Information Society

The IST Programme
Project No. IST-2001-37153

GEMSS

Grid-enabled Medical Simulation Services

http://www.gemss.de

Deliverable D6.2b
Second Annual Project Progress Report

Status:	Review Release
Version:	1.2
Security:	**Public**
Responsible:	NEC (Author: Jochen Fingberg)
Authoring Partners:	All Partners

Release History

Version	Date	
0.1	13.08.2004	Initial Template
0.2	23.08.2004	1st Draft
0.3	31.08.2004	2nd Draft
0.4	02.09.2004	3rd Draft
0.5	03.09.2004	4th Draft
0.6	08.09.2004	5th Draft
0.7	08.09.2004	6th Draft
1.0	08.09.2004	1st Release
1.1	10.09.2004	Review Release
1.2	28.09.2004	Final Release

The GEMSS Consortium:
NEC Europe Ltd. – UK
ISS – Austria
CRID – Belgium
ASD – Germany
IT-Innovation – UK
MPI of Cognitive Neuroscience – Germany
IBMTP – Austria
CFX Ltd. – UK
IDAC – Ireland
Sheffield University – UK

Second Annual Project Progress Report

Executive Summary:

This annual report covers project months 13-24 (September 1st, 2003 - August 31st, 2004) of the GEMSS project and provides a summary of technical progress made and a consolidated management overview of the present period. Activities reported are in line with the project workplan as covered by the GEMSS Contract IST-2001-37153. The Second Annual Project Progress Report will be available as a public document.

Table of Content:

1 Project Overview

GEMSS is developing an interoperable, innovative Grid middleware for medical service applications building on common Grid standards. The focus is on innovative extensions that support medical applications including security models compliant with European legal issues, fail-over and recovery from errors as well as workflow and service orchestration techniques for time-critical processes. GEMSS is a two and a half year project which started in September 2002.

1.1 Objectives

The central objectives of the GEMSS project are to:

- demonstrate that the Grid can improve pre-operative planning and near real-time surgical support by providing access to advanced simulation and image-processing services,
- build middleware on existing/developing Grid technology standards to provide support for authorisation, workflow, security and Quality of Service aspects,
- develop, evaluate and validate a test-bed for the GEMSS system, including its deployment in the end-user's working environment,
- analyse and test the European Legal Framework from three approaches (Privacy, Contract, Liability) considering the use of Grid-based-technologies / applications partially provided through the Internet, in order to know if the European Legal Framework permits the development and the exploitation of such applications and if positive, under which constraints.

1.2 Main goals

The main goal of GEMSS is to provide end-users from the medical community with advanced tools at their workplace through easy-to-use interfaces. In particular GEMSS will:

- install a secure, extendible, interoperable and collaborative test bed for GRID-enabled medical application services,
- demonstrate the medical significance of the GEMSS models,
- demonstrate the functionality of the GRID-infrastructure,
- open new business models for future commercial exploitation,
- highlight and show the main legal aspects to be considered when developing and implementing GRID-based technologies.

1.3 Priorities

Through the central objective outlined in section 1.1 GEMSS is an infrastructure project driven by its 6 prototype medical service applications.

1.4 Architecture

The GEMSS architecture will use a client/server topology employing a service-oriented architecture. Since GEMSS support a public key infrastructure there can be a number of trusted certificate authorities to issue security certificates. Service registries hold details of where services can be found and are queried before a client interacts with specific service providers. Figures 1a and 1b show the client and server architecture. Deliverable D1.2b has more details.

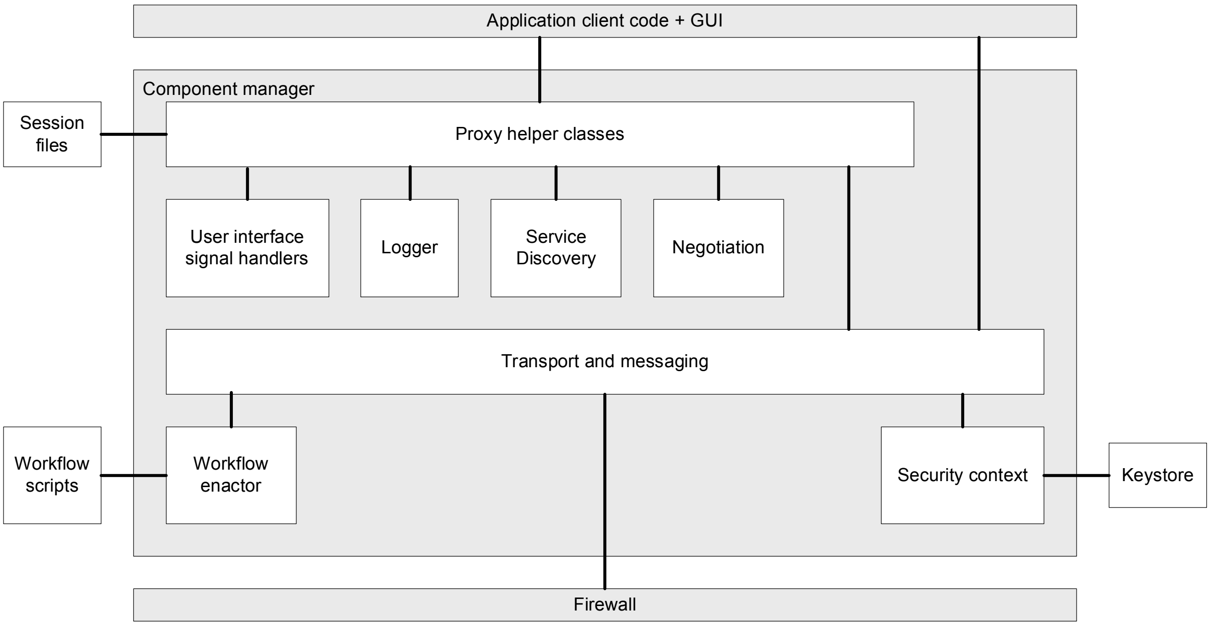


Figure 1a: GEMSS client architecture

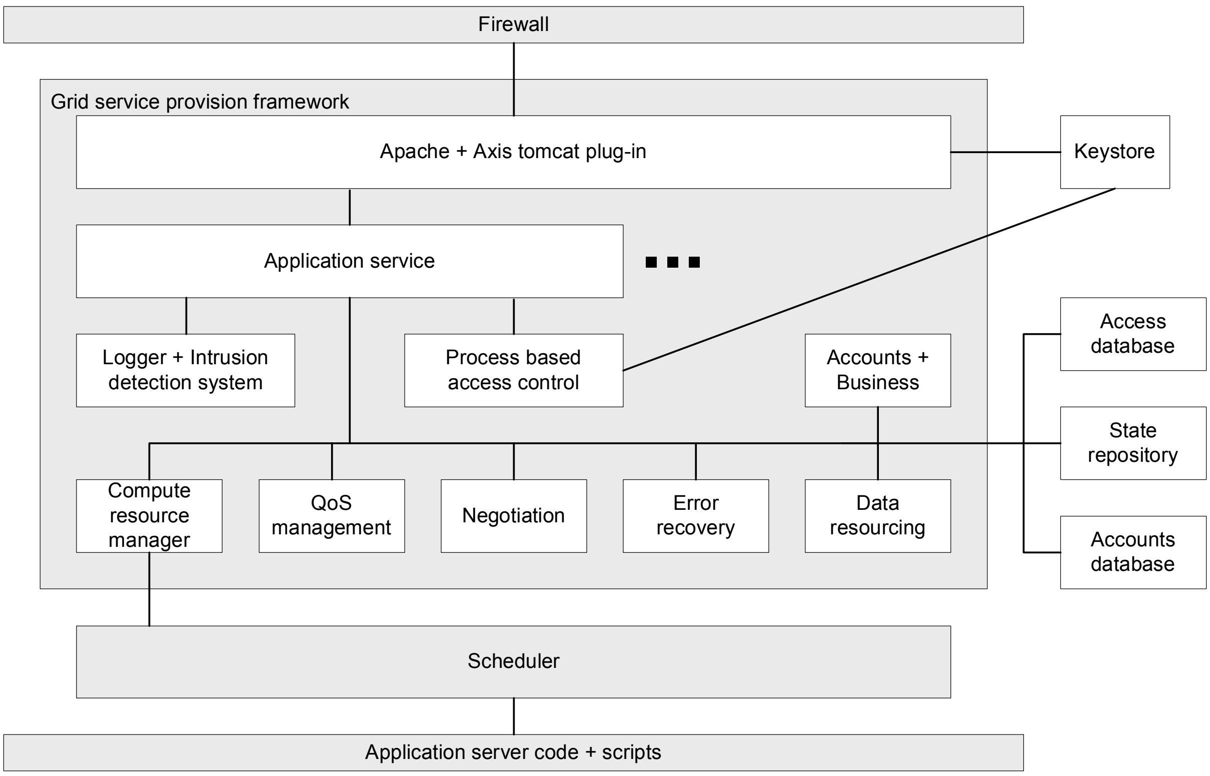


Figure 1b: GEMSS server architecture

2 Consortium

The Consortium consists of 10 partners, with three experienced in the GRID, two software developers, two University medical departments, and three end users, two of whom are specialist SME's in the area of bio-medical simulation. The Consortium thus represents a well balanced mixture of private and public partners whose activities range from basic research to industrial/commercial development and sales. The heterogeneous character of the Consortium has been designed to form a critical mass to best address the scientific requirements imposed by the problems to be solved.

No	Partner	Country	Specific Expertise, Role in the project
1	NEC	UK	Finite element simulation, mesh generation, GRID-middleware, HPC software & hardware, Project Management.
2	MPI	Germany	Medical image processing, Model validation, Acquisition of scan-data, medical end-users.
3	ISS	Austria	Programming environments, software tools, HPC tools and hardware, GRID technology.
4	USFD	UK	FSI modelling, cardiovascular and respiratory applications. Provision of validation datasets, scan data. Clinical test site for radiosurgery application. Medical end user.
5	AEA (1)	UK	Software and applications.
6	IT-Innovation	UK	GRID technology and know-how, technology transfer.
7	CRID	Belgium	Consultant in legal issues
8	ASD	Germany	FE and CFD consultancy for artificial organs and biomedical devices.
9	IDAC	Ireland	Stress-analysis consultancy, Consultancy, ANSYS re-sellers, GUIs
10	IBMTP	Austria	Image reconstruction software, acquisition of scan data, clinical end users
11	CFX (2)	UK	Software and applications.

(1) until 28.02.2003.
(2) from 01.03.2003.

The following table shows the involvement of partners in work packages. Work package and sub-task leaders are marked in red.

Partner	ST1.1	ST1.2	ST1.3	ST2.1	ST2.2	ST3.1	ST3.2	ST3.3	ST4.1	ST4.2	ST4.3	ST4.4	ST4.5	ST4.6	WP5	WP6
NEC	●	●	●	●	●	●	●	●	●						●	●
MPI	●		●						●	●					●	
ISS	●	●	●	●	●	●	●	●						●	●	
USFD	●		●								●	●	●		●	
CFX/AEA	●		●	●		●	●					●	●		●	
IT-Innov.	●	●	●	●	●		●				●				●	
CRID	●		●		●										●	
ASD	●		●									●	●		●	
IDAC	●		●	●		●	●	●					●		●	
IBMTP	●		●											●	●	

3 Progress Report

3.1 Workpackage 1: System Design and Evaluation

This workpackage has three main responsibilities:

- to capture the system and end-user requirements,
- to produce a global design for GEMSS including the medical service applications and
- to evaluate the testbeds and the medical service applications within the testbeds produced during the course of this project.

3.1.1 Sub-task 1.1: Requirements Capture

The requirements capture process was performed in the first three months of the project and provided a sound basis from which to start the global system design. All partners were asked to list their requirements and these were refined into five categories. The five types of requirements are listed in table 3.1.

Type	**Comments**
Business requirements	Commercial requirements of the GEMSS business model both for the supplier and end-user of Grid computing power.
Legal and social requirements	Legal, and to a lesser extend social, requirements expected to arise from the availability of end-user medical services.
Security requirements	Security requirements of both medical end-users and suppliers.
Performance requirements	Performance requirements of end-user and suppliers.
Application requirements	End-user requirements for the medical services they wish to support.

Table 3.1. Requirement types

Some of the most important requirements were found in the security category, since within GEMSS we are dealing with medical applications and must work with highly sensitive patient data. The security requirements were born in mind throughout the design of the GEMSS infrastructure, since from the legal analysis it was clear that we could not make patient data any less sensitive and had to ensure best practice security was applied throughout.

Deliverable D1.1 contains the full set of user requirements, and some application use cases which formed the basis of the initial system design analysis.

3.1.2 Sub-task 1.2: System Design

The GEMSS architecture will use a client/server topology employing a service-oriented architecture. Since GEMSS support a public key infrastructure there can be a number of trusted certificate authorities to issue security certificates. Service registries hold details of where services can be found and are queried before a client interacts with specific service providers. Figure 3.1 shows this basic architecture.

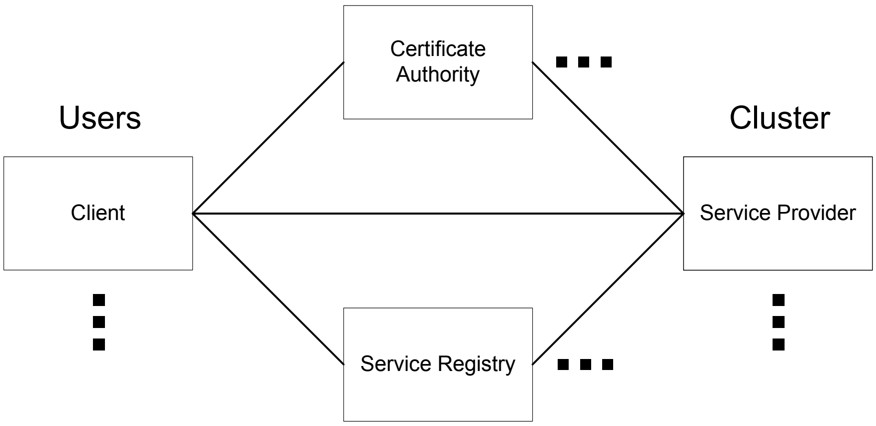


Figure 3.1. High-level client/server architecture

Client Users who have a job to run.

Service provider People who support the Grid servers that can run Grid jobs

Certificate authority Third party who provides certificate authentication after appropriate identity checks.

Service registry Registry holding a list of service providers and the services they support. Registries may be global or provided locally by a client.

Throughout the design that follows we have applied a three step process to job execution. First there is the 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. As discussed later, there is a hierarchical nature to these steps, with one or more quality of service negotiations expected per account, and one or more jobs per quality of service contract. Figure 3.2 shows this three-step process.

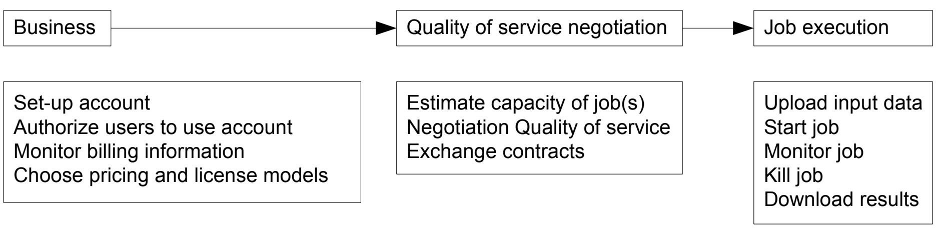


Figure 3.2. Three step process to using the Grid

3.1.2.1 Client architecture

The client architecture is shown in figure 3.3. The client would typically run an offline business workflow to open an account with a service provider and agree payment. When a Grid job is required to be run the client would then open negotiations with a set of service providers for a particular application job. The quality of service negotiation would then be run to request bids from all interested service providers who can run the clients jobs; this would result in a contract being agreed with a single service provider. The client would then upload the job input data, and any application workflow, to the service provider where the server side application would take over.

The component manager provides a way to dynamically load different versions of components as required, allowing flexibility of implementation. All components have a well-defined interface that the component manager knows.

Most workflow within GEMSS is encoded by the proxy helper classes and supports negotiation and job handling workflow. There will be, however, a workflow enactor component which can execute soft coded workflow scripts. This is intended to allow service providers to export workflows for their services that the client can run, thus allowing future services to describe how they should be used.

The client always drives the three-steps of the GEMSS process. The reason for this is the requirement to operate with firewalls and not tunnel holes through them. Given this, there can be no service provider initiated connections, or call-backs, to the client.

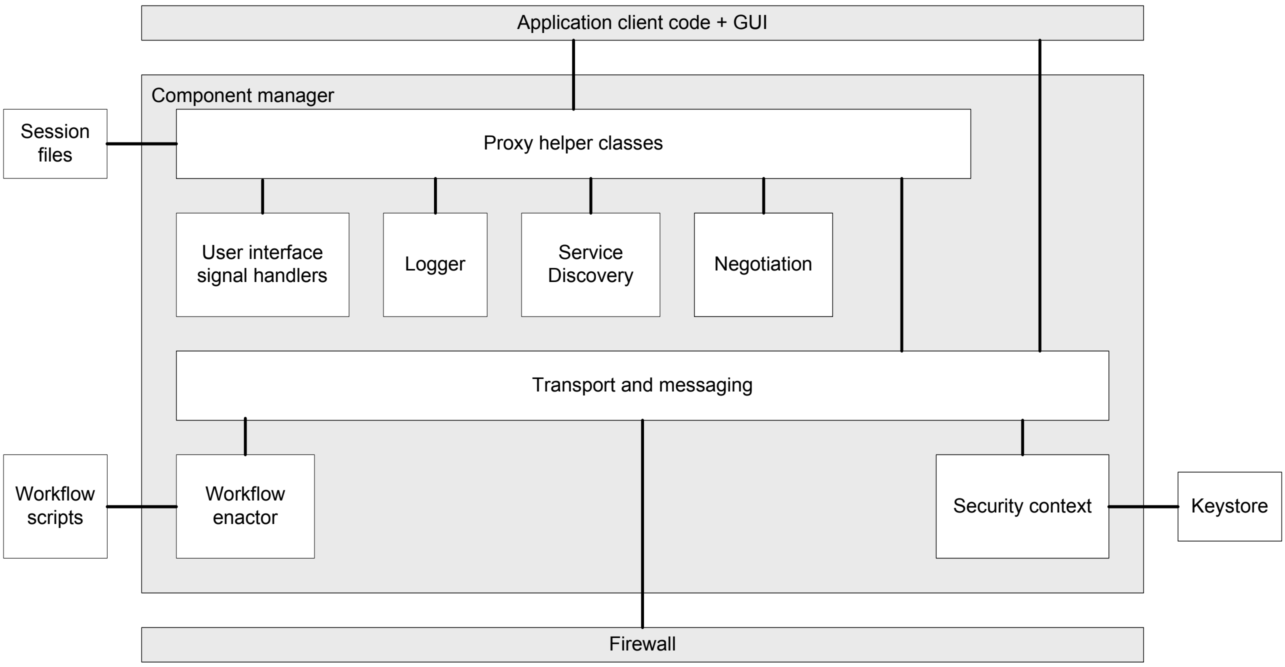


Figure 3.3. Client architecture

Application client code and graphical user interface

This is the application code, which will be linked to the component manager. The application code will provide a custom application user interface and manage the proxy helper classes. It is expected that the major part of the client application code will be the user interface and job workflow orchestration.

