augustin, Germany

Stuart E. Middleton:
IT Innovation Centre, University of Ì ì · ¢ÓéÀÖÆÅÅÆÌ ì · ¢ÓéÀÖÆPP-ÙÍ ¢ÏÂÔØ, Ìì·¢ÓéÀÖÆåÅÆ_Ìì·¢ÓéÀÖ

Corresponding author

name: Siegfried Benkner
telephone: +43 1 427738801
fax: +43 1 42779388
email: siegfried.benkner@univie.ac.at
address: Institute for Software Science, Liechtensteinstrasse 22, A-1090 Vienna, AUSTRIA

Abstract

Objectives: The European GEMSS Project is concerned with the creation of medical Grid service prototypes and their evaluation in a secure service-oriented infrastructure for distributed on demand/supercomputing. Key aspects of the GEMSS Grid middleware include negotiable QoS support for time-critical service provision, flexible support for business models, and security at all levels in order to ensure privacy of patient data as well as compliance to EU law.

Methods: The GEMSS Grid infrastructure is based on a service-oriented architecture and is being built on top of existing standard Grid and Web technologies. The GEMSS infrastructure offers a generic Grid service provision framework that hides the complexity of transforming existing applications into Grid services. For the development of client-side applications or portals, a pluggable component framework has been developed, providing developers with full control over business processes, service discovery, QoS negotiation, and workflow, while keeping their underlying implementation hidden from view.

Results: A first version of the GEMSS Grid infrastructure is operational and has been used for the set-up of a Grid test-bed deploying six medical Grid service prototypes including maxillo-facial surgery simulation, neuro-surgery support, radio-surgery planning, inhaled drug-delivery simulation, cardio-vascular simulation and advanced image reconstruction.

Conclusions: The GEMSS Grid infrastructure is based on standard Web Services technology with an anticipated future transition path towards the OGSA standard proposed by the Global Grid Forum. GEMSS will demonstrate that the Grid can be used to provide medical practitioners and researchers with access to advanced simulation and image processing services for improved pre-operative planning and near real-time surgical support.

Keywords

Computer Simulation [MESH unique ID: D003198]
Computer Communication Networks [MESH unique ID: D003195]
Computer Security [MESH unique ID: D016494]

1. Introduction

The European GEMSS Project [6] is concerned with the creation of medical Grid service prototypes and their evaluation in a secure service-oriented infrastructure for distributed on demand/supercomputing. The medical prototype applications include maxillo-facial surgery simulation, neuro-surgery support, radio-surgery planning, inhaled drug-delivery simulation, cardio-vascular simulation and advanced image reconstruction. GEMSS will enable the wide-spread use of these computationally demanding tools originating from projects such as BloodSim[0], SimBio[7], COPHIT[2] and RAPT[5] as Grid services. At the core of these bio-medical simulation applications are computationally demanding methods such as parallel Finite Element Modelling, parallel Computational Fluid Dynamics and parallel Monte Carlo simulation.

In this paper we present an overview of the GEMSS infrastructure that is currently being developed. The GEMSS Grid infrastructure and middleware is being built on top of existing Grid and Web technologies, maintaining compliance with standards thereby ensuring future extensibility and interoperability. Furthermore, GEMSS aims to anticipate privacy, security and other legal concerns by examining and incorporating into its Grid services the latest laws and EU regulations related to providing medical services over the Internet. The remainder of this paper is organized as follows: Section 2 presents an overview of the GEMSS Grid Infrastructure and the main architectural components. Section 3 discusses the provision of applications as Grid services based on the concept of generic application services and the GEMSS QoS support infrastructure. Section 4 describes the GEMSS client framework followed by conclusions in Section 5.

2. The GEMSS Grid Architecture and Infrastructure

The GEMSS infrastructure is based on a service-oriented architecture comprising multiple Grid clients and Grid service providers, one or more service registries and a certificate authority. Grid Clients are usually Internet-enabled personal computer loaded with GEMSS client software that permits communication with a service provider through the GEMSS middleware. The client side applications handle the creation of service input data and visualization of service output data. Service providers expose numerical simulation applications running on HPC hardware as generic application services accessible over the network. All GEMSS services provide support for quality of service negotiation, data staging, job execution, job monitoring, and error recovery and are usually accessed subject to a chosen business model. Service registries maintain a list of service providers and the services they support. The certificate authority, which is usually managed by a third party, provides certificate authentication after appropriate identity checks.

