Grant agreement no.211714

neuGrid

A GRID-BASED e-INFRASTRUCTURE FOR DATA ARCHIVING/ COMMUNICATION AND COMPUTATIONALLY INTENSIVE APPLICATIONS IN THE MEDICAL SCIENCES

Combination of Collaborative Project and Coordination and Support Action

Objective INFRA-2007-1.2.2 - Deployment of e-Infrastructures for scientific communities

Deliverable reference number and title: **D6.1a Distributed Medical Services Provision (Management Summary)**

Due date of deliverable: **Month 12**

Actual submission date: **31st January 2009**

Start date of project: **February 1st 2008** Duration: **36 months**

Organisation name of lead contractor for this deliverable: **University of the West of England, Bristol UK**

Revision: Version **1**

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Intended Recipients

The WP6 workpackage entitled **"Distributed Medical Services Provision"** aims to design a group of *generic* services that can be used in a number of related medical applications. These will then be implemented in order to fulfil the neuGRID specific project requirements. The services will be built according to the design philosophy presented in the WP6 deliverable in order to enhance and promote their re-usability in other related applications.

This document presents an executive summary of the work that has been carried out in year one, a number of recommendations that have emerged from the WP6 study and a set of objectives for year two in WP6.

This document is intended for project managers and EC reviewers and may be revisited as and when key recommendations presented in the document evolve due to the ongoing research and development process in the workpackage. To a lesser extent, since indirectly concerned (the recommendation and future plans in the workpackage may impact the decisions made on the potential use and exploitation of the project outcomes), neuro-scientists and prospective users (e.g. Pharmaceutical industries) as well as internal and external users of the project activities, are also anticipated as potential readers of this summary.

1 Executive Summary

The aim of the neuGrid project is to provide a user-friendly grid-based e-infrastructure, which will enable the European neuroscience community to carry out research that is necessary for the study of degenerative brain diseases. The WP6 work package, Distributed Medical Services Provision, will supply a range of services to the users of the project. The creation of a group of generic medical services will enable Grid technologies to be applied in this and a number of other medical domains. This will provide the flexibility that is necessary for interfacing with existing medical systems and will allow the reuse of packaged services that exploit Grid functionality. This work package will gradually provide an Application Programming Interface (API), which is independent from both the application domain, the LORIS clinical data management tool that will be used in neuGrid and the underlying Grid infrastructure.

By abstracting Grid middleware specific considerations and customizations from clinical research applications, WP6 services should be able to provide generic functionality aimed specifically at medical applications. In the scope of this project, these generic services will be applied to satisfy the specific requirements of neuroscientists. They will bring together sources of data and computing elements into a single view as far as applications are concerned, making it possible to cope with centralised, distributed or hybrid data and provide native support for common medical file formats. Lower-level services will hide the peculiarities of a specific Grid technology from upper layers, thereby providing application independence and enabling the selection of 'fit-for-purpose' infrastructures. These *generic* services in

WP6 will glue a wide-range of user applications to the available Grid platforms thereby creating a foundation of cross-community and cross-platform services.

This work package will consequently identify services that can be made reusable for the medical domain. The output of WP6 will consist of a set of generic medical services which will include but will not be limited to query, workflow, provenance, glueing and abstraction, anonymization and portal services. Other services that are identified by the user requirements will be added where necessary. These services will enable the medical community and Grid community to work cohesively whilst maintaining their independence and generic behaviour.

The WP6 deliverable document outlines the design philosophy that is being followed during the construction of the distributed medical services and describes a design for the set of services that will constitute the distributed medical services layer. The design and evaluation process was led by user requirements, which have been separately elicited in WP9. The services have been designed by following a set of design principles whose features include:

- Services will be designed and built upon a service-oriented architecture.
- Services will follow a loosely coupled paradigm to reduce dependencies between components and services.
- Services will be made open and extensible so that new features and components can be added where necessary.
- Services will be exposed through interoperable interfaces and standard specifications will be employed and implemented where appropriate.
- Services will not be tied to a particular technology and will be middleware agnostic.
- Service reusability will be ensured to make them reusable and customizable across applications and domains.
- Services will be made as scalable as is architecturally possible to elegantly manage application and user loads.

Since requirements analysis is an evolving process and requirements may emerge as new functionality becomes available, the services will adapt to address the changes that are necessary and will implement the functionality as and when requested by users. This extension and evolution mechanism has been established in the design philosophy and services can be seamlessly extended whenever a new feature is required. The requirements helped us to identify the following set of services that are necessary in the generic distributed services layer. The services will be designed in such a way that a variety of applications and Grid middleware can be supported.

A short summary of services that have been currently identified and introduced in the design document is as follows:

- **Pipeline Service:** In neuGrid, the core functionality of the Pipeline Service includes: 1) enabling the authoring of pipelines in a user friendly environment, using the Prodema executables as actors in the pipelines; 2) Parallelizing and Grid-enabling the abstract user defined pipeline for optimal execution over a grid; 3) Submitting and enacting the pipeline for execution on the Grid; 4) Viewing results of the execution as well as intermediary provenance data.
- **Glueing Service:** The Glueing Service is a generic service in WP6, which aims to provide: 1) a standard way of accessing Grid services without services

and applications being tied to a particular Grid middleware; 2) a mechanism to access any deployed Grid middleware through an easy-to-use service; 3) a solution which extends and enhances the reusability of already developed services across domains and applications; 4) a service-based approach to shield users and applications from writing complex Grid specific functionality; 5) a simplified approach for enabling clients to Gridify their applications without installing and maintaining too many Grid specific libraries.