Component manager

The component manager is a plug-in framework capable of supporting a set of components, each with well-defined interfaces. It provides a signalling mechanism that components can use. Components in the component manager's code repository can be dynamically loaded as required by the application. The component manager will also manage client sessions and provides a start-up and shutdown function the application can call when it is first run.

The component manager supports a mechanism to regularly check for patches to existing components and completely new components. This removes some of the burden on the users checking for upgrades, and could make fast security patching a practical feature of this system.

Logger

This is a low level logging component that is used by the service provider side intrusion detection system.

Negotiation

This component provides the functionality to run a client driven auction with several service providers. The result of this auction is an agreement to run a specific job. FIPA protocols are used to allow future agent support to the negotiation process.

Proxy helper classes

Proxy helper classes provide a simple interface that the applications can link to. These proxy classes are intended to be the main contact point for the applications, invoked via the component manager. There are proxy helper classes to run Grid jobs, agree quality of service details and negotiate with several service providers to get the best deal for a job. These proxies are persistent and support state, saved to session files, which allows jobs to be run over a protracted period without the need for a persistent client connection to the service provider.

Security context

This component supports maintenance of a security context. It provides a framework for generating, storing and verifying trust for security tokens of various types. It provides the security context used by the transport component for HTTPS and secure message encoding.

Service discovery

This component invokes the service registry and obtains a list of services that match a query. The registry this component searches can be specified by the client allowing only trusted service registries to be queried by the client.

Transport and messaging

This component is capable of creating low level messages, and sending them with appropriate transport bindings, as specified by the WSDL that defined the remote service. All calls to services are fed through the transport and messaging component. The appropriate WS security and WS policy procedures will be implemented before a message is sent, and before a response is passed to the other client-side modules.

User interface signal handlers

These signal handlers are a set of default handlers that applications can use. These signal handlers are only default handlers, and it is expected that a sophisticated application would create their own handlers, which would link directly to their own custom user interfaces.

Workflow enactment

This is a component capable of parsing and running a workflow language script. This component is expected to be used as a technology demonstrator for soft coded quality of service negotiation workflows.

3.1.2.2 Service provider architecture

The service provider architecture is shown in figure 3.4. Applications will be run by the service under the direction and orchestration of the client, subject to the necessary availability of and authority to use resources. The compute resources available at the service provider's site will be used to actually run the services. The error recovery module is there to checkpoint services and re-starts them if required. The quality of service management module is there to handle reservations with the compute resource job scheduler and provide input to the quality of service negotiation process so that sensible bids can be made to client job requests.

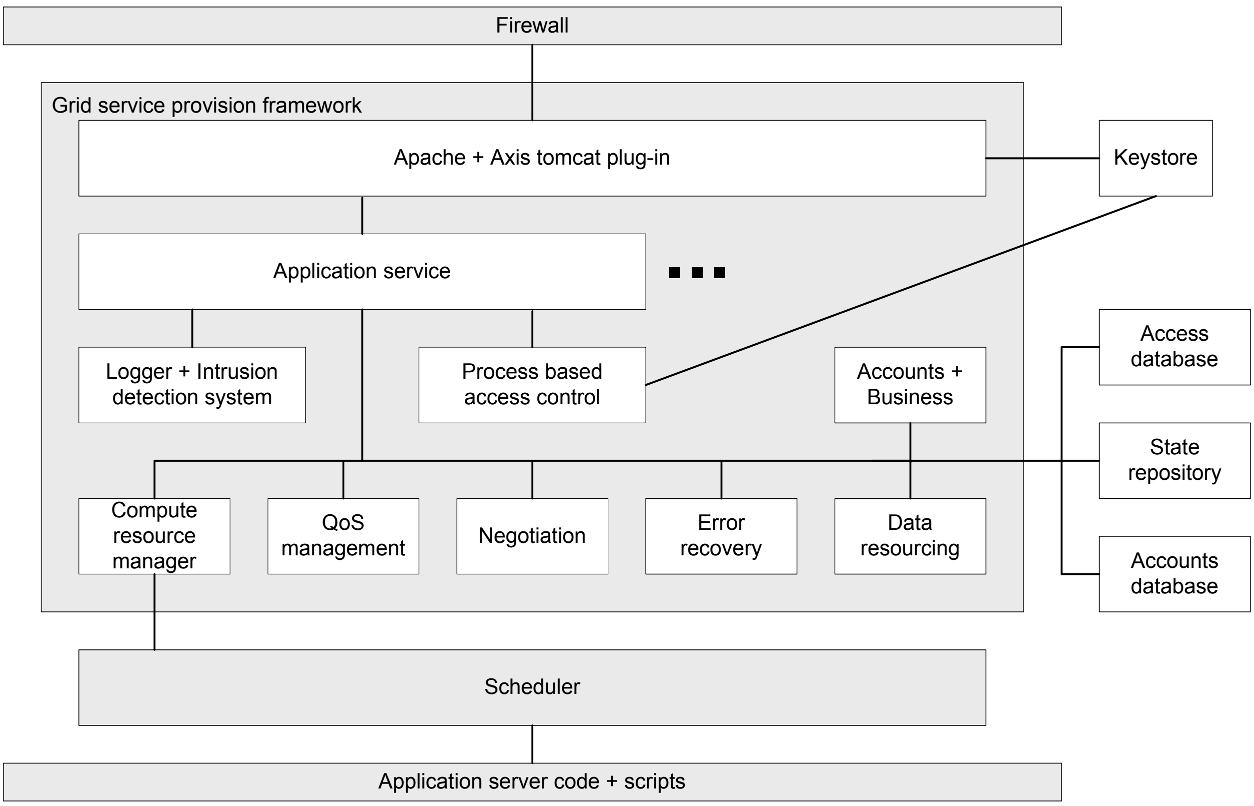


Figure 3.4. Service provider architecture

Accounts and business

This service maintains a database of auditable information about client accounts, including payment details and the current billing status. When jobs are run the business modules pricing models are used to calculate the payment due for resources used. A web interface is provided to allow client budget holders to access their billing status and review their account.

Application service(s)

The application service provides all the operations defined within the service WSDL file. These operations are routed to the appropriate module by various handlers, which are optionally installed at the service provider's site. The process based access control module is always consulted before an operation request is sent to a module.

Application server code and scripts

Each application must install its server side code and a set of scripts to run these codes. These scripts all have the same interface, and basically expect to receive an archived file containing the input data and produce a archived file with the output data. Start scripts invoke the scheduler as required by the service provider installation.

Apache + Axis tomcat plug-in

Tomcat plug-in's are used to support WS security enhancements. Tomcat itself provides the web service interface to individual application services whose functions are exposed via a WSDL interface.

Compute resource manager

This module is written to interface with the service provider's choice of scheduler. This module will be called to make reservations for jobs with the scheduler and start jobs. Since each service provider can have a different scheduler, along with different compute resources, there will have to be custom compute resource manager modules for each service provider.

Data resourcing

This module allows for persistent data between jobs, allowing intermediate files to be used as the input to other jobs etc. This is intended as a technology demonstrator within GEMSS.

Error recovery module

Closely linked to the compute resource manager, this module may use check-pointing to re-start or re-locate jobs as required (where the resource manager or the application itself enables this). This is intended as a technology demonstrator within GEMSS.

Grid service provision framework

This is a server side hosting environment that allows services to be installed and deployed.

Logger + Intrusion detection system

A module to securely log and analyse low level events in a distributed environment with the aim of detecting intrusion. The intrusion detection module may also integrate with conventional non-grid intrusion detection subsystems associated with the web server. This is intended as a technology demonstrator within GEMSS.

Negotiation

This module provides server side support for a closed-bid reverse English auction. It supports FIPA protocols and calls the QoS management module to try to get the best reservation possible. Bids are returned to the client for interpretation by the client side negotiation component encoded as web service level agreements (WSLA).

Process based access control

This module maintains an access control database of allowed process actions. The application service and other server side modules will check with the process based access control module before any operations are invoked. This provides a mechanism to check the clients distinguished name, obtained from the WS security header information, and confirm the operation was expected and allowed.

QoS management

This module handles the QoS interactions with the scheduler, via the compute resource manager. The QoS management module takes a web service level agreement (WSLA) and tried to make a reservation with the scheduler to provide the required levels of quality of service. Initially temporary reservations are made until the client confirms the reservation and agrees to run the job.

The QoS management module required an application specific performance module from each of the applications that are installed for GEMSS. This performance module is used to estimate the exec time jobs will take given a metadata description of the jobs input data.

Scheduler

GEMSS assumes the availability of a service provider specific scheduler that supports reservation. We use NEC’s COSY scheduler and the MAUI scheduler (LGPL license) within GEMSS.

3.1.2.3 Development progress

The current status of development work to date is shown in figures 3.5 and 3.6. The blue modules have been developed, the green modules are a work in progress and the red modules are technology demonstrators. Technology demonstrator modules will not be released as part of the main software release, but will be worked on until the end of the project and form the basis of a proof of concept development.

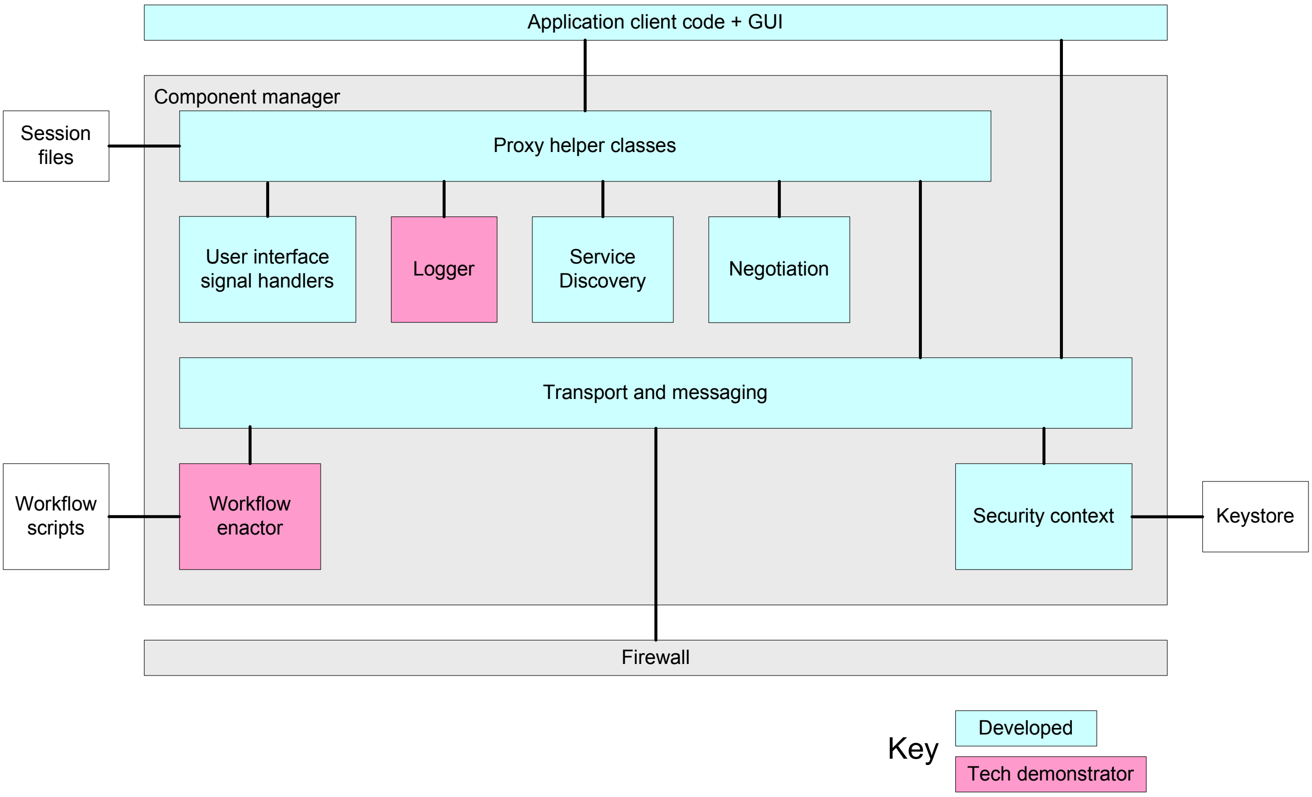


Figure 3.5. Client module development progress

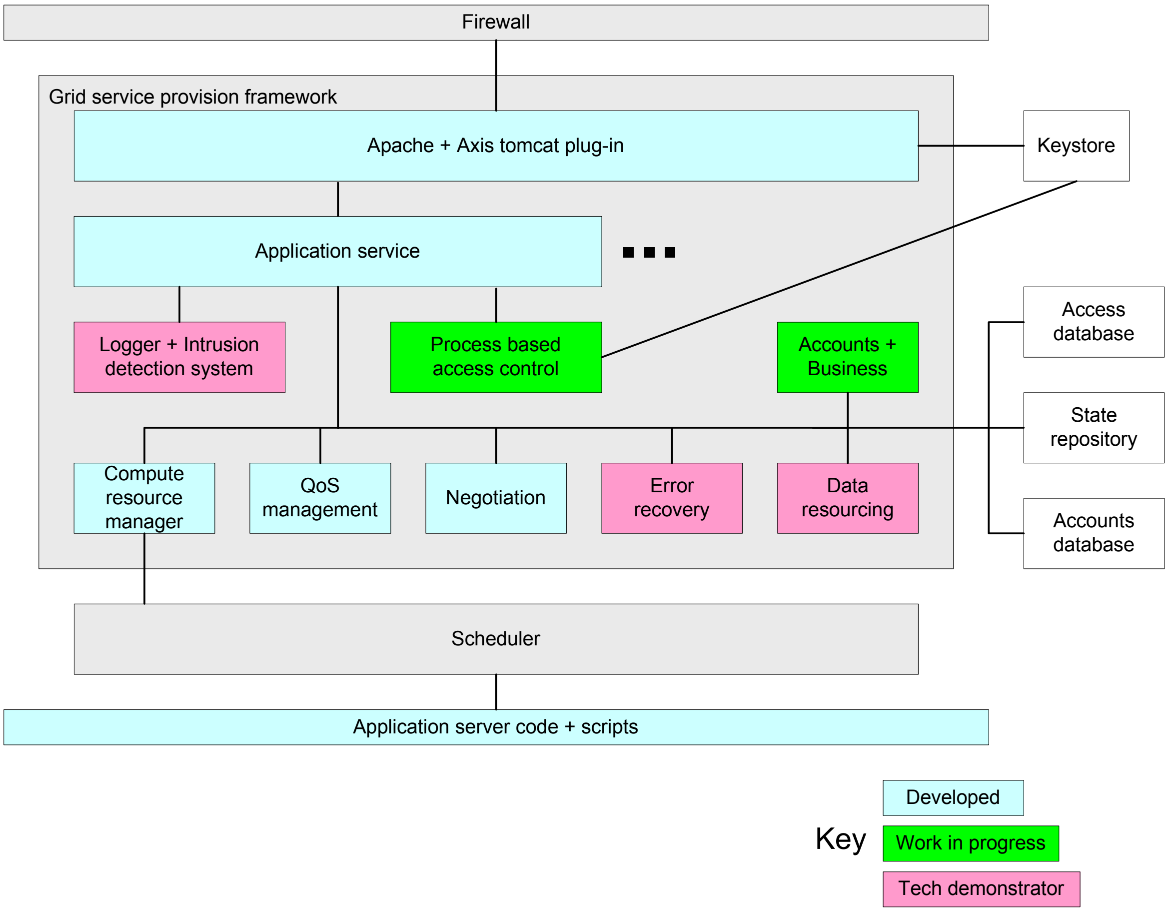


Figure 3.6. Service provider module development progress

Low-level details of each of these modules can be found in the next section, and example sequence diagrams for common Grid activities found in the section after that.

3.1.3 Sub-task 1.3: Evaluation

At project midpoint, an evaluation of the software development progress was performed, and use case diagrams as well as system goals were re-evaluated. User and developer experience was collected in form of a questionnaire. All six medical applications are Grid-enabled by now, using the initial version of the GEMSS middle ware. First performance and robustness tests validated the initial middle ware design and demonstrated the quality of the implementation.

Four tests were performed as part of the mid-term evaluation: The infrastructure test assessed the file transfer delays, showing how the size of the data set and the hardware use effect performance. The application testing characterized each application in terms of how it reacted to more CPU's, revealing some of the differences in nature between each of the six GEMSS applications. A 19 hour robustness test was conducted with all six applications, showing the GEMSS infrastructure to have excellent robustness when running jobs over the unreliable Internet. Finally, two of our applications performance models were assessed, providing an early insight into this work which is actually planned for after the mid-term release.

Important feedback was given by the application developers to optimise and level out a few problems regarding the middleware:

- The certification process and Java keystore setup should be optimised.
- The interaction of the scheduler and the GEMSS server system is an issue.
- Some of the design goals were re-prioritised during the Namur meeting in February, 2004, in order to better match requests and remaining development effort.

The next major steps in this sub-task prepare the evaluation of the final release of the middle-ware as well as the applications. This final evaluation will include:

- benchmarking file transfer rates, negotiation overhead, availability of the middleware,
- testing the application performance estimation models,
- assessing application developers and end users experience with the middle ware, and comparing the features of the final software release with the requirement specifications.

3.2 Workpackage 2: Grid Services & Security

3.2.1 Sub-task 2.1: Workflow and Quality of Service

3.2.1.1 Quality of Service

Design

The GEMSS QoS design comprises the service registry and discovery subsystem, the QoS negotiation subsystem, the QoS management subsystem, and the error recovery subsystem. The QoS negotiation subsystem, which handles the negotiation process, comprises a client component (client negotiator) and a service module (service negotiator). The QoS management subsystem includes the management and monitoring of QoS requests and offers by interacting with the compute resource manager to ensure advance reservation of resources (e.g. CPU time). Advance reservations are based on estimates provided by the application performance models. Furthermore, the QoS management monitors the progress of submitted jobs via the compute resource manager. The error recovery module handles check-pointing and re-starting or re-allocation of jobs. Check-pointing/restarting must be supported by the scheduling system and/or the native application.

