



IST-2004-004475

DataMiningGrid

Data Mining Tools and Services for Grid Computing Environments

Specific Targeted Research or Innovation Project

2.3.2.8 Grid-based Systems for Complex Problems Solving

D11(2): Common Requirements Analysis, Specification and Evaluation of DataMiningGrid Interfaces and Services

Due date of deliverable: M9 (31. May 2005)

Actual submission delivery date: 10 July 2005

Start date of project: 1 September 2004

Duration: 24 months

University of Ulster

Revision: Submitted

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
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)	

The logo for Data Mining Grid, featuring the word "grid" in a stylized, blocky font where the letters are interconnected. The background of the entire page is a light orange color with a white grid pattern that curves across the space. There are several red squares of varying sizes scattered throughout the design, some appearing to be part of the grid or data points. The word "DATAMINING" is written in a bold, black, sans-serif font at the top of the page.

DATAMINING

**Deliverable D11(2):
Common Requirements
Analysis, Specification
and Evaluation of
DataMiningGrid
Interfaces and Services**

DATA MINING TOOLS AND SERVICES FOR GRID COMPUTING ENVIRONMENTS

Deliverable D11(2): Common Requirements
Analysis, Specification and Evaluation of
DataMiningGrid Interfaces and Services

Responsible author(s):

Vlado Stankovski, Jernej Trnkoczy

Co-author(s):

Markus Ackermann, Nataša Atanasova, Werner Dubitzky, Jürgen Franke, Thomas Hunniford, Jörg Kindermann, Boris Kompore, Nahum Korda, Valentin Kravtsov, Michael May, Nataša Fidler Mis, Thomas Niessen, Gerhard Paaß, Matthias Röhm, Assaf Schuster, Martin Swain

Revision history

Deliverable administration and summary	
Project acronym: DataMiningGrid	ID: IST-2004-004475
Document identifier:	DataMiningGrid-del-D11(2)-s-v10
Leading Partner: LJU in collaboration with all Partners	
Report version: 10	
Report preparation date: 31 May 2005	
Classification: Public	
Nature: Report	
Author(s) and contributors: Vlado Stankovski, Jernej Trnkoczy (LJU) in collaboration with all Partners	
Status:	Plan
	Draft
	Working
	Final
X	Submitted
	Approved

The DataMiningGrid © Consortium has addressed all comments received, making changes as necessary. Changes to this document are detailed in the change log table below.

Date	Edited by	Status	Changes made
-	DoW	draft	"a report template will be defined that all reports will follow"
01.04.2005	vs (2 versions)	working	D11 (1) was taken as basis for this document. The goal is to update D11(2).
14.05.2005	tn (2 versions)	working	FHG contribution in collaboration with UU. Amendments to the preliminary requirements.
20.05.2005	ms (2 versions)	Working	Final changes from UU.
31.05.2005	jt, vs (2 x)	Final	Editing and final deliverable.
05.07.2005	wd	Submitted	Minor changes, final checks and editing

Note that other documents may supersede this document. A list of newest public DataMiningGrid deliverables can be found at the <http://www.DataMiningGrid.org/deliverables.htm>.

Copyright

This report is © DataMiningGrid Consortium 2004. Its duplication is allowed only in the integral form for anyone's personal use for the purposes of research or education. Citation should always be provided.

Citation

Vlado Stankovski, Jernej Trnkoczy (2004) et al. Deliverable D11(2). DataMiningGrid© Consortium, University of Ljubljana, www.DataMiningGrid.org

Acknowledgements

The work presented in this document has been conducted in the context of the project **IST 2004 004475 DataMiningGrid**. DataMiningGrid is a 24-month project that started on September 1st, 2004 and is funded by the European Commission as well as by the industrial Partners. Their support is appreciated.

The Partners in the project are University of Ulster (UU), Fraunhofer Institute for Autonomous Intelligent Systems (FHG), DaimlerChrysler (DC), Israel Institute of Technology (TECHNION) and University of Ljubljana (LJU). The content of this document is the result of extensive discussions within the DataMiningGrid© Consortium as a whole. This report owes to a collaborative effort of the above organizations.

More information

Public DataMiningGrid reports are available through DataMiningGrid public web site (www.DataMiningGrid.org).

Executive summary

This document specifies the technical requirements and the goals for the development of DataMiningGrid tools and services. It also specifies possible usage scenarios with the purpose of establishing more detailed technical requirements¹. The individual parts of Deliverable D11(2) are collected in a single, coherent requirements document that will be a major input for the system and component design.

The Partners identified and elaborated DataMiningGrid requirements in the following areas:

- Identifying (locating) DataMiningGrid resources by using metadata,
- Accessing and selecting subsets of data,
- Data transfer,
- Data (pre-) processing,
- Data mining tasks,
- Text mining and ontology learning,
- Workflow editing and submission,
- Data privacy, security and governance,
- Integration of domain knowledge,
- Grid infrastructure and middleware functionality,
- Usability, response times and user-friendliness.

All individual tasks relating to requirements analysis and specification have contributed to this joint deliverable; therefore, D11(2) is the product of the following tasks:

- Task T12: Common requirements specification;
- Task T21: Requirements analysis for data access, transfer, and manipulation and
- Tasks T61-T64, that is, demonstration tasks contributed by all Partners.

D11(2) has been composed from contributions by all Partners.

¹ End-user specific requirements will be elaborated in Deliverable D61.

Table of contents

Executive summary	6
Table of contents	7
1 Introduction	9
1.1 Grid computing and data grid systems	9
1.2 State of the art in grid computing.....	10
1.3 Research and development challenges	11
1.4 DataMiningGrid underlying philosophy.....	12
2 Representative use cases	14
2.1 Methodology	14
2.2 Use Cases Description	14
3 Requirements.....	17
3.1 Identifying (locating) DataMiningGrid resources by using metadata	17
3.2 Requirements concerning data privacy, security and governance	19
3.3 Accessing and selecting data	20
3.4 Data transfer.....	22
3.5 Data pre-processing	24
3.6 Annotation of Data mining models with provenance information	26
3.7 Data mining tasks.....	27
3.8 Text mining & ontology learning.....	27
3.9 Workflow editing and submission.....	27
3.10 Requirements concerning integration of domain knowledge.....	28
3.11 Grid infrastructure and middleware requirements	28
3.12 Usability, response times and user-friendliness.....	30
4 Objectives – research goals beyond the scope of the project	31

5 Conclusions and future work..... 34

6 References..... 35

1 Introduction

The need for data mining in grid computing environments is illustrated by describing several representative use cases. Some of these use cases are based on efforts currently underway in industry and academia, others demonstrate long-term developments in business and industry. Section 3: Requirements presents a set of features that should be present and provides a motivation for those features. Section 4: Objectives – Research Goals Beyond the Scope of the DataMiningGrid Project describes a list of features that might be useful for many use cases, but may not necessarily be addressed by the DataMiningGrid project.

The detailed specification of the DataMiningGrid interfaces will take into account the following considerations:

- The requirements that are contained in this document,
- Review comments on this document from public feedback, invited experts and working group members,
- Specifications of (or proposals for) grid data mining tools and services that meet many of these requirements,
- DataMiningGrid requirements recognized by other projects from the concertation framework of the same strategic objective “Grids for Complex Problem Solving”, such as the SIMDAT, inteliGrid and others.