The GEMSS design supports a three step process to job execution. First there is an initial business step, where accounts are opened and payment details fixed. The pricing model may also be chosen at this stage. Next there is a quality of service negotiation step, where a job's quality of service and price, if not subject to a fixed price model, is negotiated and agreed. Finally, once a contract is in place, the job itself can be submitted and executed.

The GEMSS infrastructure relies on Web Services technology that will allow integration, via OGSA[4] with various hosting platforms such as UNICORE[8] and GLOBUS[3].

GEMSS services are defined via WSDL and securely accessed using SOAP messages. For large file transfers SFTP is being investigated, as well as SOAP attachments. Since GEMSS supports a client driven approach for accessing services, it is not required that holes be tunnelled through site firewalls or any other special site security compromises.

2.1. GEMSS Grid Service Provision Infrastructure

Figure 1 (right hand side) depicts the main architectural components of the GEMSS Grid service provision infrastructure. Medical simulation applications are exposed as Web Services and hosted using a web server and a service container (usually Apache and Tomcat Axis).

Services are run under the direction and orchestration of the client, subject to the necessary availability of and authority to use resources. The quality of service management component handles reservation with the resource manager (job scheduler) and provides input to the quality of service negotiation process so that sensible bids can be made to client job requests. The error recovery component handles check pointing and re-starting of services if required. The logger manages a database for logging auditable information and a low level system log for event logging and performs most of the low level event logging for its intrusion detection module. The service state repository component manages a conversation state database that contains information about any client-service conversation allowing it to be resumed at a later time if the user logs off. The provision of applications is based on the concept of generic application services and described in more detail in Section 3.1.

2.2. GEMSS Grid Client Infrastructure

The main architectural components of the GEMSS Grid client infrastructure are shown in Figure 1 (left hand side). The client-side application code usually relies on the GEMSS client application programming interface (API) which hides most of the complexity of dealing with remote services from the application developer by providing appropriate service proxies. Service proxies are in charge of discovering services and negotiate with grid services to run jobs on its behalf. The session management component manages client sessions and maintains a security context, authenticating the current user and providing the access criteria for the certificate and key stores. A service discovery component is provided for looking up suitable services in a service registry. The workflow enactment component is in charge of running business and negotiation workflows, as well as service orchestration. The client typically runs a business workflow to open negotiations with a set of service providers for a particular application. The quality of service negotiation workflow is then run to request bids from all interested service providers who can run the clients jobs subject to QoS criteria required by the client; this results in a contract being agreed with a single service provider. The client then uploads the job input data to the service provider and starts the server side application by calling appropriate methods of the service. The client infrastructure is centred on a pluggable client side component framework which is described in more detail in Section 4.

2.3. Business Models for a Medical Grid

With an eye to possible exploitation, GEMSS will maintain a flexible business model to allow commercial operation of Grid services. There is a concept of clients having an account with each service provider, allowing payment details to be provided and monthly bills generated for Grid use. Business models considered for GEMSS include airline reservation/business models and telephone business models. A negotiation model is also supported so a client can shop around, negotiating with all the service providers who provide the required service to get the best deal. Within each negotiation the quality of service terms associated with the required job can be discussed, as well as the price involved. The aim for GEMSS is to provide a viable and flexible approach to operating a commercial Grid.

2.4. Privacy, Security and Legal Issues

Since GEMSS is concerned with the processing of highly confidential and private information, such as images of patient heads and commercially sensitive fluid dynamic models, there are serious privacy issues to consider. Within GEMSS EU law is being examined to identify privacy, contractual and ethical issues. To operate within EU law an appropriate level of security must be applied, with medical data anonymised where possible and not held for longer than is required to achieve the purpose of the Grid processing. Grid security must be periodically reviewed and patients made aware of the processing that will occur, and be able to review and correct the information held about them. It is expected that future Grid processing procedures would be integrated into a hospital's existing data processing procedures.

The GEMSS Grid infrastructure employs commercial of the shelf (COTS) technology, making full use of well-tested technology, regular security patches and best practice security procedures. Such security technologies will be evaluated with respect to the GEMSS Grid, in addition to the creation of a methodology for assessing the security needs at each GEMSS partner's site. The full GEMSS infrastructure is designed to provide a public key infrastructure, X.509 compliance, RSA encryption, service level authorization, logging and intrusion detection. The GEMSS Grid is designed to work with existing site firewalls, and does not require any insecure ports to be opened through them.