Service registry

The GEMSS Service registry has been designed as standalone web service that provides operations for service (un-)registration and advanced service query. Each service provider maintains a service registry that contains all installed services hosted by the service provider. The service registry contains all information about a registered service available at the time of installation. The information about services stored in the registry comprises common attributes (service category, service endpoint URI, etc.) and specific attributes which are arbitrary name-value pairs. The query mechanism also supports queries with both common and specific attributes.

As standalone web service the GEMSS Registry Service is component-independent, i.e. it has no dependence on the other GEMSS components/modules and it can be used within standard web services hosting environment (e.g. Apache Tomcat).

A GEMSS registry V1.0 following the specification has been released. It contains a standalone web service that may be installed within the GEMSS hosting environment. Additionally a command line tool is provided in order to support registration and unregistration of services. This tool is also used within the GEMSS service provision framework in order to publish installed services in the GEMSS registry service.

Service discovery

The GEMSS Service discovery component has been designed and released as a client component fitting in the GEMSS client pluggable component framework. The discovery component provides advanced service discovery by querying remote GEMSS registry services. A client application or any other GEMSS client component may use the discovery component in order to discover appropriate services within a given set of registry services.

The discovery component provides an abstract API allowing potentially multiple implementations of the service discovery component to be provided. The connections between service and registry are secured using web service and transport security.

QoS XML-message exchange
The message exchange in the GEMSS QoS system is based on XML documents. A client describes a request by providing an XML Request Descriptor which contains meta-information about the request (e.g. mesh size, number of iterations, etc). For the QoS requests and offers processed by the QoS management, initially the following documents are used:

- QoS Descriptor: Describing agreed QoS properties between service provider and consumer
- Request Descriptor: Describing application specific meta information about the client request
- Performance Descriptor: Describing the output of the application performance models like estimated runtime, memory requirements, etc.

The WSLA (Web Service Level Agreement) specification was evaluated and incorporated in order to perform QoS negotiations in a standard format. Thus, QoS contracts and descriptors follow the WSLA specification.

Quality of service negotiation
The GEMSS quality of service negotiation involves a single client negotiating with one or more service providers, each of which is competing to win the client's business. As negotiation strategy a "closed-bid reverse English auction" protocol was implemented. This auction protocol has been examined in the Agent research area and is encoded within the FIPA agent communication standards.
The GEMSS negotiation code automates the service discovery steps and auction protocol. The underlying mechanism for both discovery and negotiation is thus hidden from the user via a simple interface. This interface can be used via the GEMSS pluggable client component framework.

Basic QoS Negotiation Component
Besides relying on automated QoS negotiation (reverse English auction), arbitrary negotiation strategies may be implemented manually using directly the methods provided by the basic QoS negotiation component, which provides a client-side API for the QoS negotiation.

Performance Estimation
A crucial aspect of the GEMSS QoS infrastructure is performance estimation. GEMSS relies on application specific performance models which are to be provided for each GEMSS application. Each performance model may be implemented differently depending on the nature of the application. First results for the accuracy of performance models are promising (e.g. SPECT performance model has 95% accuracy for runtime estimations).

Compute Resource Manager
The GEMSS QoS Infrastructure requires a scheduling system that supports advance reservation. Therefore the COSY scheduler developed by NEC has been successfully integrated via an abstract interface and is used to handle advance reservation of HPC resources.

Error Recovery
The detailed design and implementation of the error recovery module is part of phase 5 of the GEMSS implementation plan. An interface design is available featuring up- and downloads of check-pointing data as well as restarts of applications.

3.2.1.2 Workflow

Design:

There are three levels of workflow in GEMSS, job handling, quality of service negotiation and business workflow. These features have been introduced as hard-coded workflows, and towards the end of the project the negotiation workflow will be examined for soft-coding as a technology demonstrator.

Soft-coding workflow is useful since it allows different service providers to perform negotiation, business processes, job handling etc differently without the client having to know. The client would simply download a service provider's workflow for a specific service, or set of services, and enact it. Potentially this will make the problem of adding new services and service providers to the Grid much more scalable, and reduce the integration time involved in using services.

Initial hard-coded workflow:

The initial workflow has been hardcoded in the phase I3 prototype, with job handling and negotiation workflow encoded into proxy components. Full details of the proxy components are available in the design deliverable D1.2b.

The GEMSSProxy provides an easy to use Java interface representing the GEMSS application service. As a proxy it provides java method wrappers for each of the application services functions, hiding the transport component invocation calls that are required behind the scenes. The GEMSSProxy component thus provides everything the application developers need to run jobs.

The GEMSSNegotiator provides all the functionality of the GEMSSProxy, but with additional support for negotiation workflow. This class will automatically conduct an auction with a number of service providers and agree the best WSLA contract with one of them.

Shipped with the phase I3 prototype is an example application that demonstrates how to use the proxy classes and run Grid jobs. The workflow to run GEMSS jobs is very simple, consisting of the series of steps shown in figure 3.7. The negotiation workflow is more complicated, but from the interface screen shot of the example application's GUI it can be seen "in action" represented at a high level.

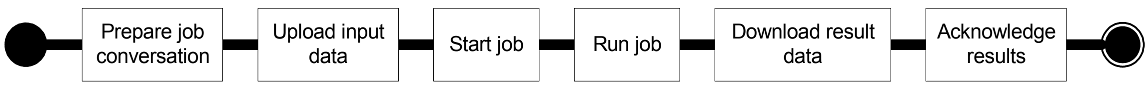


Figure 3.7. GEMSS job handling workflow

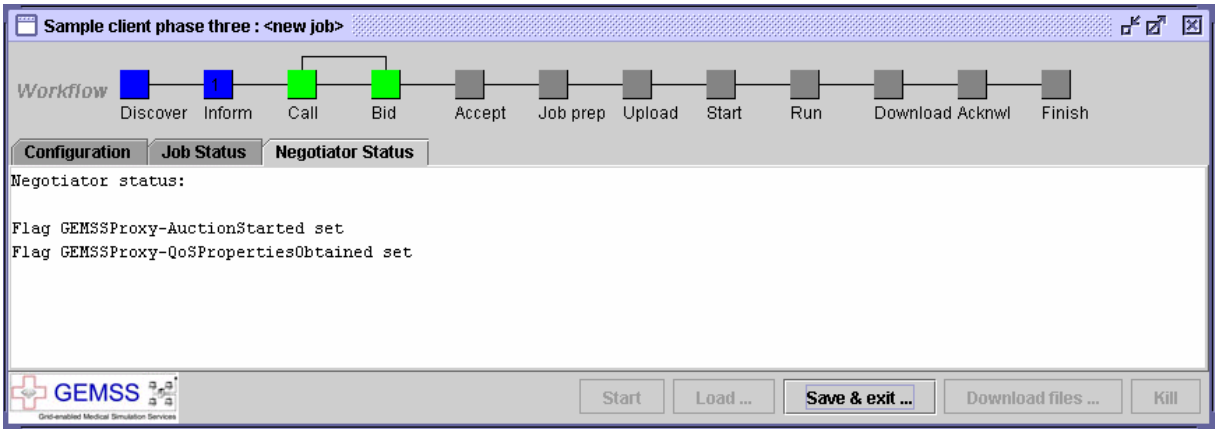


Figure 3.8 Screen shot of the example application shipped with the phase I3 prototype

The GEMSSNegotiator is a helper class that has, as an attribute, a GEMSSProxy instance. The purpose of the GEMSSNegotiator is to hardcode the quality of service negotiation workflow, allowing all negotiation steps to be performed automatically. The application using a GEMSSNegotiator component need not worry about either service discovery or negotiation with different service providers for the best deal.

Evaluation of the FreeFluo workflow enactor and Scufl workflow language:

The Scufl language: This section contains a short summary, based on the work being performed in the myGrid project [3]. The Scufl language is a high-level conceptual workflow language and full details of Scufl can be found on the Taverna Open Source project site http://taverna.sourceforge.net. Commercial workflow enactors and languages exist, such as IBM's BPEL4WS, but within GEMSS we have restricted ourselves to the promising workflow enactor called FreeFluo, which has been released as LGPL. FreeFluo enacts the Scufl workflow language, and a Scufl definition consists of three main entities.

Processors: A processor can be regarded as a function of some set of input data to a set of output data, where each function may have side effects on the execution environment that are not encapsulated within the input / output specification. Processors have a set of named input ports, a set of named output ports, a name within the scufl space, and a current execution status (initializing, waiting, running, or completed).

Data links: A data link represents the consumption of some processor output by an input of some other processor. In fact, there is nothing in the language to prevent a processor consuming one of its own outputs, although this may be rejected during the translation to some other format due to the implicit problems with cyclic workflows. Data links have a source processor and output port name, a sink processor and an input port name and an optional name within Scufl space.

Concurrency constraint: Although the data link specifications are enough to ensure correct execution ordering, since processors are allowed to have side effects on their environment it is often required to explicitly create constraints on the ordering of execution of different processors. It is possible to create a gate constraint that must be satisfied before a processor can effect a particular state change; for example, processor one is only allowed to shift state from waiting to running when processor two has status 'completed'. Constraints have a processor controlled by the constraint, a state change blocked in that processor, a gate condition, and an optional name within scufl space. Concurrency constraints are particularly useful in dealing with stateful interaction with services as shown below.

The FreeFluo workflow enactment engine: Freefluo is a Java workflow orchestration tool for web services that currently supports a subset of WSFL as well as Scufl. Freefluo is very flexible and at its core is a reusable orchestration framework that is not tied to any workflow language or execution architecture. The enactor core supports an object model of a workflow in the form of a directed graph where each node has a state machine that defines its lifecycle. Workflow scheduling and state transitions are driven by message passing between nodes as execution of the workflow progresses. The core of the enactor is decoupled from both the textual form of a workflow specification and the details of service invocation and the data model. This allows the core to orchestrate a workflow in a generic way.

The enactor core is used in the context of a particular language and service run-time environment. A workflow language parser is used to convert a textual workflow specification, e.g. a Scufl document, into the internal object representation of the enactor core. An invocation framework is then added to allow the enactor to actually invoke services in the run-time environment and deal with the specific data types passed between the services invoked, e.g. WSDL calls and XML message parts.

Freefluo can easily be extended to support different invocation methods (Web Services, Grid Services, CORBA) and has been used in other projects in this way, for example for using CORBA wrapped numerical methods and data sets in a steel modelling workbench currently in use by the European Coal and Steel Community. Details of the Freefluo Open Source project site can be found at http://freefluo.sourceforge.net

It is the ability to extend the enactor's run-time that also enables easy integration of stateful services, since the 'mini-workflow' of using these services can be encapsulated in bespoke extension. Furthermore, the run-time extensions are a natural and simple place to provide features such as iteration over datasets and automatic type casting or conversion.

Freefluo also supports generation of provenance information (what, when and where for all activities performed in a workflow) and also provides service discovery via standard UDDI if a service is not bound in a workflow specification (soon to be supported in Scufl).

An example Scufl workflow, created using the Taverna workflow-authoring tool, is shown in figure 3.9. The Scufl specification describes how to orchestrate the set of Web Services that provide the required functionality. The triangles at the top of Fig 3.9 are workflow inputs, the triangles at the bottom are workflow outputs and the green ovals are Web Service operations. The solid lines represent the data flows with the text annotations showing the data types.

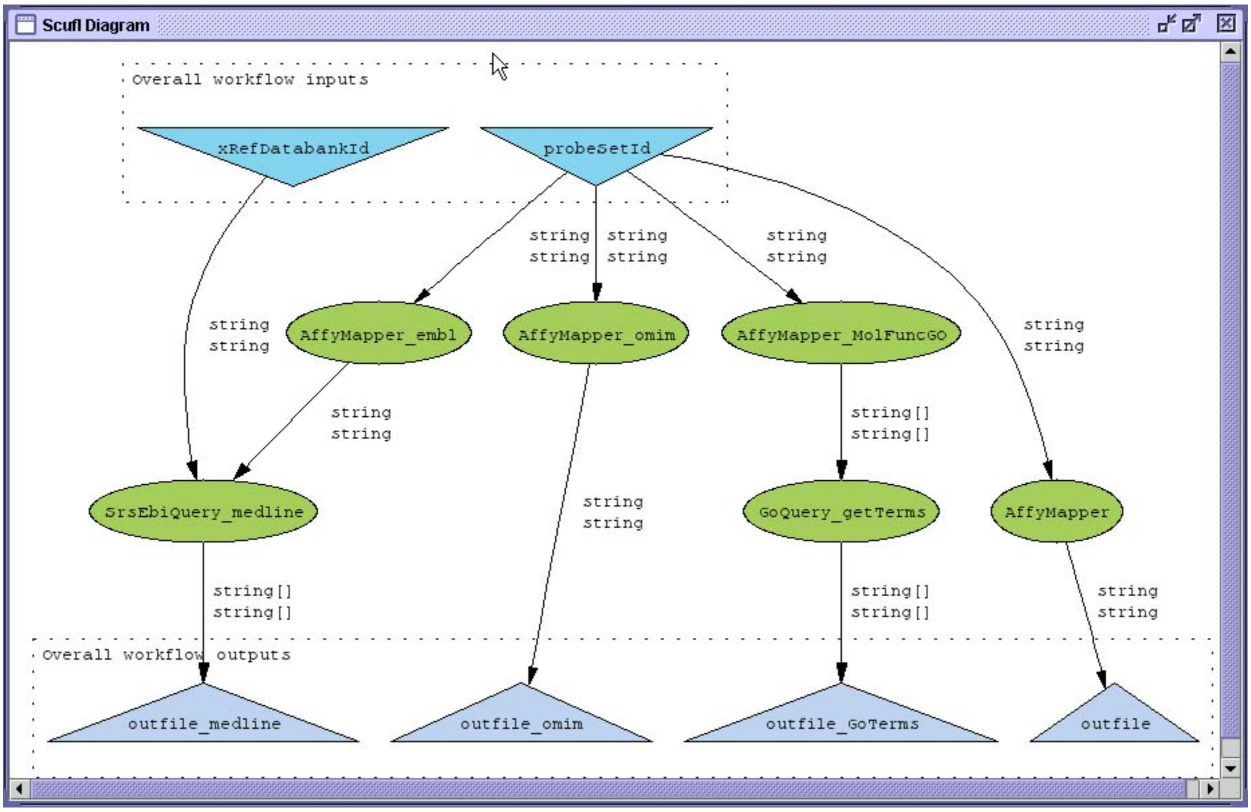


Figure 3.9 Taverna workflow-authoring tool

Scufl can also reference multiple types of workflow, such as in figure 3.10. Here some parts of the workflow require the use of two applications. Services are colour coded according to type, and the important thing to note is that these all appear at the same level of abstraction to the user despite the different levels of complexity of invoking the Web Services involved. Also shown in the diagram is a series of Web Service invocations that are cascaded together using control links. This set of invocations corresponds to the use of a stateful Web Service where a series of calls needs to be made in order to execute the application and retrieve the results. In this case, the series of invocations is explicitly visible to the user, as they are not abstracted through an extension to the workflow enactor. It should be clear from this example how the workbench and enactor could be used to invoke other stateful applications, e.g. using CORBA or Grid Services.

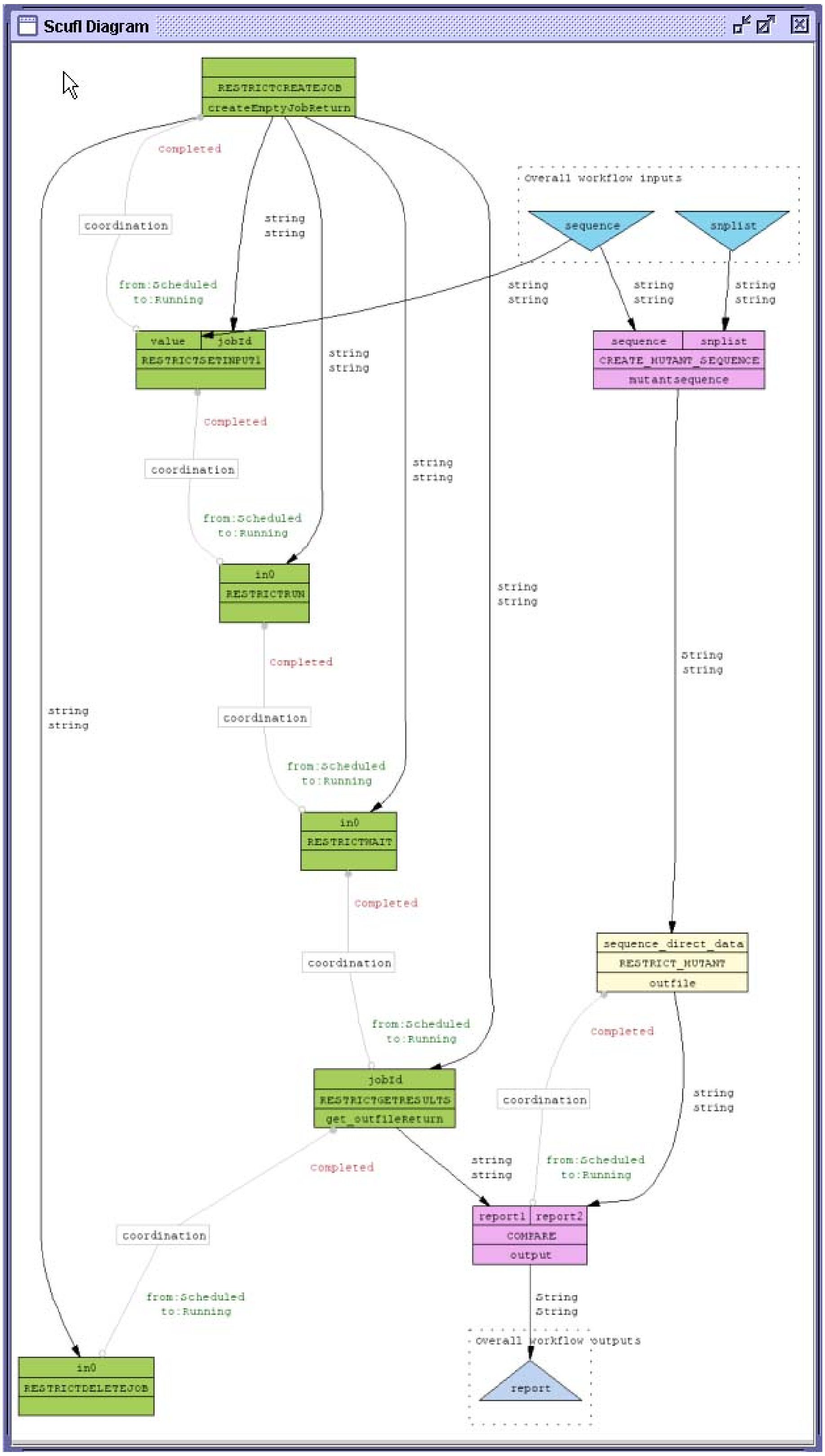


Figure 3.10 Incorporation of different types of service into a workflow

We feel that FreeFluo offers excellent opportunities for GEMSS, and could easily be wrapped up as a client side GEMSS component. Since the FreeFluo code has been released as LGPL there are no licensing issues, and since IT Innovation created FreeFluo as part of the myGrid project they already have in-house expertise in using this tool.