1.1 Grid computing and data grid systems

Grid computing could be viewed as a generic enabling technology for distributed computing. It is based on a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to computing resources anywhere and anytime. In their basic form, these resources provide raw compute power (CPU cycles) and massive storage capacity (magnetic disk or other mass storage devices) [Fos99]. These two grid dimensions were originally dubbed “computational grid” and “data grid”, respectively. However, since the inception of grid technology [Bax02], the term resource has evolved to cover a wide spectrum of concepts, including “physical resources (computation, communication, storage), informational resources (databases, archives, instruments), individuals (people and the expertise they represent), capabilities (software packages, brokering and scheduling services) and frameworks for access and control of these resources (OGSA - Open Grid Services Architecture, The Semantic Web)” [Fos02]. Using a grid to share resources, researchers and small enterprises can gain access to resources they cannot afford otherwise. Research institutes, on the other hand, can leverage their investment in research facilities by making them available to many more scientists.

Initially, the research community's focus was placed on computational grids. These have today reached maturity. Toolkits such as Globus [Globus04], UNICORE [Rom02], Condor [Condor04], and AVAKI [Avaki04] offer a wide range of services starting from job management, data transfer (for input and output), and various security primitives such as authorization and secure channels. Built on top Globus services, a number of data grid projects were initiated about three

years ago. The most notable of these is the EU-funded DataGrid project (initiated 2001) [DataGrid04].

Current data grid systems offer good solutions for file-based data access and data management problems: they allow a client to locate data, select data, take care of data replication, etc. However, many scientific and commercial applications are highly dependent on data stored in more complex database management systems, providing more sophisticated access and processing of data. Therefore, recent research has been focusing on (semantic) grid database access and integration services [GGF04], notably the DAIS standard (see below). These developments and the ever-increasing need to exploit the growing amounts of data across many sectors, give now rise to the development of generic grid infrastructure (protocols, services, systems, tools) facilitating the automated analysis and interpretation of large and inherently distributed data. However, distributed, grid-enabled environments imply new levels of complexity along conceptual, technical, legal, ethical, and other dimensions (see section 3 on research and development challenges).

1.2 State of the art in grid computing

In many of today's distributed computing applications, the idea of a *stateful* (as opposed to *stateless*) process is extremely important. Data values, and the results of operations on those values, must persist (i.e. maintain their informational state over certain periods of time). Given the inherent heterogeneity of modern platforms, maintaining state is a non-trivial concept, resulting in the development of standards in the area. The idea of web services has emerged to address heterogeneous distributed computing based on existing Internet-based standards (e.g. XML) [Fos04]. Based on this concept, the Open Grid Services Architecture (OGSA) [Fos02] was developed to provide a consistent manner for describing services in grid Environments. The Open Grid Services Infrastructure (OGSI) is one implementation of this architecture, which is now being reworked under the Web Services Resource Framework. Computational resources, storage resources, databases and programs can all be considered as services. Typically, the development of a standard in any computer application area is two-fold. It involves

1. The definition of service interfaces, and
2. The definition of protocols to invoke these interfaces.

In the area of data management the Data Access and Integration Services (DAIS) [GGF04] working group, as part of the Global Grid Forum, aims to develop and promote standards for accessing data across grids with an emphasis on existing systems. OGSA-DAI is a collaborative program developed with the aim of meeting the needs of the UK e-Science community by producing an open source database access and an integration middleware for grid applications. While it developed separately from DAIS, it aims to be fully compliant with the standards developed by them. As data access is of fundamental importance in data mining applications, the output from DAIS and OGSA-DAI is relevant to all data mining services that will be developed under the DataMiningGrid project.

Although the OGSi/OGSA standards have helped to provide grid-based services, several criticisms have been made with regards to its usability. Perhaps one of the most significant criticism is that it is too object-oriented and in violation of the “pure” web services concept (on which it was originally based). Pure web services are supposed to have no associated state or instances. As a result, the OGSi architecture is moving towards the Web Services Resource Framework (WSRF) [Czaj04]. This framework, initially inspired by OGSi, restructures OGSi concepts to fit more closely with the current web services architecture. Due to fundamental changes in the exchange of messages between these two approaches, WSRF-compliant services will not interoperate with OGSi services.

Much effort is made by both the scientific and business communities to develop standard grid middleware, allowing for easy development of grid-enabled applications and ensuring their interoperability. However, with the emergence of the OGSA standards, calling for an approach based on web services, there is much confusion and uncertainty among grid researchers and developers.

The Globus Toolkit 3.0, an OGSi based descendant of the widely adopted Globus Toolkit 2.x, required significant refactoring of already working grid environments. However, recent dramatic changes of OGSi towards further integration with Web Services (WSRF) render the previous work unusable. Such an unexpected change in the core of grid middleware makes it unclear how the latest developments may impact on other standards and the whole standardization process at GGF. Currently, the grid/WS interoperability standard is WS-I.

One of the tough challenges faced by any grid-related project today is to maintain a consistent “picture” of the latest developments and to comply with the constantly evolving standards. “Gridification” of data mining services will need to take onboard and adhere to existing and emerging grid data mining standards, such as those discussed in the next section.

1.3 Research and development challenges

As a result of the technological advances in recent years, fundamentally new, intrinsically distributed complex problem solving environments are emerging. These developments are prompting a range of new data mining research and development problems.

These can be classified into the following broad challenges:

- **Distributed data.** The data to be mined are increasingly stored in distributed computing environments on heterogeneous platforms. Consequently, development of algorithms, tools, and services is required that facilitate the mining of inherently distributed data within complex problem solving environments.
- **Distributed operations.** Emerging grid computing environments will see more and more data mining operations and algorithms be made available in grid environments. To facilitate a seamless integration of these resources into distributed data mining systems, novel algorithms, tools, grid services and other IT infrastructure need to be developed.

- **Massive data.** Development of algorithms and systems for mining large, massive and high-dimensional data sets (out-of-memory, parallel, and distributed algorithms) is needed, as more and more data mining problems in complex problem solving environments are faced with giga- and tera-scale data sets.
- **Data types.** The phenomena analyzed in complex problem solving scenarios are captured in increasingly complex data sources, structures, and types, including natural language text, images, time series, multi-relational and object data types. grid-enabled mining these data will require the development of new methodologies, algorithms, tools, and grid services.
- **Data privacy, and security.** Automated data mining within distributed computing data grids raises serious issues in terms of data privacy, security. Related issue involve ethical and legal aspects, good practices, and audits. Grid-based data mining technology will need to support researchers and analysts in using the data mining tools and services in a legal, ethical, secure, privacy-preserving, and auditable way.
- **Domain knowledge.** The data to be mined in complex problem solving scenarios will increasingly require the integration of existing domain knowledge into the mining process, particularly in knowledge-intensive domains. This knowledge is often dispersed across geographically distributed sites and organizations, either in digital form (ontologies, metadata, knowledge-based or simulation systems) or provided interactively by human experts. grid-enabled data mining technology will need to provide services and tools to support knowledge-aided data mining. (see also last bullet point).
- **User-friendliness.** Ultimately, data mining in a distributed grid computing environment must hide technological grid details from the user. To facilitate this, new software, tools, and infrastructure development is needed in the areas of grid-supported workflow management, resource identification, allocation, and scheduling, and user interfaces.
- **Resource identification and metadata.** A key characteristic of emerging grid-based complex data mining environments is that data sets, data mining operations and relevant domain knowledge resources will be dispersed around a potentially very large number of sites and nodes. Powerful metadata services resource identification tools and systems are required to navigate, select, and use the available resources effectively.