- **Provenance service:** The requirements analysis process has clearly identified the fact that keeping track of how results are produced is important to users. In response to these demands, the Provenance service will provide a means of capturing and maintaining workflow specification and execution information in a workflow/provenance database. The Provenance service will provide recording and querying interfaces to store pipeline information and retrieve previously stored provenance information.
- **Querying Service:** The querying service has been designed to: 1) create a platform which can query disparate data resources; 2) implement a solution which is platform independent and service oriented; 3) provide a mechanism by which queries are enriched and that facilitates users in getting meaningful results quickly and efficiently.
- **Anonymization Service:** Privacy of patients is one of the key requirements that the neuGrid project has to address. If researchers can use the system to share widely their patients' data in confidence, with appropriate care taken over the privacy of their data, this will provide necessary assistance in the adoption and the usage of such a system. By providing different levels of pseudonymization and face scrambling which can take place at different levels of the data acquisition process, a good degree of privacy protection will be achieved through a generic Anonymization Service.
- **Portal Service:** The Portal Service has been designed to provide a single point of entry for the users to access the neuGrid services. It will hide the complexity of the underlying low-level neuGrid architecture from the users and will allow them to focus on making best use of the functionality that is provided. It will allow the community of users easily to securely identify themselves, access the services, browse data, launch analyses and visualize results.

In the design document, an attempt has been made to produce designs of these services that can best address the user requirements (as identified in work package WP9) and at the same time are consistent with the design principles that were summarized earlier. The following sets of activities were performed in order to produce the services designs:

- An analysis of the requirements was carried out to identify the components that can best address the user requirements. A group of generic components that can address these and other requirements that may emerge in future was created.
- The components were identified that can provide common functionality or have overlaps in the functionality they offer. All such components were

bundled together to produce a group of components that can address an area of the requirements. The grouped components were exposed as the candidate services that have been described in the above sections.

- A service design was produced for each of the services using the defined design philosophy. This design process not only glued various components together, it also ensured that they can be extended when required in the future. It was ensured that the minimum dependencies existed between components and services and they were sufficiently scalable to cope with future demands in loads. Particular care was taken to make these services as generic as possible and vendor and middleware lock-ins were avoided.
- Suitable interfaces were crafted to provide access to these services. It was ensured that these interfaces promoted interoperability and ease-of-use. Standard approaches were employed in designing these interfaces with the intention that services should follow generally agreed standards to the greatest possible extent in order to make them widely exploitable.
- An analysis and evaluation process was initiated to investigate emerging technologies and tools that could help in producing the functionality as required in the services. This process helped in identifying a set of technologies that could be reused in implementing the services. This activity also produced a list of functionality in each of the services that could not be implemented using existing technologies and where new development would have to be carried out. This process also produced a comparative analysis of the state of the art in each of the services and identified missing functionality.
- The WP6 design has delivered a set of services as presented in the document. It has also considered and described how state of the art technologies can implement these services and considered the advantages and disadvantages of each. Recommendations are made regarding the use of interfaces and suitable technologies. Any missing functionality is described and a roadmap for implementing and delivering the services that have been identified has also been set out.
- Each service describes the areas in which more research is required and states what can be quickly integrated to release an initial set of functionality in each of these services. An effort will be made to deliver initial functionality in each of these services in year two and further research will be carried out in the following years to add additional functionality.

The first version of the WP6 document follows the requirements analysis process, while delivering early insights on possible solutions. It is anticipated that the document will have a series of refinement iterations in light of the requirements analysis delivery and throughout the neuGrid platform development.

2 Recommendations

This section provides recommendations for WP6 services based on the activities that have been carried out during the first year of the project and that have culminated in the production of the deliverable. This section also briefly recalls the work package objectives, how these have been addressed and the outcomes that have been generated thus far. It is felt that a good foundation has been laid, which will provide a stable basis for future efforts in WP6. The aims of WP6 are:

- To produce a set of generic medical services to sit between the user-facing services that are being developed in WP5 and the Grid Services being provided by WP7.
- To provide independence from specific Grid-technologies.
- To provide a flexible and re-usable set of medical services that follow the SHARE recommendations and that can be extendable to other medical domains in the future.