Next Stepps:

Workflow enactor component: In order to support workflow enactment the FreeFluo workflow enactor tool can be wrapped up as a GEMSS component. As a component it will provide a Java interface to the rest of the client side framework and allow execution of soft-coded workflow. This work will be performed as a technology demonstrator, shipped after the phase I4 software release as a proof of concept.

Soft-coding the negotiation workflow: Once a workflow enactor component is ready a negotiation workflow can be written. An appropriate workflow language would be selected, probably Scufl, and the workflow authored using a tool such as Taverna. Work can then commence on testing that the soft-coded negotiation workflow, enacted by the workflow enactor component, behaves in the same way as the proven hard-coded GEMSSNegotiator does. This work will be for proof of concept initially, allowing us to gain valuable insights into the complexity of soft-coding workflows and its practical benefits.

3.2.2 Sub-task 2.2: Security and Legal Issues

3.2.2.1 SECURITY ISSUES

3.2.2.1.1 Security assessment methodology

The first task performed was to create a security methodology for GEMSS, which users of the GEMSS infrastructure could apply to assess their own security. Security is like a chain, and a chain is only as strong as its weakest link. As such the security assessment methodology aims to enable users to examine their own security weaknesses, and thus enable them to address issues and improve the security of the overall GEMSS Grid. Deliverable D2.2a contains this methodology.

An asset-based analysis was proposed, which should be used to assess cost-benefits and effectiveness of GEMSS grid security features. To take account of the special properties of grid systems, the analysis should encompass dependencies on external assets, and others dependencies on one's own assets.

The main threats identified for a health Grid are:

- *Skilled crackers:* Skilled people who attack systems out of a sense of challenge. They typically do not have huge resources but do have plenty of time and the ability to find and exploit new and previously unexpected weaknesses.

- *Script kiddies:* Medium skill crackers that use the tools created by skilled crackers to perform exploits. As such they can only attack old weaknesses that will normally have patches available. However, not all system managers keep security patches up-to-date so these attacks are frequently successful.

- *Criminals:* Crackers who are motivated by financial reward for their exploits. They will use hacking tools to exploit weaknesses that will return financial reward.

- *Malicious insiders:* Disgruntled employees who have an axe to grind. They typically will have a high level of access and be very familiar with the operation of the Grid.

- *Incompetent insiders:* Users who unintentionally compromise security by failing to protect security-critical assets (e.g. passwords and private keys) or by foolishly importing malicious software or data.

- *Zealous commercial interests:* We expect that health-related grids will eventually involve not just clinicians and medical researchers, but also interested companies such as health insurers and some large employers. These companies can gain significant business benefits from access to grid-based data about (say) patients or competitors. Over-zealous attempts to access this data, potentially including clandestine industrial espionage, should be considered.

Given these likely threats, the most likely attack modes are:

- Social engineering attacks including impersonation, repeat fault attacks, etc.

- Brute-force cryptographic attacks against password files and private keys, if they can be obtained.

- Authentication attacks, including falsification of identity (by malicious insiders), and man-in-the-middle attacks.

- Authorisation escalation attempts including exploitation of bugs in services and very frequent scripted attacks.

- Network attacks including denial of service and distributed denial of service, and traffic spoofing possibly in combination with domain name service attacks.

- Malicious software including trojan horses and viruses (that may be accidentally imported by insiders, or deliberately planted), worms and root kits.

Because GEMSS is working with a health Grid, where patient data is being used for computation, the damages expected from security breaches are substantial. Litigation in the region of millions of pounds could well be expected if patient data is negligently handled.

It was suggested that the following defence options be seriously investigated for use in GEMSS, since they appear both cost effective and do not impose a large performance penalty. An analysis will of course be needed for each partner site to make an accurate appraisal.

- Encryption methods and secure communication protocols should be widely employed to authenticate communications and data and protect them from unauthorised access (this is standard practice on any grid).

- Access controls should be implemented as a matter of course.

- Software security should be implemented where necessary – e.g. to contain compromised processes or reduce the risk of importing trojans or viruses.

- Network security should be employed – firewalls are now more-or-less standard, but other techniques may be needed to counter some threats. A critical aspect of grid network security is to ensure it is consistent with existing measures against conventional threats.

- Security in depth should be maintained including standard practices such as logging. Measures such as intrusion detection and security audits should also be considered. The design of GEMSS should take account of the need for security updates, etc.

- The legal and policy issues should also be addressed. In the limit, one must have some sanction if one manages to catch an intruder, even if they are a legitimate grid user acting in accordance with the law in (over-zealous) pursuit of legitimate interests.

The outcome of the analysis of assets, threats and defences should be an estimate of:

i) the annual financial cost of establishing and maintaining security defences (should be affordable);

ii) the expected annual cost of attacks thereby prevented (should be greater than the cost of defences);

iii) the expected annual risk and cost of attacks not prevented (should be acceptably small).

What is an affordable defence or an acceptable risk should be determined by comparison with the cost of defences and the level of risk present without using the grid. The incremental cost and risk of grid participation should be small.

The analysis of GEMSS security must be carried out by each participating organisation. The overall infrastructure will be the same for all, but the costs and benefits of some local measures may be different for each participant. The goal is to enable each GEMSS partners to use the grid without excessive cost, while maintaining acceptable security for other users.

3.2.2.1.2 Security review

The security review has been published in deliverable D2.2c and did the following:

- Summarised the requirements for any GEMSS security infrastructure system
- Introduced an assessment methodology for evaluating technologies and toolkits
- Introduced and assessed possible enabling technologies and standards for providing the functionality defined by the requirements
- Established what constitutes best practice in terms of authentication, authorisation, accounting, data privacy
- Reviewed possible Grid solutions
- Evaluated off-the-shelf implementations of best practice technologies and standards.

The GEMSS security model is composed of a mixture of traditional security measures and new service-orientated security. Toolkit and technology choices for each part of the security model are based on best practice techniques.

The proposed security model has been implemented over the course of the project. The implementation plan reflects the dependencies between parts of the security model and dependencies on components within the global system design. Components that are implemented within GEMSS are not described in detail because they are described in 'GEMSS Global System Design' however these components will be developed using the best practice standards and toolkits established by this report.

Security model implementation includes the following parts:

1. GEMSS PKI – Establish GEMSS public key infrastructure including CA and RA representatives for each consortium partner.
2. Security Context Module – Develop security module that enables access to basic PKI authentication mechanisms deployed at Grid sites.
3. Transport and Messaging Module – Develop module capable of making HTTPS connections to services and encoding WS-Security integrity information.
4. Enhanced Security Module – Develop a module that extends and coordinates both the Security Context Module and parts of the Transport and Messaging Module in order to provide for end-to-end mutual authentication, key-exchange and message

confidentiality in situations where a Grid site that is composed of more than one security domain.

5. Apache-based Grid server – Provide mechanisms to host GEMSS services behind Apache SSL server.

6. Logger Module – Develop module for systematic logging of GEMSS events and data processing. The logging module will collect and log events at various points in the GEMSS system. The logging module forms the basis of the intrusion detection system.

7. Intrusion Detection Module– Develop module for detecting malicious use of GEMSS software.

8. Security Response Infrastructure – Develop basic response process for all sites including security advisory service.

9. Conversational Authorisation Module – Integrate GRIA process-based authorisation into GEMSS server side infrastructure.

3.2.2.2 LEGAL ISSUES

3.2.2.2.1 Privacy approach

- The European Privacy Legal Framework has been analysed considering the development and the exploitation of the six specific applications of the GEMSS project. The main result is that these applications may be developed and exploited under certain conditions.

Using GEMSS applications implies processing of personal data concerning the patient's health. The health practitioner will collect and dispatch the patient's personal data (e.g. imaging) through the Internet to the GEMSS provider.

The GEMSS provider will process (alone or with the support from other subcontractors) the patient's personal data in order to complete the therapeutic purpose of the data processing or the scientific research, before sending back the processed data or making them available to the health practitioner, again through the Internet.

The issue of data processing where using the GEMSS applications is only one aspect of the processing of patient's personal data concerning health by his/her health practitioner for therapeutic purposes and for scientific research.

- Many rules issued from different authorities concern the protection of privacy where processing patient's personal data and medical data by his/her health practitioner for therapeutic purpose and scientific research, such as :

- Article 8 of the European Convention on Human Rights and the judgements of the European Court of Human Rights,
- Article 7 & 8 of the Charter of Fundamental Rights of the European Union, the Convention of the Council of Europe for the protection of individuals with regard to automatic processing of personal data,
- Recommendation (97) 5 of the Council of Europe on the protection of medical data,
- Recommendation (83) 10 of the Council of Europe on the protection of personal data used for scientific research and statistics,
- Directive 95/46 on the protection of individuals with regard to the processing of personal data and on the free movement of such data,
- Directive 2002/58 concerning the processing of personal data and the protection of privacy in the electronic communications sector,
- Opinion n° 13 of the European Group on Ethics,
- Declarations, Considerations and Guidelines from the World Medical Association on Patient's Rights, Telemedicine, Health Databases, and Medical research involving Human subjects.

- It was demonstrated that the service provider is a processor (sub-contractor) of the data processing's controller. The controller and the processor must be legally or contractually bound (it implies that only well-identified partners operate the applications). The medical data should be processed by a health professional subject to professional secrecy, *or* by another person subject to an equivalent obligation of secrecy, *or* by individuals or bodies working *on behalf* of a health care professional (like a processor) and also subject to such secrecy obligations. These results are relevant and may be extended to all similar GRID-based applications in the health care sector.

The obligation of having a legal or contractual binding link between the controller and some of the subcontractors of the processor might be difficult to manage except if the GEMSS applications are operated by well-identified partners (e.g. use of a trustmark identifying the sub-contractors). There is also the obligation to process medical data by a health professional subject to professional secrecy or by another person subject to an equivalent obligation of secrecy or by individuals or bodies working on behalf of health care professionals also subject to such secrecy obligations.

Two reservations have to be mentioned. *First*, the European privacy rules allow differences between national privacy policies of the Member States. But these differences cannot prevent the transfer of personal data between Member States if the Member States have transposed the European Privacy Directives into their national laws. *Secondly*, as explicitly mentioned in their title, European privacy rules focus only on the protection of privacy. Consequently, if privacy reasons others than the one's indicated in the European Privacy rules cannot prevent personal data processing or personal data transfers between Member States, other reasons such as Deontological reasons, Public Health reasons, Social Security Reasons, etc. can still be in conflict with these operations. With regard to these potential obstacles, there are no practical legal remedies today.

Two remedies can be suggested with regard to this issue : *First*, identify the potential obstacles in each Member States and try and propose European solutions. *Secondly*, establish codes of conduct explaining the use of GEMSS applications, persons and bodies involved with their functions and duties, and clearly describing the patient's rights. On short term, the second remedy should be encouraged rather than the first one. The first remedy should be considered only in the long term.

3.2.2.2.2 Contractual approach

- Considering the provision of services throughout Europe from a contractual point of view, two questions had been addressed :

1° What is the content of the contractual obligations of the contracting parties?
2° Is it possible to conclude a contract by electronic means?

3.2.2.2.2.1 The content of the contractual obligations of the contracting parties

The content of the contractual obligation of contracting parties depends on the applicable law.

The determination of the applicable law to the contractual obligations of the contracting parties, for the provision of GEMSS services, depends on several rules of conflict of laws.

Sometimes these rules of conflict of laws have different natures:

1° There are "classical" mechanisms of Private International Law for the determination of the applicable law to the contractual obligations of the contracting parties. An example would be the traditional rule stated by the Convention of Rome, where the contracting parties have the freedom to choose the applicable law to their contractual obligations with some exceptions. If the parties have not made any choice, the Convention of Rome has created subsidiaries rules of conflict of laws.

2° There are also European Community “quasi-“ uniform material rules, related to legal aspects interested in the contractual aspects of the provision of the GEMSS services when trying to harmonize the laws of the Member States through the adoption of Directives.

Some of these “quasi-“ uniform material rules are even accompanied by classic rules of conflict of laws.

The mix of different rules of conflict of laws from different natures is quite confusing when trying to define the content of the contractual obligations of the contracting parties, for the provision of the GEMSS services.

This constitutes a major obstacle for an easy and low-cost day-to-day provision of the GEMSS services involving partners located in different Member States. Indeed it implies that the partners must invest energy and money to know the content of the applicable laws and therefore to understand the real content of their contractual obligations.

With respect to this, it is also useful to remind that it is widely stated that the diversity of contract laws in Europe is a major obstacle to the European Market.

However, the adoption by the European Union of “quasi-“ uniform material rules is definitely a very good step to support the provision of services such as the GEMSS services, even if small discrepancies between national regulations exist.

The simultaneous use of “quasi-“ uniform material rules with other mechanisms of Private International Law does increase the difficulty in defining the content of the contractual obligations for GEMSS partners, and therefore the cost of these services.

With respect to this, the indication of the applicable law by a classical mechanism of Private International Law is not an efficient solution to determine the content of the contractual obligations, even based on the choice of the contracting parties. Indeed, excepted for special cases, at least one of the contracting parties will have to consult a lawyer specialized in the chosen law to fully understand the content of her contractual obligations. This is quite feasible for important international trade contracts, but not so feasible for the provision of medical imaging services throughout Europe.

All these issues require the intervention of very specialized lawyers.

Hence the writing of a contract for the provision of the GEMSS services is a tricky thing with a lot of legal uncertainty, accentuated by the existence of numerous mandatory rules specific to each Member States concerning the Health Sector. These rules are generally ignored by the foreign contracting party e.g. Medical Law, Funding of the health cares, etc.

3.2.2.2.2.2 Contracts concluded by electronic means

Due to the adoption of Directive 99/93 on a Community framework for Electronic Signature and of Directive 2000/31 on Electronic Commerce, it is possible to consider that, in the extent of the conditions imposed by these directives, contracts may be concluded by electronic means for the provision of the GEMSS services.

- It results from the legal analysis that is not easy to define the content of the contractual terms for the provision of the GEMSS services on a day-to-day and low cost basis. On contrary it may be stated that it is easily to conclude a contract by electronic means for the provision of the GEMSS services.

These conclusions aim to suggest possible remedies to this first issue.

Numerous works have already been conducted to study this problem related to the disharmonies between national contract laws in Europe. Some of them have suggested practical solutions to solve these disharmonies.

With respect to this the Principles of UNIDROIT related to the contracts for international trade have to be reminded. Nevertheless these are not suitable for the provision of the GEMSS services.

On 26 May 1989 the European Parliament has adopted a resolution considering the beginning of the study of a European Code of Private Law (Resolution of the European Parliament of 26 May 1989, C/401 of 26 June 1989).

In 2002, the Pavesan Project has published the first part of a draft of a European Code of Contracts Professor GANDOLFI has started his collective work with a list of queries to the members of his group after a first meeting in Pavie in October 1990 (cf. GANDOLFI, G., dir., Code européen des contrats, avant-projet, livre premier, I, Académie des privatistes européens, Giufrré Ed., Italy, Milan, 2002, 576 pages).

The same Resolution has also stimulated another group of legal studies that has published the *Principles of European Contract Law,* under the direction of Professor Carlo CASTRONUOVO (Italy, Giuffré Ed., 2001, 597 pages).

In 2003 the European Commission has launched her 2003 Action Plan concerning the European Law of Contracts (Droit Européen des Contrats plus coherent – Un plan d'action, COM (2003) 68 final).

In 2003 the European Commission has also published a Green Book on the transformation of the Convention of Rome in a Community instrument and on its modernization.

Some authors state that the harmonization of the contract laws of the Member States is not essential to the construction of the European Market (cf. Fr. TERRE, Ph. SIMLER & Y. LEQUETTE, *Droit civil, Les obligations*, Paris, Dalloz, 7^e^ éd., p. 47, n° 42). They think that the harmonization of national contract laws in Europe should be limited to international contracts (international characteristic due to the nationality of the contracting parties, to the localisation of a pertinent element of the contract, etc.). This approach claims to be compliant with the principle of "subsidiarity". They suggest that this harmonization should take the shape of uniform material rules (Fr. TERRE, Ph. SIMLER & Y. LEQUETTE, *o.c.*, p. 48, n° 42 *in fine*).

Considering these opinions and considering the provision of medical imaging services through the Internet using Grid technology and involving partners located in different Member States, two possible remedies should be considered.

1° The adoption of a Regulation covering the creation of uniform material rules for all the contractual aspects concerning the provision of such services within Europe.

This would be the perfect solution to permit the use of the GEMSS services on a day-to-day basis at the lowest cost.

2° To use a model of contract referring to a corpus of rules easily accessible to the contracting partners (on a similar idea, cf. Droit Européen des Contrats plus cohérent – Un plan d'action, COM(2003) 68 final, p. 26, n° 89 & s.).

However the problem of the existence of mandatory rules specific to each Member States concerning the use of Grid technology for the provision of medical services still remains. In order to evaluate the importance of this last obstacle it could be very useful to ask to each Member States to inform the Commission on its specific mandatory rules preventing the use of medical services using Grid technology via, for example, the National Council of HealthCare Practitioners and the national Ministries of Public Health and Social Security.

The legal work provides a skeleton of contract based on the European rules concerning the contractual terms for the provision of medical imaging through the Internet based on Grid technology. This contract should be fulfilled according to the applicable laws when using the GEMSS services throughout Europe.

3.2.2.2.3 Liability approach

- The analysis of the European rules concerning the liability where using GEMSS applications is under process. These rules have been identified and studied in relation with the special aspects of the applications developed within the project. The writing of this report has started.

This analysis will be complete by the study of the ethical questions raised by the use of the GEMSS applications.

3.3 Workpackage 3: Medical Simulation System

This workpackage has the main task of providing access mechanisms for the GEMSS end users and integrating all the components together into a single system. Before launching into the construction of GEMSS, this workpackage is investigating the different means of providing ergonomic end-user access.