1.4 DataMiningGrid underlying philosophy

The underlying philosophy (approach) of the project is depicted in Figure 1. The Consortium decided to use the “three legged stool” methodology for project development process. This methodology basically consists of three main concepts:

- Use-case-driven process – all the stages in the process are driven by use cases,
- Architecture-centric – architecture has to be outlined at a very early stage of the project and it evolves over time,

- Iterative process of improvements of the various models (use case model, requirements model, analysis model, and design model). In the DataMiningGrid project we improve these models at least in three iterations.

The requirements are chosen based on the aspects of the use cases that the Consortium considered most important. As such, one should not assume that the DataMiningGrid components will directly support every aspect of each use case.

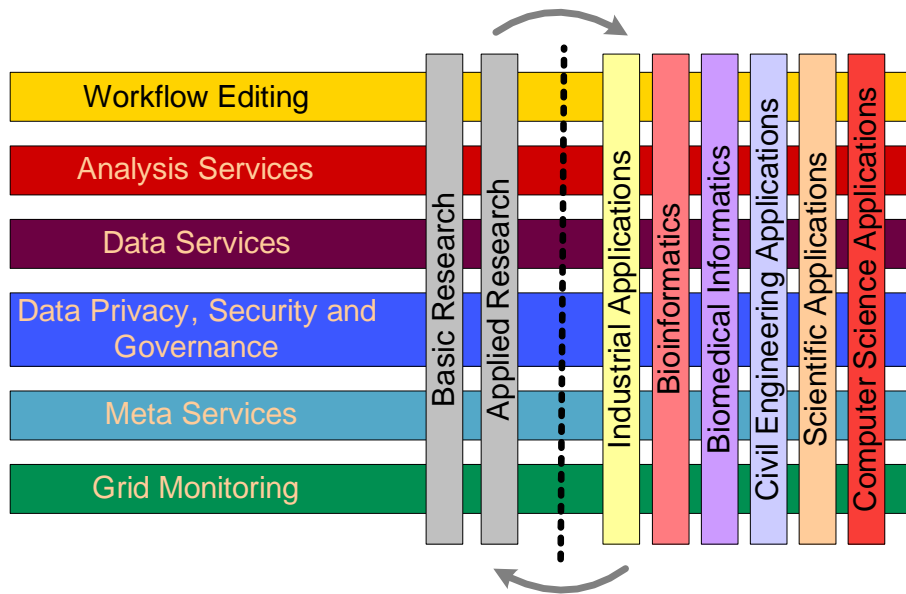


Figure 1: DataMiningGrid underlying philosophy

2 Representative use cases

2.1 Methodology

We are developing our grid-enabled data mining framework by following standard software development process. The first step in this process is requirements management, which is a systematic approach to finding, documenting, and managing requirements. The goal of the requirements management is to describe what the system should do and allows the developers and the customer to agree on that description. This stage is extremely important since bad requirements management is one of the main factors contributing to the fact that many projects end up missing user needs, are late or over budget.

In requirements management stage we elicit, organize, and document required functionality and constraints; track and document tradeoffs and decisions. In this stage actors are identified, representing the users, and any other systems that may interact with the system being developed. Use cases are identified, representing the behaviour of the system. Each use case (and the respective demonstrators) is described in detail in Deliverable D61(1). The notions of use case and scenarios proscribed in the process is an excellent way to capture functional requirements and to ensure that these drive the design, implementation and testing of software, making it more likely that the final DataMiningGrid interfaces, services and tools fulfill the end user needs. Managing use cases along with all other requirements is key to understanding the state of the project and better enables us to deliver the right technology.

The use-case description shows how the system interacts step by step with the actors and what the system does (this is part of Deliverable D61(1)). By providing a user's view of what the system should do, use cases provide functional requirements, which are subject to the present deliverable D11(2). Special requirements section of a use case, stores all non-functional requirements specifically related to a particular use case. Non-functional requirements about the DataMiningGrid framework as a whole are stored separately as supplementary requirements specification.

2.2 Use Cases Description

DataMiningGrid can be used to improve existing data mining techniques and may enable new uses of data mining in grid computing environments. The need for data mining in grid computing environments is motivated by several representative use cases from various scientific and business sectors (see Deliverable 61(1) for detailed description). Each functional requirement has been motivated by one or more use cases.

In this section we briefly describe some representative use cases. Note that this is not an exhaustive list, but instead a cross-section of interesting use cases.

These use cases serve as a guideline in choosing requirements for the DataMiningGrid tools, interfaces and services.

The representative use cases are:

- **The use of the genetic algorithms for gene regulatory reengineering.** Huge number of gene regulatory networks will need to be computationally simulated, and the genetic algorithms will be used to design and evolve these network simulations so that the optimum network configuration is discovered. Having access to large computational resources will be essential for this demonstrator, as the search space is vast, and the more simulations that can be run, the better will be the search through the possible genetic configurations. Each genetic network simulation can be run independently of the others. The simulations can be distributed over a grid, and the data produced by the simulations will need to be collected and processed by the genetic algorithm, which will then evolve new simulations based on the previous simulations. Finally, the optimum genetic network configuration will be presented to the user.
- **The prediction of protein structure and the precise understanding of protein folding and unfolding processes remains one of the greatest challenges in structural biology and bioinformatics.** Computer simulations based on molecular dynamics protein unfolding (MDPU) are at the forefront of the effort to gain a deeper understanding of these complex processes. Currently, these MDPU simulations generate a large amount of conformational data (typically in the range 100 MB to 10 GB) and are computationally expensive. More and more groups run such simulations and generate large data set, which raise new challenges in managing and analyzing these data. A data warehouse system that is embedded in a grid environment will enable the data to be adequately organized, managed, and analyzed. Data mining services that can run on or "near" the molecular dynamics unfolding data will play a pivotal role in its analysis. Such a system will ultimately allow research groups to share complex molecular dynamics simulation data worldwide.
- **Text mining use cases** including selection (similar documents), reduction (to parts of documents), organization (classification and clustering) and information extraction (for the purpose of ontology learning mechanisms) from a very large and distributed set of textual documents.
- **Grid monitoring use cases** which use data mining techniques to look for unknown patterns in the distributed log files and allow for better control over the behavior of the distributed system and its services. After submitting a grid job it is of critical importance to allow user to monitor its execution, as well as to constantly diagnose the system. Monitoring in distributed systems is extremely complex since it must combine information from numerous and diverse resources that simultaneously execute multiple computational tasks. This is why current distributed systems offer very little monitoring functionality. It is therefore necessary to provide a significantly more powerful monitoring functionality allowing ongoing analysis of the system's condition, and providing critical diagnostic information.