3. Providing Medical Simulation Applications as Grid Services

A major objective of the GEMSS projects is the development of a generic Grid service provision framework that simplifies the task of transforming existing applications into Grid services without having to deal with the details of Web Services and Grid technologies.

3.1. Generic Application Services

The transformation of medical simulation applications into Grid services is based on the concept of generic application services. A generic application service is a configurable software component which exposes a native application as a service to be accessed by multiple remote clients over the Internet. A generic application service provides common methods for data staging, remote job management, error recovery, and QoS support, which are to be supported by all GEMSS services. In order to customize the behaviour of these methods for a specific GEMSS application, an XML application descriptor has to be provided. Besides general information about a medical simulation service, the application descriptor specifies the input/output files, the script for initiating job execution, and a set of performance-relevant application parameters required for QoS support.

3.2. Service Generation and Deployment

A generic application services is realized as a Java component, which is transformed automatically into a Web Service with corresponding WSDL descriptions and customized for a specific GEMSS application using the XML application descriptor. In order to provide a native GEMSS application as a Web/Grid Service, the application has to be installed on some Grid host and a job-script to start the application as well as an XML application descriptor have to be provided. Finally, the generic application service has to be deployed within an appropriate hosting environment e.g. Apache Tomcat/Axis. As a result, the native application is embedded within a generic application service and accessible over the Internet. In order to provide support for automatic service deployment, a corresponding GEMSS deployment tool has been developed. In future, GEMSS services will be extended in order to be compliant with the Open Grid Services Architecture (OGSA).

3.2. QoS Support

GEMSS services support a model and process for agreeing, dynamically and on a case-by-case basis, various QoS properties, including service completion time, cost, availability and others. For this purpose, each GEMSS service provides methods for enabling clients to negotiate required QoS properties with a service provider before actually consuming a service. QoS enabled GEMSS services are capable of providing an estimated service completion time, based on meta data about a specific service request (e.g. image size, required accuracy, etc.) supplied by the client within a corresponding request descriptor.

The QoS support infrastructure for GEMSS is based on QoS contracts (XML), request descriptors (XML), and performance models, and relies on a resource scheduler that supports advanced reservation. For each GEMSS application, the service provider or application developer has to specify a set of application specific performance parameters in the XML application descriptor. For example, in the case of an image reconstruction service,

performance parameters typically include image size and required accuracy (i.e. number of iterations). In order to compute various QoS properties (e.g. service completion time) on the basis of the specified performance parameters, a machine-specific performance model has to be provided. Since in general, it will not be possible to build a simple analytical performance model for all GEMSS applications, we plan to build a data base relating typical problem parameters to resource needs like main memory, disk space and running time, which will initially be populated using data from test cases.

During QoS negotiation, the client has to supply a request descriptor containing concrete values for all performance-relevant parameters specified in the application descriptor. Moreover, the client has to pass to the service provider an initial QoS contract, specifying the required QoS properties and other conditions to use the service. On the service side, the request descriptor is fed into the performance model in order to determine whether or not the client's QoS requirements can be fulfilled.

4. Client Side Component Framework

On the client side, the GEMSS Project has chosen a flexible approach built around an SDK consisting of pluggable components. Using the SDK, developers can easily integrate Grid functionality into existing applications. The SDK is conservative, in that it does not expose any low level Grid concepts which applications should not need to deal with, while at the same time providing applications full control over higher level abstractions. Among the high level components identified to date are: Business Processes, Service Discovery, QoS Negotiation, and Workflow Enactment. The GEMSS SDK exposes the interfaces for these components to the applications, while keeping their underlying implementation in terms of Web Services hidden from view.

All the components making up the SDK are pluggable in the sense that they can be replaced by any implementation which supports the well defined public interfaces and properly replicates a component's documented behaviours. Support exists for simultaneously using multiple components from different providers, as well as the run-time exchange and update of the individual components. It is also planned to introduce an autonomic maintenance system in order to relieve users of the burden of downloading and installing updates to various components which are being developed by different institutions. Through the SDK, applications can use the system in one of two ways; either execute each component separately, thereby exercising full control over the process of service discovery, QoS negotiation and the service invocation, or, use a convenience component to automatically find the desired service, perform the QoS negotiation and create a service proxy. Either way, at the end of this process the application has a service proxy through which it can communicate directly with the desired service. Although most applications will use the SDK to interact mainly with GEMSS services, support is provided for interacting with any standard Web Service. Additional support for working with OGSA services will be incorporated later.