In order to achieve these objectives, the requirements process helped in identifying candidate services and functionality that these services should provide for users of the system. During this activity we also produced a design philosophy to drive the service design process. The design philosophy delivered a set of guidelines that services should follow to ensure their execution and eventual composition into biomedical applications. Some recommendations for the design of WP6 services are given as follows:

- The services will be designed and delivered following a Service Oriented Architecture approach.
- The services will be flexible and able to be composed into adaptable workflows; they will also be dynamically discoverable and component dependencies will be minimized.
- An attempt will be made to make the services loosely coupled at the specification and implementation level.
- It will be ensured that the service interface remains the same for the lifetime of the project. Generic interfaces will be defined that can be extended if new features are required.
- A WSDL description for each of the WP6 services will be created. This will describe a service, its endpoints and operations and its interfaces and features. The service roles and behaviour will be mapped to a number of WSDL documents. WSDL will be a basis of communication between different services as well as with clients.
- The service contracts and implementations will be abstracted from users and only a minimal set of information regarding them will be exposed.
- The services in WP6 will be made open and extensible. This may include decoupling the modules for improved reuse, decoupling the infrastructure for better interoperability and decoupling the clients for better evolution.
- The design and implementation of the services will be protocol independent. A means will be provided for describing and accessing the services in a generic way. The services will be interoperable. For the scope of the project, the SOAP protocol will be used to achieve service interoperability.

Mechanisms will be put in place for code reuse and plug and play communication through any protocol during the service deployment.

- Middleware agnostic medical services will be designed and implemented in WP6. The SAGA API, implementations and adapters will be exploited to provide the desired abstraction. The services will be made reusable across domains and platforms; they will also be made language and platform independent by adopting standard protocols;
- The services Black box approach will be preferred over the White box approach of reusability. The service interfaces will be designed to achieve reusability and will be made customizable for use in a range of medical applications through different configurations. The services will also be adaptive and will have access to a wide range of middleware at run-time
- The services will be made scalable and light-weight by creating multiple instances of the services that may communicate with each other. They will be made load-aware by using load balancing techniques. The services will be made stateless and idempotent to achieve scalability. Reliability will be ensured by the use of service intermediaries and by adding redundancies. The services will be made performance aware with the help of caching strategies and efficient grouping of the messages.

Once the project requirements had been discussed, it was necessary to map these against possible components and identify those that should be delivered. In the light of design philosophy, such components will act like services. The reasons for building these components as services have been presented in detail in the design philosophy. Services will exist as autonomous and loosely coupled entities that can be executed independently. Each will address requirements that cannot be handled by any other single component or service. Collectively and in cooperation with each other, services will support the user analysis process and therefore deliver a fully functional neuGrid system.

A group of services was identified that was neither middleware nor application specific. These generic services can be used by any application and should run on any Grid middleware. We have discussed the major components and requirements that each of these generic services should address. After introducing and describing the services a more detailed design for each of these services was presented. This included attention being paid to architectural considerations and the selection of the most suitable structure for each of the services, which addresses the major user/system requirements. The services design also considered the technological choices which are available and justified why specific technologies have been selected, where appropriate. The design includes individual service components, API's and interfaces that will be provided to enable interaction with other services and applications (The services designs specifications will be delivered in the next couple of weeks). Each of these services will have an implementation road map that closely aligns with project commitments and deliverable submission deadlines.

As discussed in the design document, user requirements may evolve and new sets of features may need to be added. Consequently, service designs and the functionality offered may also evolve and provision must be made in the designs to address potential future changes in functionality. The designs will therefore be periodically reviewed and any changes and future suggestions and recommendations will be considered. It is anticipated that the document will have a series of refinement

iterations in the light of the requirements analysis delivery and throughout the neuGrid platform development.

3 Future Work Plan

The first year of the project concentrated on eliciting the user requirements and mapping user requirements to specific services. The services design philosophy was defined in order to make the neuGrid services scalable, re-usable, extensible and compliant to generally agreed standards. The services have also been designed to be middleware agnostic and loosely coupled. The service designs have been guided by the user requirements and the WP6 design philosophy. The designs have been created after extensive evaluations of relevant state of the art technologies. The second year activities include the following:

I. Detailed design specifications and definitions of the APIs of the individual services will be created. The API specifications will define the functionality that each service will provide. The services designs and their API specifications will facilitate the creation of an interoperable services infrastructure.

II. Detailed plans for implementing the services according to the design philosophy and user requirements will be created. The functionalities will be identified and prioritised in each of the services that are really required to address the user requirements.

III. The design specification and the architecture that is proposed will be discussed with the potential user communities. The intended end-users will verify the functionality of each of the services, choice of the technologies that have been made to address the requirements and immediate outcome of the proposed implementation plans. After end-user agreement, initial prototypes of the services will be developed. These prototypes will have the basic functionality that is mandatory to address the essential user requirements.

IV. The user and developer guides, as well as an implementation report for each of the services will be compiled and presented to the users for their feedback and evaluations. A quality assessment mechanism will also be put in place to ensure that services are designed and delivered as per the design philosophy and meet the user expectations.

V. The prototype implementations will be designed to be feature complete, however they will not be the final deliverable services. The end users will

evaluate prototype services, and their feedback will be used to generate more accurate service design specifications. The feedback gained from the end-users will be used to create more enhanced services which will be the focus of future activities in the project. The future activities will also include steps to optimize the service implementations and their benchmarking specifications will also be produced. Service standardization measures will also be introduced to make them interoperable and widely exploitable across communities and application domains.