- **Data mining of distributed medical databases.** In this particular use case, endemic goitre and of supply Slovenian children with iodine is investigated. Currently, approximately four thousand children are being recruited in nine Slovenian regions and the data is collected in several regional databases. The collected data is being analyzed by statistical and data mining approaches. In this respect, the use case is neither concerned with massive data nor high-performance computing; however scalability is an open issue when mining larger (distributed) medical databases. It would be highly beneficial if the data is either not transferred from its original location or they could be transferred in a coded manner, thus preserving privacy and security, while at the same time facilitating medical studies on the data.
- **Ecological modeling use case** which will demonstrate the application of computationally intensive machine learning algorithms to find the best possible mathematical models of ecosystems, in particular population dynamics in aquatic ecosystems. We consider models of the dynamic change of population concentrations (or densities) that take the form of ordinary differential equations (ODE). There are two main aspects of building an ODE model of a real-world ecosystem. First, we have to establish an appropriate ODE structure (the structure identification task). Second, we have to identify acceptably accurate values for the constant parameters of the ODE model (the parameter estimation task). Each of these aspects will be implemented on the grid as a separate computational service. However, a basic service for simulating ODE models is also necessary, that will allow the user to simulate and compare behaviors of the different models.

3 Requirements

The use cases motivate a number of technical requirements for the development of DataMiningGrid tools and services. The DataMiningGrid Consortium currently feels that the **technical requirements** described below are **essential** to the DataMiningGrid© technology. Each requirement includes a short description and is motivated by one or more use case(s) from the previous section.

Functional requirements lists the set of operations that the DataMiningGrid services and tools must be able to perform in order for the services and tools to be considered functional. **Non-functional requirements** are additional criteria that generally focus on the system properties, such as environmental and implementation constraints, performance, platform dependencies, maintainability, extensibility, reliability etc. of the services and tools. Non-functional requirements are further divided into *special* (non-functional) requirements and *supplementary* (non-functional) requirements. The special requirements are related to the specific use cases and need to be handled in subsequent system models, such as the analysis, design and implementation models. The supplementary requirements are generic requirements and cannot be connected to a particular use case or a particular real world problem. They should instead be managed separately in a list of supplementary requirements.

The main classes of functionality required by users were identified as:

- Identifying (locating) DataMiningGrid resources by using metadata,
- Accessing and selecting subsets of data,
- Data transfer,
- Data (pre-) processing,
- Data mining tasks,
- Text mining and ontology learning,
- Workflow editing and submission,
- Data privacy, security and governance,
- Integration of domain knowledge,
- Grid infrastructure and middleware functionality,
- Usability, response times and user-friendliness.

3.1 Identifying (locating) DataMiningGrid resources by using metadata

Functional requirement 1: Provide mechanism for accessing metadata

Rationale: For the client to make decisions about how to handle the data provided by the data service, it is necessary for the client to access the metadata associated with the data. There are different forms of metadata. Application metadata describes properties of primary data that help the user to interpret it. Technical metadata describes how the primary data is stored in physical

resources. So in a biological example, the application metadata might describe biological variables like proteins, genes, metabolic networks, and so on, while the technical metadata would describe the hardware and software used to store this e.g. a Progres relational database version 7.3 on a 2.2 GHz PC running windows XP.

Initial Assessment: Application metadata would be most useful if provided the early stages of workflow design. This would suggest that application metadata should be part of the information supplied by the data service registry.

Technical metadata would mainly be used internally by the data services to construct data access and mediation workflows.

Motivation: Text mining use cases especially the DC use case for Subtask T61.3: Finding related and similar documents in the intranet. Document repositories on the intranet contain metadata describing the kind, format, up-to-datedness and accessibility of their data.

Functional requirement 2: Provide mechanism for locating data services

Rationale: Data services are distributed all around the world. Locating all the required data services using ad-hoc search techniques is unlikely to be successful. By providing an automatic mechanism for locating data services, appropriate data services will be found quickly and easily.

Initial Assessment: This requires a data discovery service. Data discovery services identify data items by matching specified characteristics, attributes, or metadata such as:

- A description, summary, or overview of the data source,
- Data provenance: how and where the data was generated i.e. what scientific instruments were used to collect the data; or what processing has been applied to the data; who is the creator, the owner, or the last modifier of the data,
- Physical metadata such as the data size, access control policies, transfer rates, error handling, number of disks, and the number of heads,
- It is also important to identify those data sources that are also available as computational resources so that data can be analyzed at its source.

Motivation: text mining use cases especially the DC use case for Subtask T61.3: Finding related and similar documents in the intranet. Identification and location of document servers/repositories on the intranet which can or have to be searched.

3.2 Requirements concerning data privacy, security and governance

Functional requirement 3: Authenticate request and provide appropriate feedback.

Rationale: The security of access to data and other resources may be critical for some applications. It will be necessary to authenticate requests and provide appropriate feedback. Not all services are public. For those that are not it is necessary to confirm client identity and to check client access rights. The actions that the client takes next will depend on knowing if the authentication was successful or not, so the client requires appropriate feedback.

Initial Assessment: Authentication can be provided via a simple username/password combination or more elaborate techniques can be employed. On the other hand, for public access sites, authentication may not be needed at all.

Many individual datasets may be protected with different authentication techniques. Any service that provides access to these datasets may not only have to deal with the heterogeneous nature of data itself but also with heterogeneous authentication and access procedures.

There may also be restrictions on who can have access to which datasets or datasets may need to be anonymised (substituting untraceable codes in place of personal details) in order to ensure that personal information is not revealed.

Motivation:

- These use cases are the text mining from DC and FHG, general data access cases from UU, and system diagnostics from TECHNION,
- DC text mining use cases: Data and document repositories are distributed over different business units and departments. This implies different access policies on the distributed data. Therefore authentication must not only take place for the provided grid services in general, but the grid service itself must do authentication on a per server/repository basis.

Nonfunctional requirement 1 (supplementary): Useable from firewall-protected environments.

Rationale: To protect systems from unauthorized access, many networks make use of a firewall. This restricts external access to particular ports. Ideally the data services and tools will not be affected by the restrictions set by a typical firewall.

Initial Assessment: Most firewalls will allow the HTTP protocol on port 8080 This is sufficient to allow Web access but not allow unauthorized access beyond

the confines of public directories. Thus any protocol that makes use of port 8080 is likely to be accepted by most firewalls.

Motivation:

- DC text mining use cases: Data and document repositories at DC are distributed over different business units and company departments. The DC intranet is not only protected against the internet by a firewall, but different units within the company intranet are also protected against each others by firewalls,
- Data mining distributed medical databases use case,
- Ecological use case.

Nonfunctional requirement 2 (supplementary): Use secure, encrypted channels.

Rationale: It may be necessary to pass sensitive information across the Internet. In such circumstances the information should be encrypted to ensure that sensitive information remains secure.

Initial Assessment: The inclusion of secure socket layer (SSL) software will encrypt data traffic to and from the server.

Motivation:

- Text mining use cases from DC,
- Data mining distributed medical databases use case.