Finally, the SDK also includes methods for application dependent session handling. Applications can designate those service proxies or other objects which are to constitute a "session" and then have them serialized for later re-instantiation. This provides a lightweight incarnation of a persistent environment useful for working with the types of long running services commonly found in GEMSS.

5 Conclusions

The GEMSS Grid infrastructure is based on standard Web Services technology with an anticipated future transition path towards the OGSA standard proposed by the Global Grid Forum. Since the GEMSS project is concerned with the Grid-provision of advanced medical simulation applications, security at all levels, privacy and compliance to EU law have been integrated into the GEMSS infrastructure from the beginning. The GEMSS client-side and service-side infrastructures hide most of the Grid’s complexity from the application

developers and service providers. Since timely solution is critical when using simulation services in clinical practice, GEMSS services support a QoS model that lets the end-user specify the time of required job completion. Appropriate automatic negotiation processes will then choose a service provider that can fulfill a service request within the specified time constraints and subject to a chosen business model. Currently the GEMSS infrastructure is under active development. A first prototype Grid test-bed with all six GEMSS application services is already operational and currently being evaluated.

Acknowledgements

The GEMSS project is funded by the European Commission in the Information Society Technologies (IST) Programme under cross-program theme 'Grid Testbeds' - Framework V Project No. IST-2001-37153

References

1. The Bloodsim Project. EU Esprit Project 28350, 1998-2001. http://www.software.aeat.com/cfx/European_Projects/bloodsim/bloodsim.htm.
2. The Cophit Project. EU IST Project IST-1999-14004. http://www.software.aeat.com/cfx/European_Projects/cophit/index.html.
3. I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International J. Supercomputer Applications, 15(3), 2001.
4. I. Foster, C. Kesselman, J. Nick, S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.
5. A Gill, M Surridge, G Scielzo, R Felici, M Modesti, G Sardu, 1996, RAPT: A Parallel Radiotherapy Treatment Planning Code, In High Performance Computing and Networking Europe, Springer: Lecture Notes in Computer Science 1067, ed. H Liddell, A Colbrook, B Hertzberger, P Sloot, 183-193.
6. The GEMSS Project: Grid-enabled medical simulation services. EU IST Project IST-2001-37153, 2002--2005. http://www.gemss.de/
7. The Simbio Project. EU IST project IST-1999-10378, 2000-2003. http://www.simbio.de/.
8. The UNICORE Forum. http://www.unicore.org/

Figure Captions

Figure 1 caption:
GEMSS Client (left) and Service Provider Infrastructure (right hand side)

Figures

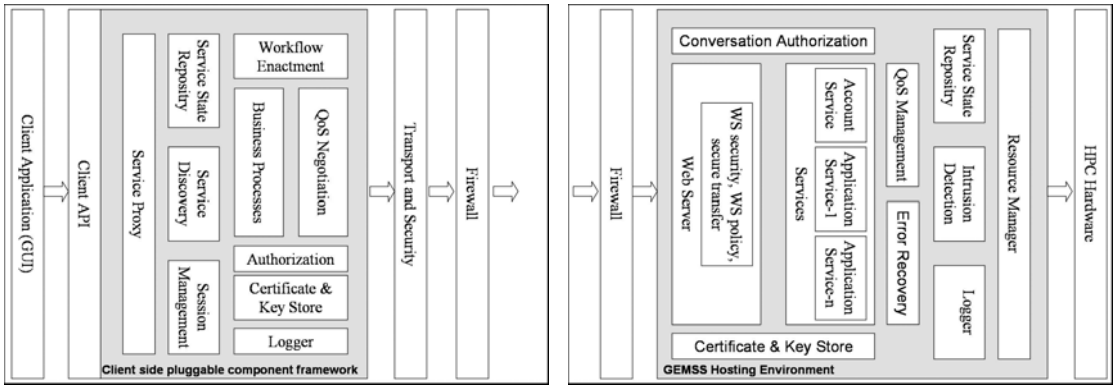

Figure. 1