3.3.1 Sub-task 3.1: Portals and Access

During the second 12 months of the project, sub task 3.1 concentrated on its core objectives of surveying alternative technologies and creating a lightweight framework for incorporating all the client side components. The first objective was pursued by IDAC, ISS and NEC, while the second objective was pursued by ISS and NEC.

For the purposes of the survey work, three technologies were chosen for evaluation:

1. Web Services,
2. OGSA and
3. Unicore.

Then a template containing a standard set of questions was developed and applied to each of these technologies in turn. IDAC undertook the evaluation of Web Services, ISS evaluated OGSA, and NEC evaluated Unicore. During the course of the evaluations, IDAC, ISS and NEC each deployed a simple testbed for the technology they were surveying; hence, this was not a pure theoretical evaluation, but one based on actual (albeit limited) experience. The results of the survey work were described in deliverable **D3.1a** which was completed in mid December 2003.

Work on the light weight, client side framework continued throughout the year, cumulating in the final release of the framework as described in deliverable **D3.1b**, which was finished in August 2004. During this time period NEC designed and implemented a framework for containing the client side components and providing the applications with a basic interface into the GEMSS environment. The client framework automatically locates and instantiates all client side components, as well as providing communication and security mechanisms for the components and applications. The framework was implemented using an iterative design-develop-evaluate cycle in which features were added over the course of several iterations. This enabled the early release and use of the software by the other software developers.

While NEC was developing the client side framework, ISS developed several key components, namely, the *Basic QoS proxy* and the *Discovery* component. The *Discovery* component, which as finished in March 2004, enables applications to discover new GEMSS services by contacting a known service registry. The *Basic QoS proxy* , which was released in August 2004, allows applications to query the server about basic QoS properties. Both of these components are fully described in deliverable **D3.1b**.

3.3.2 Sub-task 3.2: System Integration and Testbed Deployment

Integration Methodology:

The work and actions of this subtask has been internally organized by an implementation and integration plan. The involved partners (ISS, ITN, and CCRLE) defined deadlines for both the development of software packages and their integration into the GEMSS environment. This approach allows the testbed functionality to improve incrementally over the project duration and also provides the project partners with a clear picture about the current and future installations. The alternative approach of having only very few releases, say an initial and a final version, had been dropped, since the consortium identified the following risk: the final version might suffer from unforeseen problems that could lead to significant delays. In particular, the important evaluation activities are dependent on the availability of a running GEMSS environment to perform their actions. Since these tasks require the allocated period of time, the project would run into severe problems. However, in the incremental approach the delay of a (final) version would simply mean that only some functionality is missing and thus limiting the overall risk to this extent.

Testbed Deployment:

Initial Phase: Already for the first review in November 2003, the involved partners achieved to set up and demonstrate an initial GEMSS environment. This environment was installed at the PC cluster systems at ISS and CCRLE. The initial version contains functionality to upload compressed input data from the client to the server, to start a particular GEMSS application and to download compressed result data after job termination. Already at this early stage SSL security and message level signatures are part of the GEMSS environment. The resource scheduling system is used only for simple job submission and execution operations. A component framework on client side is used to dynamically manage the client entities. At this stage two applications were integrated in the environment.

Phase 2: In February, all GEMSS applications were finally integrated. Additionally, optimizations for large data support are considered and implemented in order to reduce the time for moving data between client and server entities. Another important issue at that time is stability. Since in a real world setting issues like reliability and stability play a key role, we defined a robustness test scenario. The robustness test was designed to test each of the six applications, and the infrastructure, over an overnight period of 19 hours. The cluster at NEC was cleared of its usual jobs for a 19 hours period, allowing GEMSS testers full access. All six applications were then started off, running in non-interactive batch mode, each running with a single test dataset. Each of the applications submitted jobs sequentially, waiting for the first job to finish before submitting the next. Only two jobs failed for reasons that are not directly linked to the GEMSS software. In essence we can state that the GEMSS prototype testbed already demonstrated a considerable degree of stability already in this phase of the project.

Phase 3: Much effort went into the implementation and deployment of this phase. The current testbed is extended by a basic negotiation and QoS module. The QoS module itself is strongly linked to the resource scheduling system in order to query possible – future - allocations (→

advanced reservation). The negotiation module selects the most appropriate offer relative to a underlying protocol. Additionally, security was further improved by introducing message level encryption, the so called advanced security. At the time of writing this, phase 3 two GEMSS plus a test service are deployed at CCRLE and ISS.

Known Problems:

Integration has proved to be the most time consuming task for a prototype release since it involves cross-partner co-ordination and the use of developed modules previously unseen be each partner. Initial prototype integration involved sequential integration, where one partner would integrate a developed code module and pass the integrated system on to the next partner. This proved inefficient as the project progressed, so for the later prototypes we have moved to using a co-ordinating partner and parallel integration threads.

The development status of each module is summarized in the ST1.2 section, and described in more detail within deliverable D1.2b. A number of lessons have been learnt from the phase I3 integration and we expect the final prototype integration task to be significantly less effort. These lessons include:

1. Identifying a single partner to co-ordination integration of the client and server work.
2. Revision of the server build environment to support easier integration of new modules.
3. Revision of client and server build scripts to automate unpacking and use of new module distributions. Automation will remove human error when copying jar files etc to their appropriate locations.

Application Integration:

The task of application integration has been, to date, taken on by ISS and each of the application developers. Each application developer must generate, per service provider, a set of scripts to start a job, get the status of a job and kill a job. In addition application developers must generate a performance model that ISS's QoS software can invoke to estimate execution time for specific jobs. Application integration has proved less time consuming than the infrastructure module integration, with the application developers evolving their scripts and performance models once the initial work of understanding what is required has been completed.

3.3.3 Sub-task 3.3: Grid-based Support and Consulting

Work carried out in project year two:
During the last 12 months the bulk of the evaluation work was carried out in this subtask D3.3. – Grid-based support and consulting options. Sixteen tools were evaluated in total. The evaluation work took up most of the time, as installing and testing each tool was quite time-consuming. A number of subjective and objective criteria were chosen with which to evaluate the various support and consulting tools as follows:

- Installation,
- Ease of use,
- Cost,
- Availability,
- Compatibility,
- Security,
- GRID Features.

The tools were subdivided into the following subcategories:

- Remote Control Applications,
- Web Conferencing,
- File Sharing & Groupware and
- Complementary Tools.

All tools were evaluated by IDAC staff and a record was kept of the results and included in the report. A number of recommendations were made after the evaluation had finished. These were based on IDAC's viewpoint as an SME and may not be appropriate to other organizations. A table was also drawn up showing the cost of each tool and if IDAC used the tool or not.

Work planned:
IDAC will continue to work with the other partners to implement the Access tools.

3.4 Workpackage 4: Medical Service Applications

This workpackage is responsible for the adaptation and integration of the medical simulation and image processing software into the GEMSS testbed. It includes a variety of medical service applications which

- have different GRID requirements concerning computation time (near real-time requirements vs. standard batch processing), memory usage, encryption, etc. and
- address different medical areas (cranial, pulmonary, cardio-vascular system) to address an end-users group of sufficient size for service provision.

Name	Domain	Class	Users
Maxillo-facial surgery simulation	Medicine – pre-surgical planning	Distributed supercomputing / On demand	Medical doctors, researchers
Neurosurgery support	Medicine – intra-operative planning	On demand	Medical doctors, researchers
Radiotherapy planning	Medicine – Monte Carlo treatment simulation	On demand / distributed supercomputing	Medical end-users; Doctors, researchers
Inhaled drug delivery simulation	Medicine – air flow dynamics	On demand / distributed supercomputing	Medical end-users; Doctors, researchers
Cardio-vascular system simulation	Medicine – blood flow dynamics	On demand	Medical end-users; Doctors, researchers
Advanced image reconstruction	Medicine – nuclear / in vivo diagnostics	On demand	Medical end-users; Doctors, researchers

Figure 3.11 Classification of the 6 GEMSS Medical Service Applications

3.4.1 Sub-task 4.1: Maxillo-facial Surgery Simulation

Medical background:
In patients suffering from severe maxillary hypoplasia and retrognathia, conventional therapeutic surgery often fails to guarantee long-term stability. Using a rigid external distraction system for midfacial distraction osteogenesis is a new method to correct the underdevelopment of the midface, surpassing traditional orthognathic surgical approaches for these patients. Currently surgical planning is only based on CT images. The treatment consists of a midfacial osteotomy (bone cutting) followed by a halo-based distraction (pulling) step. The goal of this sub-task is the modeling of this distraction process to allow predictions on its outcome.

Tool chain outline:
A simulation of the maxillofacial surgery involves a lot of individual tasks, which may be outside the surgeon's expertise. Therefore, emphasis has been put on automating this chain as far as possible. The steps of the planning simulation are as follows:

1. The image data of the patient is acquired (typically CT, in DICOM format)
2. The image data is converted into a format understood by the tools (Vista format) and interpolated to isotropic voxels
3. The image is segmented into bone and (one or several kinds of) soft tissue. Alternatives are using a thresholding approach and using a template-based registration approach.
4. A surface mesh of the bone is generated
5. The surgeon interactively specifies bone cuts and displacements on the surface mesh (virtual osteotomy)
6. The cuts and displacements are used to create a 3D finite element (FEM) model of the problem. This includes mesh generation and application of boundary conditions.
7. The FEM problem is solved by a simulation code.
8. The results are visualized and interpreted by the surgeon.

Workflow:
As the toolchain consists of several independent tools, some means of coupling them has to be applied, as well as some means of allowing the user to interact and customize the workflow. We chose the Triana workflow editor for this purpose. The individual tools have been wrapped into Triana units, forming building blocks for constructing toolchains using a graphical programming language. Our toolchain contains 3 major points of user interaction: First, the (optional) non-linear registration requires the definition of landmarks on the rendering of skull and skin surface. Second, and most importantly, the surgeon has to define the surgery operation by specifying cuts and displacements. And finally, the results are visualized. Between this interaction points, substantial data processing is going on. Therefore, we have defined several chains which allow the job to resume execution at a later stage, without having to wait for the next interaction.

Grid enabling:
The task with the largest requirements of computer resources is clearly the FEM simulation; it has to be carried out on a remote high-performance computing (HPC) platform, unless one can accept a rather coarse approximation. Other tasks like mesh generation and non-linear registration are also demanding, but can be carried out on the client side. In principle, these tasks are also eligible for remote execution; however, only a parallel implementation of these

tools will give substantial gains.
This scenario can be termed *thick client.* A different scenario is the *thin client*, where everything runs on the server, and the client uses a remote control tool like VNC to connect. Currently, we are investigating a variant of this solution to provide remote assistance to our medical partner (see below).

In order to provide Quality of Service (QoS), data has been gathered allowing the prediction of the runtime and memory consumption of the remote FEM solver. The integration into the GEMSS QoS framework is currently being tested. During execution of the remote simulation, the client can query continuously its current status and get an updated estimate of the remaining time until completion.

Effort has been put into making the solver more robust, eg. by automatic timestep adaptation to avoid convergence failure. In general, however, making a (nonlinear) solver robust requires a number of advances techniques like remeshing and automatic parameter and algorithm selection, which are beyond the scope of this project. We believe however that these issues are critical for the success of application scenarios like ours, where simulations are run by non-experts.

Evaluation and Test cases:
Synthetic tests have been set up allowing to quickly test client and server functioning.
In order to ensure clinical testing and evaluation by a medical practitioner, a collaboration agreement has been signed with Dr. Hierl, Department of Oral and Maxillofacial Surgery, University Clinics Leipzig. A Linux workstation equipped with the client software has been installed at Dr. Hierls office. Several improvements suggested by Dr. Hierl have already been incorporated into the toolchain.
A graphical landmark-picking tool has been implemented allowing the definition of currently 33 different landmarks on the patients skull and skin. This tool will allow quantitative measurements of the simulation errors.

Status summery:
A complete version of the toolchain is running at several sites, including a clinical one.

- Integration into GEMSS middleware
- Integration into Triana workflow, several customized workflows created
- Integrated tool to convert DICOM images to the Vista format internally used
- Rigid registration of patients-to-model for mapping auxiliary data like clipping planes (still testing)
- Interactive landmark-definition tool with user-guidance created
- Landmark-based non-linear registration implemented
- Fully functional version of interactive cutting tool implemented, allowing to move several components independently
- Volume mesh generation allowing smooth interfaces between several materials
- Performance model for predicting runtime and memory needs of remote FEM solver
- implemented
- Implemented time step control for non-linear solver
- Enhance visualization based on volume rendering

Next steps:

- Set up VNC for remote assistance
- Test integration with GEMSS QoS framework (work in progress)
- Evaluation of simulation accuracy
- Fully integrate rigid/nonrigid registration

Summary:

The maxillo-facial surgery application has reached a fully functional stage and can be used by clinical practitioners. Feedback from our clinical partner Dr. Hierl has already resulted into substantial improvements to the toolchain, additional minor modifications can be expected during the evaluation period. The applciation has been integrated into the basic GEMSS framework; integration into the new QoS framework is under way. Still missing is a full integration of the non-rigid registration into the chain; this will be useful both for optimizing the toolchain in terms of computational efficiency and user friendlyness, and for evaluating the quality of the surgery prediction by the simulation.

The GEMSS approach and the toolchain for maxillo-facial surgery simulation has been presented at a number of workshops and conferences, for a complete list see the section on dissemination below. It is planned to continue development of the maxillo-facial surgery simulation past the end of the project.

3.4.2 Sub-task 4.2: Neuro-surgery Support

Background:
The major shortcoming of image-guided surgical planning based on pre-surgically acquired functional MRI (fMRI) data is the brain shift phenomenon. The occurrence of surgically induced deformations invalidates positional information about functionally relevant areas. This problem is addressed by non-linear registration of pre-operative fMR images to intra-operative MRI acquired by an Open-MR scanner, or to intra-operative 3D ultrasound data.

Workflow and Grid Enabling:
As requirements for the chain to work, a pre-operative MR scan together with its aligned fMR data of the patient is needed. During the first stage of the surgery, before the skull is opened, an usually low-resolution image with the Open-MR scanner is acquired. After the correction of possible RF-Field inhomogeneities a linear registration of this image with the anatomical high-resolution pre-operative data takes place. The registration parameters are stored as starting position for further linear registration steps. The registered image will be the reference for further steps of the chain.
After the opening of the skull further intra-operative images are acquired. These images are also corrected with respect to possible intensity inhomogeneities and registered with the first intra-operative data using the stored parameter set. To gain the same intensity distribution in both input images, a linear intensity adjustment might be made. Then the non-linear registration will be executed. The resulting displacement field of the non-linear registration process is applied to the pre-operative fMR data. In the last step the deformed fMR data will be overlaid to the linear registered open-skull data set and later sent to a presentation device.
The fluid based non-linear registration method produces best results if its input images are originating from the same scanner. That is the reason why the first (closed skull) intra-operative image has been acquired and used as a reference image for further processing and not the pre-operative high-resolution data which is usually not acquired with an Open-MR scanner.
At the beginning of the project, processing time for this chain was about 4 h (Intel Pentium III, single processor). To avoid a too long delay of the progress of surgical intervention, a maximum processing time of approx. 10 min is acceptable. The time-consuming registration steps are readily parallelisable on shared or distributed memory high performance computing platforms. It is planned to compare the performance of this processing chain computed on single processor machines (for off-theatre validation), local workstation clusters, and remote high performance computing platforms. Grid technology allows switching between environments easily.

Elements of the Image Processing Chain and their Current Status:
The currently available image processing chain focuses on the registration of low-resolution Open-MR data to low- and high-resolution MR data:

- Data transfer and conversion: Not implemented due to the lack of an Open-MR scanner. Data is expected to be in the machine-independent Vista format. Converters are available which can easily convert images from DICOM in Vista format.
- Correction of intensity inhomogeneities: This step is very time consuming due to the solution of a huge linear system. To improve speed the linear system is only coarsely solved within a multi-resolution framework, which is sufficient to estimate a proper bias field. A further speed improvement was obtained during operating on an enlarged head mask that can be extracted from the given dataset. The biggest improvement of

performance was gained by computing the solution of the linear system in parallel. Finally, the algorithm has been enhanced to work in a multiprocessor shared memory environment yielding nearly linear speedup. This was realized by an overlap of iterations: The next iteration is starting as soon as all necessary data has been computed, although the current iteration did not finish yet. This is the reason why the maximum number of usable processors is bounded by the number of slices in the MR image. A different strategy had to be applied in a distributed memory environment. Here, the data set will be partitioned into blocks and each block is solved independently from the others.

- Linear registration: The registration of the low resolution intra-operative dataset to a high resolution pre-operative image is realized by maximizing their normalized mutual information (NMI) resp. cross correlation (CC). NMI is used when data from different scanners is going to be registered and CC is used for datasets originating from the same scanner. The registration is done in a multiresolution framework which produces a significant speedup. To achieve a fast convergence, the down-hill simplex optimization algorithm is used, which performs well and does not require any gradient information. To improve speed of the registration, a parallel evaluated speculative down-hill simplex was developed that needs the same number of iterations to converge as the original method, but twice as fast. To reduce the risk of finding local optima instead of the global optimum a parallel evaluated genetic algorithm is used on the lowest multiresolution level. By partitioning the data into blocks and evaluating the cost function (NMI and CC) in parallel, a further, nearly linear speedup was gained. The results of the processed sub volumes are cumulated to obtain the cost function value. All the parts of the linear registration that have been mention above, can be computed in a shared and distributed memory environment.
- Intensity adjustment of two scans: The result of the non-linear registration step of the processing chain depends, among other things, on the similarity of the intensities of both images. Since this step is not very time consuming, it can be performed in serial.
- Non-linear registration: To obtain a deformation field that can be applied to a fMRI image a non-linear registration is required. In this chain a method based on fluid mechanics is applied. The time consuming part here is once more the solving of a huge linear system. To speed its evaluation up and to avoid local minima the system is solved using a multi-resolution approach. A further speedup is achieved by solving the system in parallel, in a shared memory environment. Here each processor operates on a single "slice" of the system. If there is only a distributed memory environment available, all multi resolution steps - except the one at the highest resolution level – are computed in serial. For the last resolution level, that is actually more time consuming than all the other steps before taken together, the data is again partitioned into blocks. They are processed in parallel and therefore solved independently from each other. To avoid partitioning artefacts, the blocks have an overlap of 2 slices on each border.
- Application of a deformation field to fMRI data: The deformation field obtained by non-linear registration will be assigned to a fMRI dataset by shifting each voxel of the dataset by its corresponding vector from the displacement field.
- Overlay of deformed fMRI data with intra-operative dataset: In this step the deformed pre-operative functional dataset that was initially aligned with the pre-operative anatomical MR scan will be overlaid to the registered intra-operative image to show regions of activation with respect to the brain shift.
- Conversion and transfer to presentation device: A graphical user interface has been developed which shows the results of the image processing chain on a monitor.