3.3 Accessing and selecting data

Functional requirement 4: Initiate data service in response to successful request

Rationale: Assuming that a data service will be connection-oriented a session is required that will maintain the current state of the data service.

Initial Assessment: Data services can also be connectionless, like Web servers. In this case only one instance of the data service is needed, but each request must be self-contained and include authentication details where necessary. This means that a single request has the potential to be very complicated. On the other hand, a connection-oriented data service will maintain a current state, like FTP. Each individual request is simpler because its context does not have to be restated every time it is passed to the data service.

Motivation: All use cases.

Functional requirement 5: Data service provides access and mediation services to one or more datasets

Rationale: A data service may provide access to one or more datasets. The client application requires a consistent approach to access and handle the data provided by the data service. A mediation service will provide an integrated view of distributed data to the client application.

Initial Assessment: Data sources need to have structure in order for them to be recognized or understood. This structure should be described by metadata. Data sources are usually file orientated or database orientated. Sources of datasets include:

- File systems e.g. NFS or AFS as a flat file or collection of flat files such as ASCII text files that have been formatted according to some clear convention or documents from which data can be extracted and then presented in a structured manner,
- Structured or semi-structured XML data,
- Database management systems: particularly relational (e.g. ORACLE or MySQL),
- Abstract or virtual data that has been derived from a subset of one or more data sets,
- Directories of grid services i.e. data that is used to describe available resources, to support the operation of the grid, and to describe its state and configuration.

Motivation: Most, if not all, use cases will require access to data for data mining.

Functional requirement 6: Data service processes basic queries from client

Rationale: The data service must accept queries that allow the client to select data and perform other basic operations.

Initial Assessment: Users want to access both data and metadata from the located source(s) and select a data subset for processing or mining. For the client to make decisions about how to handle the data provided by the data service, it is necessary for the client to access the metadata associated with the data. The way in which data is structured will affect the manner in which it can be accessed:

- Data access can involve both reading data from, and writing data to, a data source. These operations will be implemented in different ways for different resources,
- Data access may be needed to update data sources, and to ensure the consistency of all data replicas,

- Data sources have different access mechanisms:
 - File system based commands like open, close, read, write. Data subsets can be selected using file processing scripts,
 - Relational query mechanisms like SQL, XQUERY, XPATH,
 - Mechanisms for hierarchical storage systems.

Diverse sources will result in a diversity of access mechanisms. To enable robust generic access mechanisms we will want:

- Uniform methods to access a data source e.g. use GridFTP, although it is also good for multiple mechanisms to exist,
- To be able to transform data amongst the different data types that are stored in the different data sets,
- To be able to mediate between different data models and database schemas,
- Federated or virtual databases may be required in order to integrate various data sources into a transparent global schema. Individual data sources may still need to be available for direct access,
- Specific views of a data source may need to be provided.

It is often desirable to be able to select a subset of the data accessed from a data source before it is transferred or stored elsewhere. Data selection may be an interactive operation and require sophisticated tools to guide the user:

- At the simplest level a data subset can be selected using relational database query mechanisms,
- At the most complex level there are many different ways of filtering, combining, or processing data, and these methods can require sophisticated code to be executed.

Motivation: All use cases.

3.4 Data transfer

Functional requirement 7: Data service must be able to transfer datasets from one server to another

Rationale: By its very nature, grids are geographically distributed. Therefore, in order to access the resources available at different locations, it must be possible to move datasets from one geographical location to another.

Initial Assessment: Moving data requires no knowledge of the data structures. The basic unit of data transfer and access is:

- Primitives: floats, integers, characters and arrays, images or objects,
- Files: these are uninterrupted sequences of bytes.

GridFTP is the fundamental data access and transport mechanism and it provides a uniform interface to different file storage systems. When moving data it is necessary to consider:

- Maximizing file sharing, minimize replication and to monitor shared file space i.e. perform garbage collection,
- Dealing with failure and restarting transfers,
- Synchronizing updates to all replicas,
- Allowing different types of file transfer:
 - The simultaneous transfer of multiple files with each job individually monitored and managed,
 - Reliable data transfer i.e. GridFTP with some enhanced features.

Motivation: All use cases.

Functional requirement 8: Data service must be able to transfer at least 20 000 files in a row without user interaction

Rationale: In the area of text mining raw data is typically stored in flat files. The number of files containing training data, test data or new data for classification may range from only one to tens of thousands (e.g. data from news agencies). These files usually reside inside a few, if not a single, directory in a file system. As it is unfeasible for users to specify each individual file, data services must be capable of transferring the whole contents of these user specified directories without any additional user interaction.

Initial Assessment: GridFTP is the fundamental data access and transport mechanism and it provides a uniform interface to different file storage systems.

Motivation: Text mining use cases.

Functional requirement 9: Data service must be able to transfer multiple files of at least 500MB in a row

Rationale: In the area of text mining raw data is typically stored in flat files. The number and sizes of files containing training data, test data or new data for classification may range from a great number of files of only a few kilobytes each to very few (i.e. <10) or only one file of hundreds of megabytes.

Initial Assessment: GridFTP is the fundamental data access and transport mechanism and it provides a uniform interface to different file storage systems.

Motivation: Text mining use cases.

3.5 Data pre-processing

Functional requirement 10: Include additional functions such as data cleaning operations and data transformation operations

Rationale: Operations on large amounts of data should be kept as close to the data as possible in order to reduce data transfer overheads. Ideally such operations would be integrated with the data services and tools.

Initial Assessment: Additional grid data service functions may execute specific algorithms on the data, such as imputing in missing values or transforming the data in different ways (discretization, normalization, summarization, etc). This can be done more efficiently by the data service since fewer data transfers are required.

At the extreme level, all the data mining tasks could be performed by the grid data service. Programs (probably either scripts or Java programs) could be uploaded to the grid data service and used to perform tasks on the data.

Motivation:

- Genetic algorithms for gene regulatory reengineering use case,
- Molecular simulations of protein unfolding use case,
- Text mining use cases,
- Data mining distributed medical databases use case.

Functional requirement 11: Data processing taking place near where data is located

Rationale: For large amounts of data, there may be high overheads of transferring the data from where it is stored to where it is processed. There might also be cases, where the right to move or distribute the data to other servers may be restricted due to data privacy or copyright reasons. Where possible, as much processing should take place as near to where the data is located as possible in order to keep data transfer times to a minimum and to avoid inconvenience to other users due to the increased traffic caused by data transfers. Other, non-technical (i.e., legal, ethical) reasons may constrain the transfer/copy of data from one physical location to another.

Initial Assessment: Processing data close to its source may be important in order to:

- Scale computation,
- Reduce the dataset's size before it is transferred,
- Realize virtual datasets i.e. a new dataset that has been derived in some way from one or more data sources. These results could then be

accumulated in other data collections: this is similar to virtual data warehousing,

- Format the dataset (into XML for example) before it is transferred,
- Access and process data which can not be redistributed or transferred,
- Process highly dynamic data, where data changes faster, than transferring the data to another place would take.

Motivation:

- Text mining use cases: Text data should be converted in place from their original document format (different binary and proprietary file formats like MS Word, PowerPoint, PDF, ...) to a data format suitable for performing text mining and retrieval (e.g. ASCII, XML, vector-representation, ...). If the conversion takes place on the data repository, the converted data can be cached for future queries and analyses,
- Genetic algorithms for the gene-regulatory network re-engineering use case,
- Molecular simulation of protein unfolding use case.