Status Summary:
Due to the requirement of processing the whole chain (including image acquisition) in less than 10 minutes, large computation power is needed. This will be usually provided through a local cluster with multiple processors. Those clusters are quite expensive, need maintenance and are often under-worked. Instead of a cluster the grid could be used to execute the chain on a currently available computation centre. For grid interaction, a cheap local terminal is fully sufficient.

Now the image processing chain takes about 10 minutes on a AMD AthlonMP 1.6 GHz distributed memory PC cluster to be executed over the grid using the GEMSS middleware in phase 2. To achieve this execution time of 10 minutes 10 to 12 processors have to be used.

Currently the development and testing of a GEMSS middleware phase 3 compliant client is in progress.

3.4.3 Sub-task 4.3: Cranial Radio-surgery Simulation

Application Scenario:
Gamma Knife® Radiosurgery is a non-invasive medical procedure using beams of ionising photons from 201 ^{60}Co sources to treat intra-cranial lesions. The Gamma Knife® unit comes with a treatment planning system, GammaPlan® that uses an approximate description of the photon interactions within the head of the patient to calculate the energy dose deposited by these photons in the region of the tumour. There is significant benefit to be obtained from improving the fidelity of these calculations, particularly in cases where photons traverse regions of widely differing electron densities (e.g. soft tissue and bone). Monte Carlo calculations provide such a model and can be used initially to complement GammaPlan®, with the potential to eventually supersede it in the event of short enough calculation times. The goal of this sub-task is to adapt a Monte Carlo code (RAPT/EGS4) written for conventional radiotherapy to that of stereotactic radiosurgery, for calculation of the energy dose delivered to the brain from a Gamma Knife® treatment unit.

Workflow:
The Grid-enabled radiosurgery application uses RAPT as a front end to the EGS Monte Carlo engine to model ionising radiation transport through the head of the patient. It requires:

- Definition of patient geometry
- Specification and distribution of material types contained within the geometry
- Position of beam isocentre
- Beam properties – intensity profile, spectrum
- Beam distribution – number of beams and their arrangement
- Quality of simulation parameters – total number of photons, interaction types

This data is specified by the contents of numerous text files that are loaded into the EGS solver at startup. The files are not native to EGS but are created for EGS through use of a bespoke RAPT GUI. All input files are stored in a default *Input* working directory, zipped, encrypted and dispatched to the Grid. The simulation process simulates the dose given to the region of interest by modelling millions of photons, and following their paths, employing information from photon scattering data, to correctly give the photons their positions, angles of deflection, and energies or to absorb them in the tissue as they interact with the atoms of the target. The energy distribution within the geometry equates to the dose distribution of the model. Thus the output from the modelling process is:

- 3D patient mesh geometry
- Accumulated dose at each element of the solution mesh
- Flux variance at each element of the solution mesh

The results archive is pulled from the Grid in encrypted form and unzipped on the client within the working *Outputs* directory. This data is accessible to the user as a set of text files, but a visualisation application has been written within MatLab to enable the user to explore the distribution of radiation dose within the treatment volume. The dose data can be represented as contours on specified planes through the volume mesh or visualised as isosurfaces in 3D. Additionally, a metrics utility has been developed to permit quantitative comparison of plans. This is useful when comparing the merits of different proposed treatments or comparing the dose distribution obtained by GammaPlan/experimental methods with the solution obtained by RAPT.

Grid Enabling:
The RAPT software has been run and tested under LINUX and Windows NT environments. Integration of RAPT with the planning process in radiosurgery has been accomplished by creation of a bespoke GUI targeted for compatibility with GammaPlan. Secure data transfer relies on X.509 protocols that includes a Registration Authority appointed in the Medical Physics Department at Sheffield University. A data controller within the hospital has been identified and operation of secure RAPT from within the clinical radiosurgery department has been demonstrated. An important goal of the project is to evaluate the software within the clinical domain, and preliminary clinical example results computed by RAPT are very encouraging. The hospital firewall is very tightly monitored, so it is not possible to 'push' data from an external source into the NHS network. However, the ability to 'pull' the data from within the bounds of the hospital firewall can enable the running of remote Grid jobs from the hospital environment, and this is being explored. It is the reliance of GEMSS middleware on Web Services that makes this transaction possible.

Data security is paramount and utilises 128-bit encryption. The QoS model is designed to let the user specify the time of job completion and uses an estimate of job run time based on benchmark data supplied by preliminary evaluation tests. Timely solution is critical in this clinical application, aided by the ability to assess job status by polling the Grid to determine the state of the simulation while it is running. Currently a simple business model is supported by the middleware, and involves negotiation of service provider and cost prior to commencement of each job.

Evaluation: Sample Problems and Test Cases
Suitable test cases have been sought that:

- Confirm accuracy of the RAPT/Monte Carlo solver
- Permit assessment of the efficacy of the RAPT radiosurgery planning process in GEMSS
- Act as a metric by which the RAPT peformance can be judged, and may act as a suitable QoS benchmark.

Over 400 jobs have been submitted to the Grid as part of a preliminary evaluation exercise. They can be broadly grouped into four different categories as outlined below:

i) Simulation of radiation transport through water. Evaluation is through assessment of linear attenuation coefficient etc. and comparison with published data.
ii) Simulation of dose distribution in a spherical head phantom. Evaluation is through comparison with the GammaPlan computed dose distribution.
iii) Experimental validation of the computed dose distribution obtained using a spherical head phantom. Evaluation is by comparison of the RAPT solution with the experimentally determined dose data.
iv) Simulation of dose distributions in clinical treatment plans. Evaluation is through comparison of results between GammaPlan and RAPT.

Several months of preliminary evaluation have confirmed that Grid-enabled RAPT is robust and able to produce clinically relevant results of acceptable accuracy in under an hour (at least 30 processors are needed - compute time is a function of photon statistics and available processors). Grid reliability peaked at 80% during this period, whilst RAPT scalability was observed to be encouragingly linear. However, even with Grid assistance it is clear that iterative planning with RAPT is not possible since it is insufficiently responsive to provide rapid dose distributions for the iterative planning process (a speed increase of several orders of magnitude is required if RAPT is to compete with the 'real time' response of GammaPlan).

With solve times less than one hour, clinical experience demonstrates that Grid-enabled RAPT has a valuable role to play in confirming the dose distribution determined by GammaPlan, particularly in those circumstances that contravene the assumptions inherent within GammaPlan. Under these conditions the doses calculated by the two planning methods may differ and consequently RAPT has the opportunity to have a significant impact upon patient management.

Status Summary:
This summary outlines key features of the GEMSS RAPT service accomplished within the project to date:

- Grid enabled RAPT accommodates all aspects of a clinical treatment plan through an intuitive user friendly interface
    - 201 beams with support for beam plugging
    - Multi-shot treatments
    - Support for Bubble head geometry
    - Arbitrary positioning of isocentre
    - Support for 4 material types (vacuum, air, tissue, bone)
    - Support for calculation of treatment shot times

    - Current principle limitations of RAPT are:
        - 1mm maximum spatial resolution
        - 200x200x200mm target/patient geometry volume
        - No CT/MRI skull geometry support

- Documentation is provided with this software, and includes a tutorial guide and supplemental technical information
- A results visualisation GUI has been developed that permits exploration of dose within the head geometry.
- A metrics utility has been developed that enables direct comparison between RAPT, GammaPlan and experimental dose data from experimental film measurement.
- The RAPT solver has been proven to be accurate and robust
- Solutions utilising in excess of 500 million photons are required for clinical acceptability (statistical errors < 3%).
- Multiple shot treatments with closely co-located isocentres can be solved within RAPT to clinically acceptable accuracy in under an hour (30 processors)
- Comparison of RAPT with GammaPlan and experimental data indicate that RAPT offers improved description of dose in those circumstances in which the assumptions relevant to GammaPlan are contravened.
- Application development has been driven strongly by clinical recommendations. These have influenced the development path of the software
- The Grid enabled application fulfils the requirements specification of D1.1
- A MatLab licence (version 6.5) is required for the RAPT client.
- The service complies with the GEMSS security infrastructure (X.509 compliant)

Next Steps:

- Integration with the most recent release of the GEMSS middleware (Phase 3), permitting:
    - Improved job status control
    - Estimation of solution time (QoS feature – performance model)
    - Improved job scheduling (QoS feature - scheduling)
    - Choice of compute server (QoS feature - negotiation)
    - Grid interaction through a well defined and transparent business model

- Extensive evaluation that features:
    - Clinical plans
    - Feedback from the clinical environment
    - RAPT computed dose comparison with known dose distributions
    - Documented Grid reliability
    - Documented Grid-RAPT performance

Summary:

Grid-enabled RAPT is now operating successfully on the GEMSS Grid. It is apparent that the large Monte Carlo problems encountered in Gamma Knife radiosurgery can be solved effectively through the computing resources provided by GEMSS. The accuracy of the simulation results is a strong function of computation time and resources, and the Grid offers a significant advantage in this respect. A preliminary period of evaluation has demonstrated the utility of Grid enabled RAPT in a clinical environment, and has implications for patient management in those cases in which the assumptions of GammaPlan are contravened.

A dissemination programme that includes publications, conference presentations, and demonstrations is progressing steadily. RAPT can have clinical impact in the radiosurgery environment and therefore continued funding beyond the end of the GEMSS project is being pursued.

3.4.4 Sub-task 4.4: Inhaled Drug Delivery Simulation

Application Scenario

The inhaled drug delivery simulation has adopted the acronym 'COPHIT' – '**C**omputer **O**ptimised **P**ulmonary delivery in **H**umans of **I**nhaled **T**herapies'. It is an acronym which emphasises that optimised delivery of medication to the lungs can be achieved by modelling the drug delivery process. Knowledge of the characteristics of the delivery device (eg. inhaler) and the airways is required, as are the physical characteristics of the medication. These elements determine how medicament is entrained in the airflow and how it is deposited on the airway walls. The software allows drug designers or manufacturers of inhalation devices to experiment *in silico* with device geometry, delivery timing, pharmacokinetics or physical formulation of the medication, and thereby maximise anticipated dose to the desired region of the respiratory tract.

The CFX computational fluid dynamics solver is the means by which such problems can be solved. It permits computation of the inhaled flows and drug deposition based on real breathing profile data and real lung geometries obtained from CT scans. However, the solution of inherently complex physiological problems allied to respiration is fraught with difficulties. The presence of Cophit-motivated coupled compartments designed to improve the fidelity of the simulation only complicates matters further. As a whole, such respiratory simulation requires a level of expertise and computing power that makes it completely inaccessible to all but the most expert of users.

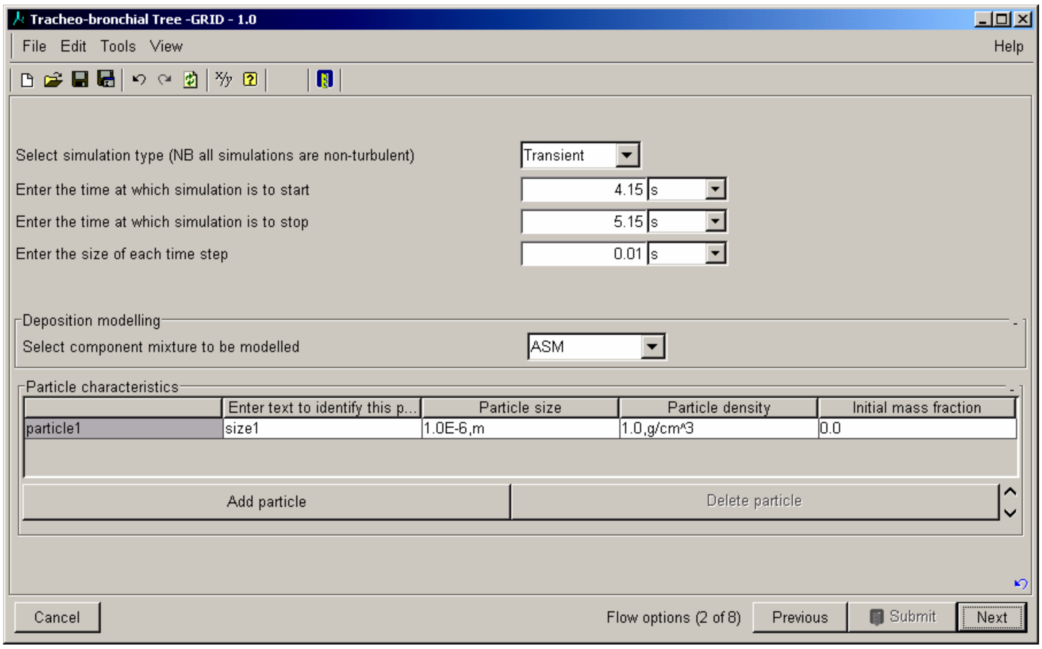
It is for this reason that GEMSS offers an Inhaled Drug Delivery Simulation *service*. The ethos of such a service is that the expertise required for this application resides with the developer (e.g. University of Sheffield Medical Physics, or consultancies such as IDAC and ASD). The developer, with the availability of additional tools, is in a position to create a bespoke application tailored to the needs of the customer. This is made possible through the creation of an interface (using EASA) that is able to hide the complexity of the simulation behind a simple point-and-click GUI, *but it does require that the problem can be expressed in parametric form.*

Workflow

The parametric approach makes respiratory simulation readily accessible to the end user, whilst the Grid meets the requirement for accessible large scale computing resources. It is a powerful combination. The primary disadvantage is loss of flexibility, since users are unable to explore issues that are outside the scope of the GUI that was originally designed for them. Solution of wider problems is possible, but requires construction of a new EASA enabled application. From our experience with EASA, building an application from scratch is a significant undertaking, but subsequent modifications are relatively easy to implement and the modified application represents a new bespoke application for the client.

The application requires the user to enter relevant data in a series of steps spread over eight data entry screens as follows:

- Screen 1 - Confirm choice of lung mesh geometry
- Screen 2 - Specify time-step information and material properties
- Screen 3 - Specify coupled alveolar compartment properties
- Screen 4 - Specify time varying inlet flow/pressure and substance quantity
- Screen 5 - Specify initial conditions for the 3D flow field
- Screen 6 - Specify pharmacokinetic model parameters
- Screen 7 - Specify the nature of the results to be displayed
- Screen 8 - Choose solver (local or on the GEMSS Grid)



An example data entry screen provided by the Cophit GUI

The X.509 environment of GEMSS secures data transmission between the Client and the Grid server where the computationally intensive fluids problem is solved. The results data is returned to the client and presented in numerical and graphical form. Tools are available for easy visualisation of flow data, systemic drug uptake etc. and there is the option for interactive examination of drug deposition on a 3D *vrml* model of the lung mesh. In addition, all data is saved in tabulated ASCII format for direct import and manipulation in packages such as EXCEL.

Grid enabling

The EASA client is a customised interface that accepts numerical data entry from the user for the purposes of defining and submitting a Cophit job. This approach simplifies interaction with the Cophit environment. Note that the purpose of the inhaled drug delivery simulation service of GEMSS is to embed the user's requirements of Cophit in parametric form, as an application residing on the EASA server. At the point of job submission, the parameters are passed from the EASA client to the EASA server hosted on the Grid client. The user has specified the properties and initial conditions for all compartments, following which the model information is assembled into a 'DEF' file, which is the input for the CFX computational fluid dynamics solver. Auxiliary text files specify the compartment properties, and the variation of input flow rate and drug mass over time. These files are encrypted and sent to the Grid server, which performs checks on these files. Note that all executable files reside on the Grid server and none are transmitted to it. Consequently, a malicious user at the EASA client has little opportunity to cause mischief since their input is tightly constrained. The Grid server performs the computationally intensive fluids solution step and post-processing, and the results files are retrieved later. Because this software is used as a research tool, solution is not time-critical, unlike some of the more clinically-orientated applications. A typical job might take of the order of a week to run. It is anticipated that Grid resources would be used at periods of low demand in order to reduce the cost of running the job, which is currently based on a telephone service model.

Preliminary Evaluation
For the end users within GEMSS (represented by ASD and IDAC), a typical respiratory simulation scenario has been constructed. The EASA application permits parametric exploration of the following problem:

> *"For arbitrary flow and mass fraction inhalation profiles, what is the likely deposition of particulates (of arbitrary size or composition) within the central and peripheral airways, and how might this be reflected in systemic drug uptake?"*

The application has been installed on the Web server at IDAC's headquarters in Ireland, and has been used to solve example problems on the Grid with run times from minutes to hours.

For the application developer (represented by the University of Sheffield in GEMSS) it is important to imbue confidence in the application by establishing that Cophit can provide credible solutions. Therefore, effort has been directed at numerical simulation of experimental studies performed within the Medical Physics department at Sheffield, and also scenarios discussed in the scientific literature. In both cases, agreement with Cophit is encouraging. Scalability tests indicate that Cophit run time is enhanced by approximately 5-10 times with up to 16 processors, but the benefits rapidly diminish beyond this.

Feedback from industry contacts, such as Astra Zeneca, Aventis Pharma and Boehringer Ingelheim, etc. indicates that the Grid-enabled Cophit software is considered to be easy to use. Demonstration of the software generates sufficient interest that it often precipitates a request for additional features (which would require the building of modified EASA applications). A recurring appeal is for demonstrable evidence that Cophit can deliver what it claims, underlining the need for validation exercises that can promote confidence in the technology. The emphasis that GEMSS places on security is commended, and in general the provision of a Grid computing platform is viewed positively – provided that it is accompanied by a clear and well-defined business model.

Status Summary

This summary outlines key features of the GEMSS inhaled drug delivery service accomplished within the GEMSS project.

- The inhaled drug delivery service is implemented as a bespoke application negotiated between an expert developer (typically the portal provider) and the end-user
- The application has been made accessible to the end user by providing a user-friendly interface through EASA.
- The application has been partitioned into local-client and Grid-server sections.
- The service complies with the GEMSS security infrastructure (X.509 compliant)
- Enhanced security measures have been implemented on the Grid server to check incoming files from the portal for malicious code.
- Application development has been partially driven by industry recommendations which have dramatically altered the original GEMSS application profile
- The Grid enabled application fulfils the requirements specification of D1.1
- A MatLab licence is not required for any client or server system (although this is contrary to the original design specification, it resolves MatLab licensing issues)
- The documentation provided with the software release includes a tutorial guide and supplemental technical information
- A COPHIT demonstration CD including PowerPoint and html-based presentations has been developed as a means of introducing COPHIT to prospective users. It shows the steps of the solution process and demonstrates how COPHIT is used to solve inhaled drug delivery problems.