Functional requirement 12: Storing of intermediate results from preprocessing tasks for further usage by other algorithms

Rationale: In data mining typically many preprocessing tasks have to be performed before the actual data mining/analysis algorithm can be applied. These preprocessing tasks each modify the original data set and provide the intermediate results as input for the next task. While not all of these intermediate results need to be stored permanently, some of them are potentially reusable either at later stages of the process (e.g. during a classification phase) or when repeating the whole process with different parameters (e.g. of the data mining algorithm). The reuse of these results would then lead to a significant reduction of computing/processing time as the intermediate results do not have to be recomputed.

Initial Assessment: For this purpose these intermediate results need to be stored somewhere in a grid environment and be made available to users. Furthermore, to find the intermediate results resulting from pre-processing tasks performed to train an individual model, these results need to be mapped to this model in some way.

Motivation: Text mining use cases, ecological use case

3.6 Annotation of Data mining models with provenance information

Functional requirement 13: Provide mechanism for permanently recording provenance information about intermediate results in certain stages of a data mining process

Rationale: It is not enough to only store intermediate results on some node in a grid, since these results need to be found by potentially different users in later stages. Therefore, these intermediate results need to be annotated in a way that allows users to search for them and to receive their storage location. Minimal provenance information that needs to be recorded permanently in order to enable users to find these results even after a longer period of time includes the following:

- Date of creation,
- Name of user who created it,
- Name and version of algorithm/program that created it.

Initial Assessment: Annotation of all sorts of intermediate results in a generic way that covers all aspects (e.g. how they were generated, information about the original data used, etc.) is very difficult and cumbersome to implement due to the lack of a universal standard for this purpose. However, if only limited provenance information is stored (e.g. in a central repository) basic data services can be applied for this task.

Motivation: Text mining use case, ecological use case

Functional requirement 14: Provide mechanism for locating intermediate results based on provenance information stored with the results

Rationale: In principal two different ways of using intermediate results exist. Intermediate results can serve as input to algorithms that are executed immediately after these results have been generated. In this case the location of these results can be passed to the algorithm and no search operation for them has to be performed. However, in some cases intermediate results will also often be stored somewhere in a grid and not re-used immediately. In these cases the user has to specify in the workflow which intermediate results serve as input for certain algorithms he intends to use. As the actual storage location of these results in a grid may be unknown, mechanism are needed that enable the user to receive the location of an individual set of intermediate results base on the provenance information stored together with these results as specified above.

Initial Assessment: Basic data services for querying information repositories such as database and file systems can be used to receive the location of particular intermediate results. However, the user searching for this location needs to know the restrictions on the parameters he can search for that describe these results.

Motivation: Text mining use case, ecological use case

3.7 Data mining tasks

Functional Requirement 15: Decoupling WEKA components

Rationale: Decoupling Weka components (data mining services, etc.) for optimal extensibility and platform/language independence

Initial Assessment: Weka [Weka04] is a widely known and accepted open source data mining tool. It contains more than 100 algorithms for data preprocessing, data mining, validation, and visualization of the results. Weka's modular structure and uniform application programming interface facilitates the integration of many of its algorithms into grid systems.

Motivation: Weka's components can be used to build one integrated system encompassing demonstrators from several (or all) Partners. Furthermore, with Weka being able to execute on the grid users can choose between many more additional analysis services.

3.8 Text mining & ontology learning

All text-mining demonstrators will be realized on decentralized document sources and the processing of the documents will also be realized decentralized in a grid environment. The requirements relating to the "text mining" and "ontology learning" use cases have already been covered in sections 3.2 to 3.4 (accessing distributed data services, data preprocessing and transfer).

3.9 Workflow editing and submission

Functional requirement 16: Ability to prepare workflows by graphical editor and submit workflows for execution

Rationale: Data mining is a complex process consisting of many tasks for pre-processing, analysis, and visualization. In order to compile these tasks for automatic execution, a graphical workflow editor providing such capability is needed. Also such a visual tool will reduce error rates compared to traditional batch scripting and shield users from the complexity of the underlying system. Furthermore, as the workflow editor will not execute the workflow, it has to be able to submit it to some machine where it is executed.

Initial Assessment: In order to contribute to the system's usability for end users, the workflow editor has to support various demonstrators while hiding as many grid specific aspects as possible from end users. This demands for the editor to provide a generic interface for adding new services during runtime by

using the Data & Analysis Discovery/Location services as explained in functional requirements #1 and #2 to present available services to the user.

The workflow editor will also perform basic validation of the workflow chain ensuring syntactical soundness.

These functions require the computational and data services to provide extensive descriptions about themselves. It has to be carefully evaluated which information is needed in order to keep data exchange and overall complexity to a minimum while still ensuring full functionality.

Execution of the workflow is not part of the editor. Therefore, it is necessary to submit the workflow to some sort of manager, which starts the execution and monitors the progress.

Motivation: The basic idea is to provide an extensible workflow editor, which can be used in many demonstrators.

3.10 Requirements concerning integration of domain knowledge

Functional requirement 17: Ability to store models in a knowledge base

Rationale: The data to be mined in complex problem solving scenarios will increasingly require the integration of existing domain knowledge into the mining process, particularly in knowledge-intensive domains. This knowledge is often dispersed across geographically distributed sites and organizations, either in digital form (ontologies, metadata, knowledge-based or simulation systems) or provided interactively by human experts. Grid-enabled data mining technology will need to provide services and tools to support knowledge-aided data mining.

Initial assessment: Storing formulae in a knowledge database could be highly beneficial for some machine learning algorithms, e.g. Lagrange.

Motivation: Ecological modelling use case.

3.11 Grid infrastructure and middleware requirements

Functional requirement 18: Middleware able to execute generated workflows including conditional control flows

Rationale: Data mining is complex process including many sub-tasks. As some may only be executed if specific conditions are fulfilled and others may be repeatedly executed a specific number of times, the workflow editor has to provide conditional control flows.

Initial Assessment: Conditional control flows are essential to ensure usefulness of the workflow editor, since the workflows of many use cases include branches

and loops according to certain conditions. Without such control flows the workflow editor would only be of limited or no use for these use cases.

Motivation:

- Text mining use cases: Distributed document clustering and training of classifiers takes place as iterative process, being repeated until a certain quality criteria is met,
- Ecological use case.

Nonfunctional requirement 3 (supplementary): The DataMiningGrid services and tools should be compatible with existing grid infrastructure.

Rationale: Use of existing grid infrastructure will reduce development and evaluation complexity. However this will only be possible if the existing grid infrastructure is capable of supporting the required services and tools.

Initial Assessment: Where possible, existing protocols (grid interoperability profile WS-I+) and infrastructure should be used. This may mean compromising on some (functional) requirements of the project, but will enhance compatibility and reduce the risk of project failure.

Motivation: All use cases.

Nonfunctional requirement 4 (supplementary): Grid infrastructure, test-bed for the DataMiningGrid project.