Next Steps:

- Closer integration of the EASA application with the most recent middleware release (Phase 3), providing:
  - improved job status control
  - improved estimation of solution time (QoS feature – performance model)
  - improved job scheduling (QoS feature - scheduling)
  - improved choice of compute servers (QoS feature - negotiation)
  - integration of a well defined and transparent business model

- Evaluation
  - Validation studies
  - Simple and complex EASA jobs
  - Assessment of Grid performance

Conclusion

A Grid-enabled COPHIT application is now operating successfully on the GEMSS Grid. It is apparent that large problems (i.e. ones that exceed the computational capacity of small business PC networks) can be solved securely through the computing resources provided by GEMSS. The accuracy of the simulation results is a strong function of computation time and resources, and the Grid offers a significant advantage in this respect. The latter assertion can be confirmed by a suitable period of evaluation, the desirable outcome being confidence in the ability of the Grid to deliver effective computing and the ability of the software to deliver effective and accessible respiratory simulation. A dissemination programme that includes publications, conference presentations, industry visits and demonstrations is well advanced and is an essential adjunct to evaluation, since it is the means by which the profile of the Grid can be raised and the fulfilment of its promise promulgated.

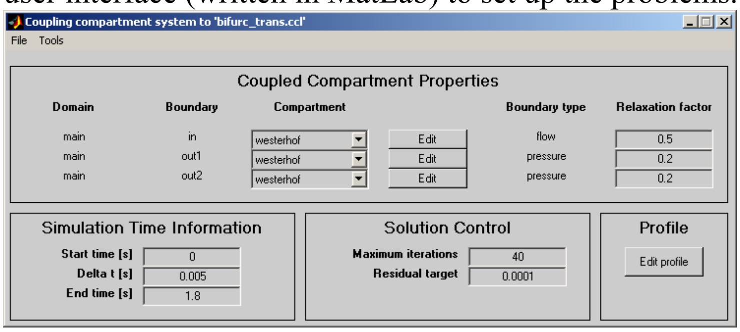
3.4.5 Sub-task 4.5: Cardiovascular System Simulation

Application Scenario

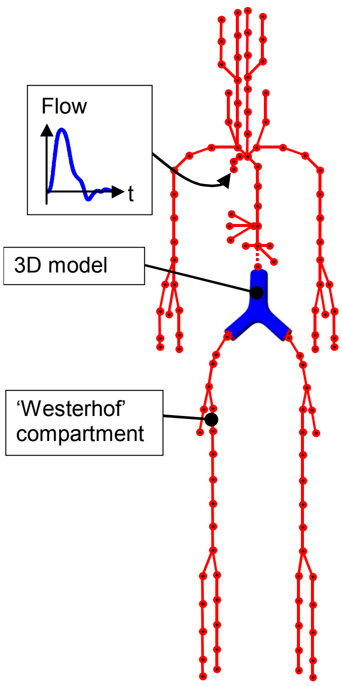
Modelling of blood flow in arteries requires the use of sophisticated three-dimensional computational fluid dynamics (CFD) software. Using such software, flow through isolated vessel sections can be simulated to provide insight into pathologies of the heart and vasculature. However, even if the flow in only a section of the vasculature is of interest, the properties of the whole vasculature should be taken into account, because effects like peripheral vasoconstriction can have a profound impact on flow anywhere within the cardiovascular system. Unfortunately, as more of the vasculature is included within the CFD simulation, the number of vessels increases exponentially and the computational problem becomes intractable. Thus it is necessary to reduce the magnitude of the problem by encapsulating the properties of the peripheral circulation in a compartment model. In this way the problem becomes computationally tractable. Even so, transient simulation of just a small section of artery is computationally demanding, and benefits from HPC resources accessed through a grid. This application couples the full three-dimensional CFD model of a vessel section to terminating compartments and builds on principles developed in ST4.4. The software has been designed to be extensible so that new types of compartment system can be added if so required. Whereas ST4.4 was EASA-driven, we chose to use MatLab for the cardiovascular application because of its flexibility and provision for the sophisticated mathematical calculations required by the compartment systems.

Workflow

The specification of the three-dimensional model is performed using CFX – a commercial fluid dynamics package. The software developed for this work package takes a CFX model and couples to it compartment systems representing the remainder of the circulatory system. The software has been implemented to have very general applicability, and it is possible for expert users to add their own compartment types. This gives great freedom in the choice of problem, as is necessary for research use or novel consultancy applications. The software is flexible, but nevertheless it has been made as user-friendly as possible through the use of a graphical user interface (written in MatLab) to set up the problems.

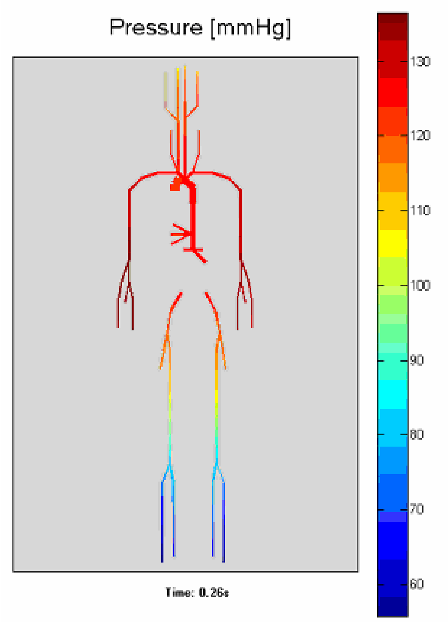


User interface for adding coupled compartments

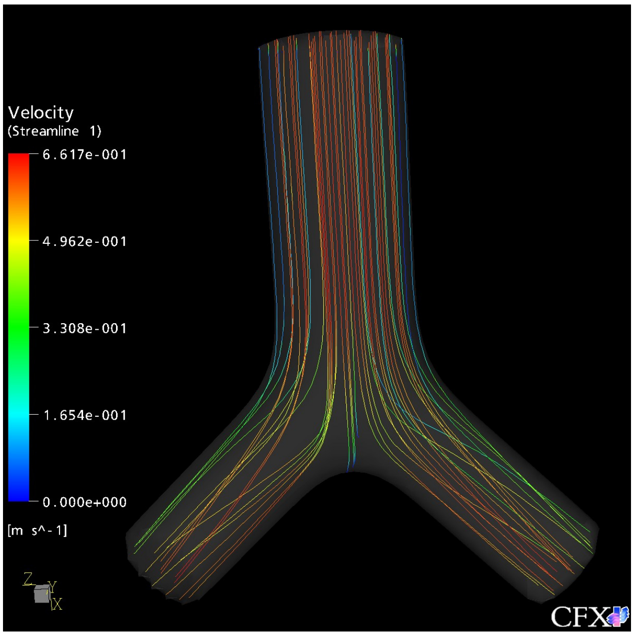


Example Coupled Simulation

Once the problem has been defined, input files are generated and submitted to the GEMSS client middleware, which authenticates the user to the Grid server and encrypts the input data for transmission to the Grid server. The Grid server performs the computationally intensive fluid dynamics simulation step, and solution progress can be monitored using a job management utility. The results data is returned to the client for local visualisation, and includes both 3D flow results (visualised using CFX), and compartment system results (visualised using custom-written software).



Compartment system results



Flow streamlines visualised using CFX

Grid Enabling

The cardiovascular simulation application has been split into two parts, the client GUI and pre-processing code, and the server-side simulation code (CFX + User FORTRAN code to implement the compartment system). The server side code is installed at the service provider site. The client GUI is written in MATLAB, and this connects to the GEMSS infrastructure through a loose coupling, where calls to batch script files are made to perform grid actions (upload/start job, status enquiry, abort, and download results). The client GUI is divided into three interlinked parts – pre-processing, solution control, and post-processing. The pre-processing GUI allows the problem to be set up and then submitted for processing on the GEMSS Grid. A session file is created once a job has been uploaded so that the client does not have to remain connected for the entire duration of the job. The job-management GUI allows the users to see the job status, abort jobs, and download results. Once results have been downloaded, a post-processing GUI facilitates visualisation of both the 3D flow results and the compartment system results.

User access to the grid is controlled through the PKI infrastructure. However, in order to deal with the situation where a GEMSS account is compromised, additional checks of the application input files are performed on the GEMSS server. All executable code is pre-installed on the grid server, and so the user is denied the possibility of running arbitrary code of their choosing on the server.

For typical use of this service, compute times of up to one week would be considered acceptable. For the consultancy use of this software, computer resources would only be required occasionally, and therefore it is not cost effective for the end user to own their own HPC resource. By having access to a Grid resource, the consultancy company would be able to run more computationally demanding simulations, allowing improved simulation results.

Preliminary Evaluation

The Grid-enabled cardiovascular application couples a 3D fluid dynamics solver to compartments that represent the remainder of the circulatory system. The detailed 3D model allows for detailed investigation of the flow and pressure distribution in a particular section of artery, whereas the compartment model allows global effects to be examined and influence the local 3D solution. The software allows the haemodynamic aspects of arterial disease to be investigated on both a local and a systemic level.

Although the formal evaluation phase for ST4.5 has not yet begun, certain tests and checks have been performed during the course of the software development. These include a comparison of compartment system results against published data, and an assessment of grid ease of use, speed and reliability.

This software has been developed in association with the end users represented within GEMSS (ASD and IDAC), and their comments have influenced the development process. The following issues have been highlighted:

- The software has been made as easy to install and operate as possible.
- The software has been written to allow easy addition of new types of coupled-compartment that may be required to solve novel problems.
- The software is compatible with MATLAB 5.3 – it does not require the very latest version.
- The solution step can be run locally as well as on the grid, using the same user interface. This allows approximate solutions to be obtained for a coarse mesh (and tested for convergence) before a computationally intensive high-accuracy grid run is submitted.

Our end users will test the software during the formal evaluation phase, and a questionnaire will be sent to them to assess the extent to which the software meets their requirements. This will highlight ways in which the software performance could be improved.

Status Summary

This summary outlines key features of the GEMSS Cardiovascular simulation service accomplished within release of deliverable D4.5.

- The cardiovascular simulation service provides coupled compartment tools for researchers or consultancy companies. These tools allow a 3D CFD model to be coupled to compartments which represent the arterial system both upstream and downstream of the model.
- The client interface was written in MatLab (as opposed to EASA).
- The application has been partitioned into local-client and Grid-server sections.
- The service complies with the GEMSS security infrastructure (X.509 compliant).
- The Cardiovascular client interface has been implemented using MATLAB.
- Enhanced security measures have been implemented on the Grid server to check incoming files for malicious code.
- The Grid enabled application fulfils the requirements specification of D1.1.
- A MatLab licence is not required for the Grid server (although this is contrary to the original design specification, it resolves MatLab licensing issues)
- Documentation is provided with the software release and includes a tutorial guide and supplemental technical information.

Future Goals

Evaluation of solutions obtained using the compartment model system for:

- solution stability
- solution accuracy

Closer integration with the most recent middleware release (Phase 3), providing:

- solution time estimation (QoS feature – performance model)
- improved job scheduling (QoS feature - scheduling)
- improved choice of compute servers (QoS feature - negotiation)
- integration of a well defined and transparent business model

Conclusion

A Grid-enabled Cardiovascular application is now operating successfully on the GEMSS Grid and has been used by end-users at USFD and within GEMSS. It is apparent that large problems (i.e. ones that exceed the computational capacity of small business PC networks) can be solved securely through the computing resources provided by GEMSS. The accuracy of the simulation results is a strong function of computation time and resources, and the Grid offers a significant advantage in this respect. The latter assertion can only be properly determined after a period of evaluation, the desirable outcome being confidence in the ability of the Grid to deliver effective computing and the ability of the software to deliver effective and accessible cardiovascular simulation. Now that the software has been released it will be promoted through dissemination activities, highlighting the potential of the Grid for solving these types of problem.

3.4.6 Sub-task 4.6: Advanced Image Reconstruction

Application Scenario:
Tumour diagnosis and monitoring of metabolism are the main tasks of in vivo diagnosis in nuclear medicine by visualization of distribution of radioactive tracer in the human body. Although SPECT reconstruction suffers from low spatial resolution and poor signal-to-noise ratio compared to modern x-ray CT and MRI, it provides complimentary functional information, and is indispensable in modern clinical diagnosis. The diagnostic procedure requires the patient to receive an ionising radiation dose which provides the radiation necessary for acquisition of multiple projections by the gamma camera. The projection data is computationally reconstructed for subsequent reporting and diagnosis. A large variety of reconstruction algorithms exists, many of them based on standard filtered back-projection (FBP). This method finds extensive use in current clinical practice but it is only applicable to single slices. The modern iterative algorithms available within GEMSS offer benefits because they encompass technical and physical constraints of the imaging process and are easily extendable to 3D, but this comes at the expense of high computational effort. The Grid is well suited to the task in addition to which an implementation of image reconstruction as a Grid service could also bring access to highly sophisticated image processing resources for better diagnosis.

Workflow:
Diagnostic SPECT images are reconstructed from projection data, (ie. the sum of emitted photons along a linear manifold). In practice this manifold is a cone-like sub-volume of the object, from which photons are recorded by surrounding detectors. A rectangular detector array is rotated around the patient and a series of projection data is acquired. Iterative reconstruction repeatedly modifies a postulated image matrix through comparison of pseudo-projection and measured projection data. By this method a succession of intermediate images is generated until a convergence criterion is fulfilled. The weighted contribution of each voxel of the image to a specific projection value permits accurate modelling of collimator geometry and photon scatter, and all weights for all projection values define the system matrix which characterises the system response. If N is the length of the image cube, then the number of calculations is of order N^6 for full 3D reconstruction compared to order N^4 with slice reconstruction.

Grid-enabling:

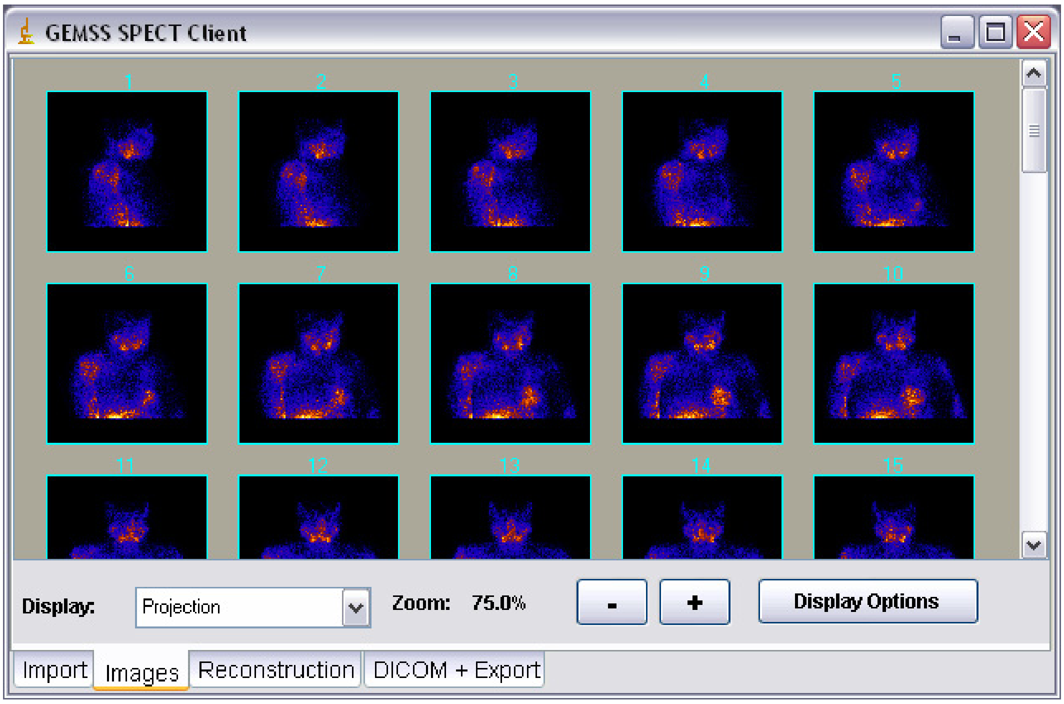
Reconstruction Kernel:
A state of the art algorithm – the OS-EM (Ordered Subsets – Maximum Likelihood) algorithm which is an advanced version of the well-known ML-EM (Maximum Likelihood – Expectation Maximization) algorithm of Shepp and Vardi – has been adopted to satisfy numerical and algorithmic constraints of fully 3D reconstruction and implemented. This algorithm is based on a stochastic model of Poisson distributed generation and detection of photons. Both, improved resolution and robust convergence criteria are characteristics of this algorithm.
The algorithm was implemented in ANSI-C employing a hybrid parallelization paradigm optimised for SMP-clusters. Symmetric shared memory parallelization was realised using

OpenMP directives. Inter-node communication was implemented using MPI from the mpich library.

Graphical User Interface (GUI):
The GUI serves as a portal to the GEMSS Grid. As such it relies on the high-level GEMSS client API and provides the following functionality:

- Visualization of projection/image data
- DICOM import/export
- Reconstruction parameters
- Quality of Service support
- Image processing functionality

The GUI can be launched as stand-alone application or as ImageJ plug-in. The latter enables the use of all image processing functionalities offered by ImageJ. That are not only those functions included in the basic version but also every plug-in implemented with ImageJ. As ImageJ can be easily extended by new plug-ins this provides a very powerful possibility for researchers in the field of medical image processing.



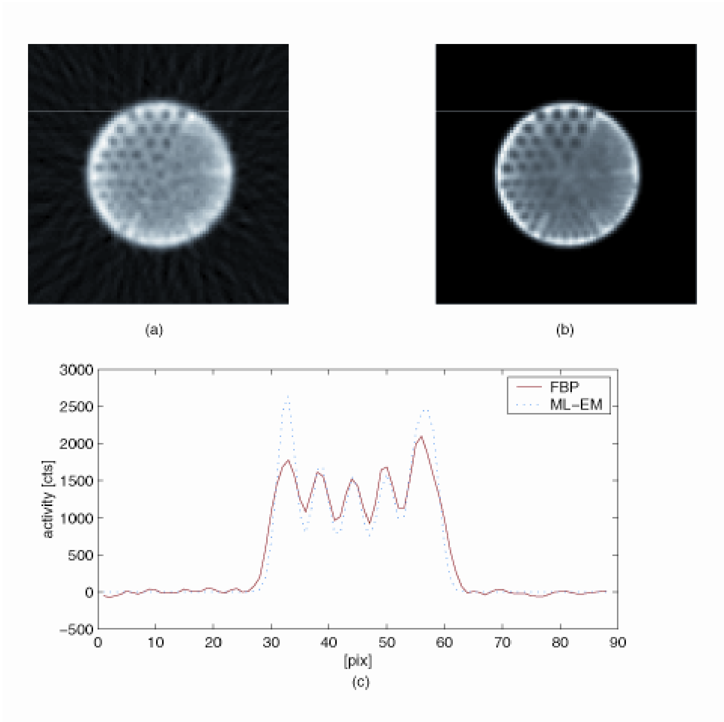
Screenshot of projection views provided by the GUI

DICOM Import/Export:
The Digital Imaging and Communications in Medicine (DICOM) standard was created by the National Electrical Manufacturers Association (NEMA) to aid the distribution and viewing of medical images. Generally spoken it addresses the exchange of digital information between medical imaging equipment and other systems.
Since data compatibility with imaging modalities of different vendors is a strong criterion for the wide-spread usage of the reconstruction services it is important to provide an easy-to-use

tool to import/export DICOM image and projection data. As a part of the DICOM standard the import/export of image and projection data has been implemented with the GUI.