Rationale: Grid infrastructure should be developed by the project Partners for technology testing purposes.

Initial Assessment: Where possible, existing protocols and infrastructure should be used. This may mean compromising on some aspects of the project but will enhance compatibility and reduce the risk of project failure.

Motivation: All use cases.

Functional Requirement 19: Provide monitoring services.

Rationale: Constant system diagnosis is of critical importance in complex, distributed systems. In such systems it is extremely difficult to track and monitor job execution, and to understand what causes failures and poor performance. An effective ongoing analysis of the system's condition is necessary in order to enable corrective actions, and improve system performance.

Initial Assessment: Ongoing analysis of the system's condition enables early detection of the occurring problems, detects points of failure, and exposes

weaknesses that potentially or actually harm the system's performance. The corrective actions and the performance improvement based on this analysis need to be automated to the maximal extent catering for the dynamic and often unpredictable modifications and alterations occurring in the system. The user needs to be notified regarding the status of her job submissions, and provided with the sufficient information to carry out corrective actions, if these are within her capacity as a user (for example, job resubmission with the complete and correct data). The system administrator needs to be updated regarding the reoccurring problems and potential weaknesses, and provided with the sufficient information to reconfigure the system if necessary.

Motivation:

- Grid monitoring use cases,
- Text mining use cases,
- Ecological modeling use case.

3.12 Usability, response times and user-friendliness

Nonfunctional requirement 5 (supplementary): Data services and tools response time is within a reasonable time period.

Rationale: The user will wish to know if an operation has succeeded or failed within a reasonable time period.

Initial Assessment: Response time when client-application connects to a service should be within seconds (10 seconds is realistic response time for job submission confirmation. However, resource allocating facilities of the current distributed systems operate in minutes, rather than seconds. For example, EDG Resource Broker takes on average 5 minutes to execute a simple echo job). Response time when client-application sends a query should also be within seconds in order to inform the user if the query can be processed or not (30 seconds is suggested as a reasonable time). For the DC Text-Mining Use case "Find related documents" even shorter response time guarantees seem reasonable. Otherwise, system should give the user regular progress reports.

Time to transfer a large amount of data will depend on the amount of data to be transferred (no limit can be placed on this).

Motivation: All use cases.

4 Objectives – research goals beyond the scope of the project

In addition to the set of features that should be part of the DataMiningGrid technology (as defined by requirements in the previous section), there are additional features that would be useful for many use cases. These requirements can be classified as “nice-to-have” requirements. These useful, but non-essential, requirements will be addressed by the Consortium if possible, but the Consortium may decide that there are good reasons for excluding them or leaving them to be implemented by a later project. Some of these objectives are not fully defined, and as such need further clarification if they are to be addressed by the DataMiningGrid© Consortium. Note that the order of the objectives below does not imply relative priority or degree of consensus.

Functional requirement 20: Provide mechanism for locating computational services

Rationale: Similar to the data services, the analysis services are also distributed all over the grid. Since ad hoc search techniques are inadequate to locate such distributed services, an automatic mechanism for locating them is needed in order to find them in a quick, easy, and convenient way.

Initial Assessment: This requires an analysis discovery service. Similar to data discovery services, analysis discovery services identify analysis services by matching specified characteristics, attributes, or meta-data such as:

- A description, summary, or overview of the analysis service,
- A description of the input and output data,
- The version of the service,
- Who is the creator, the owner, or the last modifier.

Motivation: All the use cases which will need to locate appropriate computational service(-s) will benefit from automatic service discovery.

Nonfunctional requirement 6 (supplementary): Data services and tools must be scalable

Rationale: Scalable means that the processing larger datasets should be possible without causing the system to grind to a halt.

Initial Assessment: Where possible, data should be divided into subsets and sent to distributed processes or processors. The data can then be processed in parallel and, in theory, should not take significantly longer than the time it would take one data subset to be processed on by one process. In practice there will be additional overheads due to scheduling of processes and distribution of

datasets. Furthermore there may be other bottlenecks, for example if all the data must pass through a single gateway.

Where data is already distributed, it may be possible to avoid bottlenecks by transferring the data directly to the distributed process without passing through the client or a specific service node. However, this raises some technical problems in how to clean the data and prevent repetitions in the data.

Motivation: All use cases.

Nonfunctional requirement 7 (supplementary): DataMiningGrid services and tools must be transparent.

Rationale: The end-user should not need to know about or understand the low-level details of how the technology works. However, users should always be aware of the possible intricacies of distributed systems, like latency, availability and reliability (see [Spol00] and [Spol02]).

Initial Assessment: It is assumed that end-users will be conversant with how to set up data mining workflows. However, the end-user will not necessary be conversant with the structure of the grid technology that underlies the data services and tools. Data services should be presented to the user as if it were a file or any other source of data.

Motivation: All use cases.

Nonfunctional requirement 8 (supplementary): Extensible and “future-proof”

Rationale: The design of the DataMiningGrid services and tools should be flexible enough to allow additional functionality to be included without inhibiting backward compatibility.

Initial Assessment: The key to making a system “future-proof” is simplicity. For example, a simple set of functions and operations that can be combined to perform any task are better than a smaller set of functions that require more arguments and are more likely to require change in the future. If necessary, new functions can be introduced but these should follow the same principal.

Motivation: All use cases.

Nonfunctional requirement 9 (supplementary): Grid infrastructure allows interoperability between heterogeneous programming environments and different operating systems

Rationale: The existing data mining components of the project Partners run on different operating systems (Linux and Windows) and are written in different and incompatible programming language. To reduce development time it must be possible to adopt them to the grid with out having to port them to another programming environment or operating system.

Initial Assessment: Grid interoperability standards should guarantee this requirement. It may be necessary to write glue/bridge code in some cases.

Motivation: All use cases.

Nonfunctional requirement 10 (supplementary): Data service interactions should be kept simple.

Rationale: A complex protocol is more difficult to implement and more likely to fail. The additional functionality provided by a more complex protocol can often be achieved by using several simpler commands.

Initial Assessment: A simple text-based language that is both simple and extensible. Every command has a response in order to confirm the command has been successfully processed.

Motivation: All use cases.

5 Conclusions and future work

While investigating the generic data mining requirements of emerging grid-enabled problem solving environments based on a selected set of representative application sectors and problem domains, the DataMiningGrid© Consortium aims to develop algorithms and tools facilitating grid-based data mining services for future and emerging complex problem solving environments.

The deliverable D11(2) was prepared jointly by the Partners to address some important requirements for data mining in grid computing environments. As such, it represents only a subset of the most important requirements that the Consortium has to address at the beginning of the DataMiningGrid project. More requirements will certainly arise during the next phases of the project.

Due to the impending data influx in many sectors, the need for data mining is likely to increase. With the advent of increasingly flexible and powerful grid-computing infrastructures, it is high time to think of and to instigate the development of grid-enabled data mining services. Such services would facilitate:

- a dynamic and secure way of accessing, retrieving, and manipulating (join, subset selection, filtering, etc) data sets from heterogeneous and distributed sources,
- a dynamic and secure association of these data sets to operations provided by data mining servers available in a distributed computing environment, i.e. the grid,
- a dynamic and secure execution of these operations on the data sets,
- a dynamic and secure pipelining of such operations and the resulting intermediate data sets,
- a dynamic and secure allocation and addition of new data mining services and operations (servers), new databases and data sets to the grid,
- a highly interactive, intuitive, and secure way for users to define, execute, monitor, and manage a data mining workflow in such a distributed data mining environment.