Quality-of-Service, Performance Model:
As the actual image reconstruction takes place at some distant Grid site it is important to keep the end-users up-to-date about the progress of submitted jobs. The GUI provides information about job status (is the job still running or has it been disrupted?) and a progress bar visualizes the data provided by the GEMSS QoS module and the sophisticated performance model. This answers the question when the job will be finished.

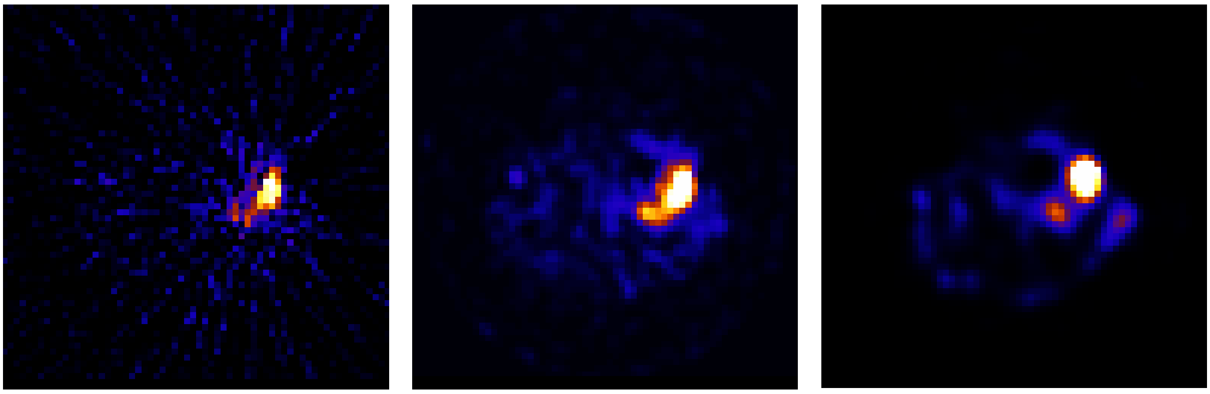
Test Case:
The first test case does not utilise patient data but uses a phantom to allow for quantitative evaluation of the accuracy of the 3D method. Projection data from a Jaszczak phantom was reconstructed and compared to filtered back-projection (FBP). The phantom contained 600MBq Tc99m and projection data was acquired on a GE Hawkeye SPECT scanner with a 128x128 projection matrix, 3 degrees rotational increment, 30 s acquisition time per step. Images were reconstructed using FBP with a Hanning window and compared with fully 3D OSEM reconstruction using an optimised system matrix with the alpha release software of the GEMSS project. From this phantom data the enhancement of contrast in both sections of the phantom, i.e. cold spheres and cold rods, could be shown.



Filtered Back Projection (FBP) (a) compared to OSEM-3D (b) and profiles (c)

The second test case compares reconstructions from an I^{131} patient study of the endocrine system (projection size 64x64, 60 projections, voxel size 8.84x8.84x8.84mm).
The test data was reconstructed using the GEMSS SPECT reconstruction service. The reconstruction was performed using the fully 3D OSEM (Ordered Subsets – Maximum Likelihood) algorithm with 4 subsets and 15 iterations. For comparison reconstructed images from standard 2D filtered back-projection (FBP) and iterative 2D reconstruction without attenuation correction (IRNC) are available. The following figure shows reconstructed images

arranged by reconstruction method. The 3D reconstruction algorithm clearly shows better results than traditional 2D reconstruction techniques.



(a) (b) (c)

Filtered Back Projection (a) compared to IRNC (b) and GEMSS OSEM-3D (c)

Status Summary:
This summary outlines key features of the GEMSS Vienna SPECT service accomplished within this release.
Within the GEMSS project, fully 3D image reconstruction was further developed from a stand-alone parallel application into a flexible Grid service, providing an easy to use, intuitive client front-end together with a fully transparent Grid-service part.

- The service complies with the GEMSS security infrastructure (X.509 compliant).
- The Grid enabled application fulfils the requirements of D1.1
- The architecture of the reconstruction service was designed based on detailed user requirements and specifications, reflecting strong feasibility criteria defined by clinical users, and the trust model developed upon this information.
- The architecture reflects transparent access to GEMSS Grid resources.
- On the user front-end of the service a GUI was implemented. It provides intuitive and easy usability of the complex reconstruction service.
- Basic image processing and analysis tools are implemented into the GUI.
- The GUI is implemented as stand-alone application and as plug-in to the ImageJ biomedical software-package. This implementation guarantees easy extensibility of the functionality.
- No additional software for visualization, e.g. Matlab, Analyze of reconstruction results is necessary, and thus no license fees have to be paid.
- Easy installation enforces great acceptance from the user community.
- The reconstruction service was implemented in the GEMSS test bed, i.e. the ISS cluster in Vienna, and the NEC cluster in St. Augustin, demonstrating the realization on different hardware and security configurations.
- A distinct security policy was implemented based on the GEMSS security model.
- Data compatibility to clinical SPECT systems was achieved by implementing the industrial DICOM standard for exchange of medical image data.
- Application of flexible cost models need robust estimation of prospective runtime and necessary system resources. Reliable QoS is a fundamental basis for implementation of cost models. An accurate QoS model was developed and tested in the GEMSS testbed.

Employing fully 3D reconstruction as a Grid service provides advanced methods of image processing to a broader medical community, benefiting from improved resolution and contrast, leading to increased sensitivity in nuclear medicine diagnosis.

Future Goals:

Within the GEMSS project validation of the reconstruction service will be done using the GEMSS test-bed with selected clinical end-users. A strong focus will be put on detailed testing of the final GEMSS middleware in the context of QoS negotiations and business models.

Further development of the GEMSS package could focus on integrated tools for real time monitoring of processes and work load distribution. Both would increase the reliability and administration of the software. Resource location tools and automated package installation tools would provide a further step towards automated administration of Software in the Grid space. For reliable use in a wider clinical context sophisticated error recovery strategies must be developed and integrated into the middleware.

Conclusion:

With the release of the GEMSS SPECT reconstruction service, fully 3D image reconstruction is accessible for a greater medical community. This meets one of the main EU R&D efforts for establishing and improving networks of experts on a European level, aided by modern IT infrastructure.

With GEMSS a framework for a secure middleware, especially addressing the issues of patient data privacy and legal issues of healthcare at an international level, was developed. Cost control is a major issue for most health services. In GEMSS billing procedures based on detailed Quality of Service models were integrated, which is a necessity for the acceptance of the reconstruction service in clinical practice.

All the application tasks within GEMSS demonstrate the development of Grid architectures from distributed data models towards workload sharing. This prospering field of Grid research is based on highly sophisticated Quality of Service models, providing accurate runtime prediction. Exactly defined performance models build the basis for a reliable end user scenario.

Beyond the scope of the GEMSS project, but as a further step towards more stability and reliability of the services, integrated tools for real time monitoring of the processes and work-load should be developed. Until now only basic failure redundancy strategies are implemented, further developments of the middleware would cover this topic. Valuable end user experience should be reported after further large scale clinical field studies.

3.5 Workpackage 5: Exploitation, Information Dissemination and Clustering

This workpackage is executing tasks related to exploitation and information dissemination. In particular it will plan and co-ordinate information dissemination activities among the partners, as well as produce a post-project exploitation roadmap. A second task will support project clustering activities along two axes: generic GRID technologies and technologies for applications in the health care domain.

3.5.1 Exploitation Planning

The GEMSS consortium has broad exploitation and use intentions for the project output, covering non-commercial and commercial exploitation. The details of the GEMSS exploitation plan, documented in a separate internal deliverable – Exploitation Planning Report, D5.1a, are confidential. D5.1a was prepared under Work Package 5 *Exploitation Planning and Information Dissemination* of the project, and complements the material in the eTIP (http://etip.cordis.lu) containing the Technology Implementation Plan. The D5.1a report sets out to describe planning of the exploitation of the results of the project. Particular attention is given to the markets in which GEMSS exploitation can take place, the opportunities within these markets, the barriers which would need to be overcome, and the planning required to exploit these opportunities. The dissemination activities necessary to complement the exploitation activities are described separately in the Dissemination and Use Plan, D5.2. The exploitation planning report firstly presents a very general description of the exploitation potential of the project. These general notes are followed by the description of domain specific activities, and the description of individual partners' specific plans, and how they can be integrated for maximum benefits. A draft Technology Implementation Plan has been implemented using the Commission´s online eTIP facility. A printed version of the public information has been submitted as deliverable D5.3a after the first project year. Currently the partners are finalizing the Full Technology Implementation Plan, D5.3b.

3.5.2 Dissemination Activities

GEMSS project partners observe current developments in the area of GEMSS activities through the attendance of conferences and events to ensure that the public will be informed about the work of GEMSS and that GEMSS will be aware of other projects´ activities. Relevant conferences and meetings including events organized by the European Commission are listed in the table below.

Name	Location	Date
Annual Congress “International Society for Rotary Blood Pumps”	ISRPB annual meeting	31.8.- 2.9.2003
Security Workshop: Federal Commission for Health Telematics	Namur, Belgium	2.9.2003
Conference ParCo2003	Dresden, Germany	3.9.2003
Annual Congress “European Society of Artificial Organs”	ESAO annual meeting	3.-6.9.2003
Annual Meeting of the Austrian, German and Swiss Society of Biomedical Engineering.	Salzburg, Austria	24.-27.9.03

IST2003	Milan, Italy	1.-4.10.03
One day introduction to EORTC trials.	Louvain-la-Neuve, Belgium	3.10.2003
« La sous-traitance de données à caractère personnel en support à l'octroi de soins de santé. L'exemple des applications GEMSS »	EORTC Brussels, Belgium	4.10.2003
Security Workshop: Federal Commission for Health Telematics	Namur, Belgium	14.10.2003
WebDay Conference	London, UK	14.10.03
European Respiratory Society Meeting	Vienna, Austria	Sep. 2003
21st CAD-FEM Users' Meeting 2003	Berlin, Germany	12-14.11.003
MEDICA2003, HealthGrid Workshop	Düsseldorf, Germany	19.11.03
21st FSI Advisory Group meeting	Dundee, UK	22.-25. Nov. 2003
e-Science security workshop. Edinburgh, UK	Edinburgh, UK	Nov 2003
AxGrids Conference & EC Grid Concertation Meeting	Nicosia, Cyprus	27.-30.1.04
The Future of Grid and Web Services	Imperial College, London, UK	28.1.04
HealthGRID 2004 Conference	Clermont-Ferrand, France	29.-30.1.04
IPEM Head & Neck cancer radiotherapy conference	London, UK	11 Feb 2004
Global Grid Forum (GGF10)	Berlin, Germany	9.-12. Mar 2004
Monte Carlo conference 'NPL Workshop on Monte Carlo Codes'	Teddington, UK	17/18 Mar 2004
Grid Technology Conference 'Grid Technology and your business: current benefits and future plans'	Leeds, UK	18 Mar 2004
12th International meeting of the Gamma knife society	Vienna, Austria	16-20 May 2004
Conference on Genetic Data	Louvain-la-Neuve, Belgium	4 & 12.3.2004
Seminar on Medical Informatics	Centre d´études namurois d´information médicale	13.3.2004
Séminaire d'actualité de Droit Médical, "Le respect du corps humain pendant la vie et après la mort, Droit éthique et culture".	Toulouse, France	6 & 7.5.2004
Certificat inter-universitaire de traitement de l'information hospitalière	UCL-ULB Namur, Belgium	14.5.2004
UNICOM Conference Web Services and The Grid	London, UK	24.-25.5.2004
Workshop on Grid Applications Interfaces	eScience Centre Edinburgh, UK	19.-21.7.2004
EC Concertation Meeting	Brussels, Belgium	17.9.2004
Global Grid Forum (GGF12)	Brussels, Belgium	20.-22.9.2004

- **Contacts with industry**
  GEMSS has established a strong focus on industrial dissemination through exhibition stands, user conferences, industrial presentations, and discussions direct with potential users.

  GEMSS has links to Industry through contacts established in other projects and existing contacts especially by the industrial Consortium partners who also disseminate information about GEMSS related work to their (industrial) user communities. For example, IDAC have been talking to 4 medical device manufacturers in the US and a number in Germany and Ireland. We have also been collaborating with a company in Japan to help them deliver their applications successfully to their customers. IDAC has been closely collaborating with CADFEM in Munich to provide internet delivered vertical applications to CADFEM's clients using EASA and potential GEMSS technologies. IDAC has also been working closely with EASA to help promote the use of EASA for GRID computing in the area of medical device applications and more general areas. The presentations from the Medical Device Seminar held in Galway, Ireland on the 23rd of January 2002 were published on the IDAC website http://www.idacireland.com/seminars/presentations.php4.

- **Collaboration with other projects**
  GEMSS participates in the HEALTHGRID cluster formed by EU projects in the e-Health / Grid area. NEC became a member of the HEALTHGRID association.

  GEMSS has established and maintained its link with the GRIDSTART project. Partner IT Innovation is a contractor of the GRIDSTART IST project. GEMSS partners chaired two GRIDSTART Technical Working Groups and took an active role.

  In particular, GEMSS kept a close link to the GridLab project exchanging information, sharing experience and providing use cases.

- **Information sources**

  - **Websites**

    The public GEMSS website http://www.gemss.de was designed according to the EC DG Information Society guidelines version 1.1. It provides general information about the project. The index page offers the following links: Home, Project Details, The Consortium, Reports and Presentations.

    Further material on the project is contained within a secure website, with partner access only. It provides publicity material which can be used by individual partners for their own dissemination activities.

- **Printings**

  GEMSS has so far released:

  - a Project Handbook available to IST projects,
  - a Project fact Sheet,
  - a general brochure detailing the Project,
  - general Project posters,
  - application specific posters and handouts,
  - referenced publications in scientific journals,
  - conference proceedings,
  - press releases etc. (IST Results, PraxisComputer supplement to Deutsches Ärzteblatt, Computing No 40715),
  - CD ROMs with project and application-specific informations.
  - other media articles (partner presentation, application brochure) prepared on a by-request basis.

3.5.3 Clustering

GEMSS participates in the HEALTHGRID cluster formed by EU projects in the e-Health / Grid area. Partner NEC is an official member of the HEALTHGRID association. GEMSS participates in the annual HEALTHGRID conferences contributing numerous presentations and posters. In addition the project contributed to a white paper on Grid for Health.

GEMSS has extended its active links with the GRIDSTART project. Partner IT Innovation is a contractor of the GRIDSTART IST project. Active support was give to various GRIDSTART activities such as EC Concertation, Technical Working Group meetings and Grid workshops.

3.6 Workpackage 6: Project Management

This workpackage co-ordinates day-to-day running of the project and maintains all correspondence with the European Commission. The main goal of WP6 is to ensure the project's success by monitoring and reporting progress against goals and time scales, by establishing processes to ensure quality in project work and by anticipating and managing risk and change to the project.

3.6.1 Project Communication

The communication strategy aims to keep all the Partners fully informed about the Project status, the planning and all other issues which are important to the Partners in order to obtain maximum transparency for all involved and to increase the synergy of the co-operation. Interactive management meetings and technical meetings take an important role in the communication strategy:

- All information (like minutes of meetings, visit reports, tasks reports, relevant publications etc.) is communicated to the Project Co-ordinator, who is responsible for also channelling this information to the Partners, where appropriate.
- BASIC SUPPORT FOR COLLABORATIVE WORK (BSCW): A BSCW server has been set-up as a transparent shared work-space at the co-ordinators site. The server complements the web-server and contains all project internal documents. In addition, it serves as a common repository for documents used in preparation of deliverables. Currently 42 users from all partners are registered.
- Email reflectors have been set-up at NEC for the following (sub)groups:

All partners:		gemss@gemss.de
PMB:		pmb@gemss.de
WP1:		wp1@gemss.de
	ST1.1:	st11@gemss.de
	ST1.2:	st12@gemss.de
	ST1.3:	st13@gemss.de
WP2:		wp2@gemss.de
	ST2.1:	st21@gemss.de
	ST2.2:	st22@gemss.de
WP3:		wp3@gemss.de
	ST3.1:	st31@gemss.de
	ST3.2:	st32@gemss.de
	ST3.3:	st33@gemss.de
WP4:		wp4@gemss.de
	ST4.1:	st41@gemss.de
	ST4.2:	st42@gemss.de
	ST4.3:	st43@gemss.de
	ST4.4:	st44@gemss.de
	ST4.5:	st45@gemss.de
	ST4.6:	st46@gemss.de
WP5:		wp5@gemss.de

Bug report and support request tracking:

To track bug reports and support requests and automate their processing as much as possible a tracking system has been installed (roundup from http://roundup.sourceforge.net). Roundup is a simple-to-use and easy-to-install issue-tracking system with web, e-mail and command-line interfaces. It automatically keeps a full history of changes to issues with configurable verbosity and easy access to information about who created or last modified *any* item in the database. The system supports automatic notification and file attachments added through the web or email interfaces.

3.6.2 Management Overview

The project management has acted as contact point for all correspondence between the project and the Commission. Project-internal communication has been simplified by the provision of GEMSS mailing lists for each workpackage or subtask. All correspondence via the GEMSS lists is automatically archived. During the second year of the project, the Consortium Agreement (modified Unified Consortium Agreement) has been signed by all partners. The pending Contract Amendment (AEA became CFX Ltd.) has been finalized. The second cost statement copvering the period from March 2003 to February 2004 has been submitted in time and processed by the corresponding unit of the European Commission.

Three full project meetings took place in the reporting period. Project Management Board (PMB) meetings were held during the two-day meetings. The project meetings were used to define and review workplan implementation details and collaborations and also to address exploitation possibilities. The regular meetings were accompanied by several technical sub-group meetings as required by the project.