Along with these developments the DataMiningGrid© Consortium shall demonstrate the deployment of the developed Web-based, grid-enabled, data mining applications, modeling tools and services in a carefully selected sets of representative application sectors (see Figure 1). The selected applications and technologies include: re-engineering of gene regulatory networks via distributed genetic programming, analysis of biological databases for gene/protein annotation, data mining based monitoring of grid systems (including analysis of recurring failure and descriptive analysis), distributed text classification and ontology learning for customer relationship management and quality management, finding related and similar documents in the intranet, mining of digital (scientific) libraries, analysis of distributed medical databases for endemic goitre and iodine deficiency studies, distribution of algorithms for mining of data in the context of sanitary engineering, and mining of literature databases (including equation discovery, data check/filtering).

6 References

- [Avaki04] AVAKI at <http://www.avaki.com/>
- [Bax02] Baxter R, A Complete History of the Grid at www.nesc.ac.uk/talks/sdmiv/SDMIV-25Oct2002-Baxter.pdf
- [Condor04] The Condor Project at <http://www.cs.wisc.edu/condor/>
- [Condor-G04] <http://www.cs.wisc.edu/condor/condorg/>
- [Czaj04] K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. David Snelling and S. Tuecke, "From Open Grid Services Infrastructure to WSResource Framework: Refactoring & Evolution," <http://www-fp.globus.org/wsrf/OGSI%20to%20WSRF%201.0.pdf> , Tech. Rep. 1.0, 12/2/2004, 2004.
- [DataGrid04] The DataGrid Project at <http://eu-datagrid.web.cern.ch/eu-datagrid/>
- [Dee90] Deerwester, S. S. & Dumais, T. K. Landauer, G.W. Furnas, and R.~A. Harshman (1990). Indexing by latent semantic analysis, in: Journal of the American Society of Information Science, 41(6):391--407, 1990.
- [DQP04] www.ogsadai.org.uk/dqp/
- [Dub01] Dubitzky, W., Krebs, O., and Eils, R. (2001), 'Minding, OLAPing, and Mining Biological Data: Towards a Data Warehousing Concept in Biology', Proc. Network Tools and Applications in Biology (NETTAB), CORBA and XML: Towards a Bioinformatics Integrated Network Environment, Genoa, Italy, pp78-82.
- [ELDAS04] www.edikt.org/eldas
- [Fer03] Ferreira, L., Jacob, B., Slevin, S., Brown, M., Sundararajan, S., Lepasant, J., Bank, J. (2003) Globus Toolkit 3.0 Quick Start <http://www.redbooks.ibm.com/redpapers/pdfs/redp3697.pdf>
- [Fos02] I. Foster, C. Kesselman, J.M. Nick, S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, at www.gridforum.org/ogsi-wg/drafts/ogsa_draft_2.9_2002-06-22.pdf
- [Fos04] I. Foster, J. Frey, S. Graham, S. Tuecke, K. Czajkowski, D. Ferguson, F. Leymann, M. Nally, T. Storey, W. Vambenepe and S. Weerawarana, "Modeling Stateful Resources with Web Services," <http://www-106.ibm.com/developerworks/library/ws-resource/ws-modelingresources.pdf> , Tech. Rep. 1.0, 01/20/2004, 2004.
- [Fos99] I. Foster, C. Kesselman, The GRID, Morgan Kaufmann Publishers, Inc., San Francisco, 1999.
- [GGF04] The Global Grid Forum Database Access and Integration Services Working Group at http://www.gridforum.org/6_DATA/dais.htm
- [Globus04] The Globus Alliance at <http://www.globus.org/>

- [Gra97] Ingo Graf and Ulrich Krebel and Jürgen Franke, Polynomial Classifiers and Support Vector Machines, 1997, 00 -- 00, Inter. Conf. on Artificial Neural Networks, Zürich
- [Gru93] Gruber, T. R. (1993): Toward principles for the design of ontologies used for knowledge sharing, in: Formal Analysis in Conceptual Analysis and Knowledge Representation, Kluwer 1993.
- [Hof01] Hofmann, Thomas, Unsupervised learning by probabilistic latent semantic analysis, in: Machine Learning,42:177--196, 2001.
- [Joa98] Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features, in: Proceedings of the Tenth European Conference on Machine Learning (ECML '98), Lecture Notes in Computer Science, Number 1398
- [Lan97] Landauer, Thomas K. & Dumais, Susan T. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge Psychological Review 104 (2): 211--240, 1997.
- [Leo02] Edda Leopold & Jörg Kindermann (2002): Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?; in: Machine Learning 46, pp. 423 - 444.
- [Man99] Manning, C. D. & Schütze, H. (1999): Foundations of Statistical Natural Language Processing, MIT Press: Cambridge MA, London.
- [OGSA-DAI04] www.ogsadai.org.uk/docs/current/doc/DAIOverview.html
- [Park03] Park, K., and Kanehisa, M. (2003), 'Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs', Bioinformatics, Vol. 19, pp. 1656-1663.
- [Rom02] M. Romberg, "The UNICORE Grid Infrastructure", Scientific Programming Special Issue on Grid Computing, 2002, 10, pp. 149-158.
- [Sal83] Salton, G.& McGill, M. J. (1983) Introduction to Modern Information Retrieval, McGraw Hill, New York.
- [Spol00] Spolsky Joel; Three Wrong Ideas From Computer Science, August 22, 2000; www.joelonsoftware.com/articles/fog0000000041.html
- [Spol02] Spolsky, Joel; The Law of Leaky Abstractions, November 11, 2002; <http://www.joelonsoftware.com/articles/LeakyAbstractions.html>
- [Stan04] Stankovski V., May M., Franke J., Schuster A, McCourt D., Dubitzky W., A Service-Centric Perspective for Data Mining in Complex Problem Solving Environments, H.R. Arabnia and J. Ni (eds) Proc of Int'l Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'04), Vol II, pp. 780-787, 2004.
- [Stork04] www.cs.wisc.edu/condor/stork/papers/stork-icdcs2004.pdf
- [Stur03] Sturgeon, B., McCourt, D., Cowper, J., Palmer, F., McClean, S., and Dubitzky, W. (2003), 'Can the Grid Help to Solve the Data Integration Problems in Molecular Biology?', 3rd International Symposium on Cluster Computing and the Grid (CCGRID 2003) , Tokyo, Japan, pp. 594-600.

[Tom99] M. Tomassi. Parallel and distributed evolutionary algorithms: A review. In K. Miettinen, M. Makela, P. Neittaanmaki, and J. Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, pages 113-133. J. Wiley and Sons, Chichester, 1999.

[Weka04] – Weka Homepage; <http://www.cs.waikato.ac.nz/ml/weka/>T

[Weka99] – Witten, Ian.H, Frank, Eibe; *Data Mining Practical Machine Learning Tools with Java Implementations*; Morgan Kaufmann; 1999