



CONTRACT NUMBER 508830

**DEISA**  
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR  
SUPERCOMPUTING APPLICATIONS**

**European Community Sixth Framework Programme**  
**RESEARCH INFRASTRUCTURES**  
Integrated Infrastructure Initiative

Support of a series of important materials science codes  
with grid-enabled features

Deliverable ID: DEISA\_D-JRA1-6

Due date: April 30th, 2007

Actual delivery date: May 28, 2007

Lead contractor for this deliverable: Organisation, Country

Project start date: May 1<sup>st</sup>, 2004

Duration: 4 years

<b>Project co-funded by the European Commission within the Sixth Framework Programme (2002-2008)</b>		
<b>Dissemination Level</b>		
<b>PU</b>	Public	X
<b>PP</b>	Restricted to other programme participants (including the Commission Services)	
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission	
<b>CO</b>	Confidential, only for members of the consortium (including the Commission Services)	

## Table of Contents

Table of Contents .....	1
1. Introduction .....	2
1.1 Executive Summary.....	2
1.2 References and Applicable Documents .....	2
1.3 Document Amendment Procedure .....	2
1.4 List of Acronyms and Abbreviations .....	2
2. Support of further applications.....	4
2.1 NAMD .....	4
2.1.1 Input file interpretation .....	4
2.1.2 NAMD EJBs and Web Service .....	5
2.1.3 NAMD Portlet Application .....	7
2.2 LAMMPS .....	9
2.3 GROMACS .....	9
2.4 PWscf .....	9
3. Enhancements of the portal.....	9
4. Interoperability demo DEMO with AHE at SC06.....	10
5. Contributions by HLRS .....	10
6. Workplan Update .....	10

## 1. Introduction

### 1.1 Executive Summary

Work in months 31 to 36 was focused on the extension of application support in the materials science portal. Enhancements include the functionality of the portal, the number of supported applications, and the maintenance of the applications so far provided.

For the simulation codes LAMMPS and NAMD application plug-ins have been developed and integrated into the portal. For the GROMACS package, a basic selection and submission mechanism via the portal has been implemented, and studies have been started to analyze the options for plug-ins for GROMACS and for the PWscf simulation package.

In addition, the newly developed interface of the *Application Hosting Environment (AHE)* of UK eScience to the materials science gateway and the UNICORE package was successfully used for interoperability demos during the Supercomputing Conference SC2006 in Tampa, US.

The work of HLRS was focused on the provision of the CP2K simulation package on the NEC SX-6 supercomputer .

The workplan for the next period has also been updated.

### 1.2 References and Applicable Documents

- [1] <http://www.deisa.org>
- [3] <http://www.realitygrid.org/>
- [4] [http://www.realitygrid.org/publications/NeSC\\_paper\\_edit.doc](http://www.realitygrid.org/publications/NeSC_paper_edit.doc)
- [5] <http://gridsam.sourceforge.net>
- [6] <http://www.omii.ac.uk/>
- [7] Condor Project Homepage <http://www.cs.wisc.edu/condor/>

### 1.3 Document Amendment Procedure

### 1.4 List of Acronyms and Abbreviations

API	Application Programming Interface
CP2K	Car-Parrinello molecular dynamics 2000
CPE	Common Production Environment
DESHL	DEisa Services for Heterogeneous management Layer
EJB	Enterprise Java Bean
GPFS	General Parallel File System
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Secured version of HTTP
IE	Microsoft Internet Explorer
Java EE	Java Enterprise Edition

JCP	Java Community Process
JRA	Joint Research Activity
JSDL	Job Services Description Language
JSR	Java Specification Request
MDA	Model Driven Architecture
MVC	Model-View-Controller pattern
NJS	Network Job Supervisor
O/R Mapping	Object/Relational Mapping
OASIS	Organisation for the Advancement of Structured Information Standards
OMG	Object Modelling Group
PDA	Personal Digital Agent
POJO	Plain Old Java Object
SA	Service Activity
SOA	Service Oriented Architecture
UML	Unified Modelling Language
UNICORE	UNiform Interface to COmputing REsources
WS	Web Service
XHTML	XML syntax compliant HTML
XLST	XML Style Sheet
XMI	XML container syntax for UML
XML	eXtensible Markup Language
XSD	XML Schema Definition

## 2. Support of further applications

After fully supporting the CP2K package in DEISA, work on applications in months 31-36 was focused on the support and integration of further important applications. As already stated in the 30-month deliverable, code candidates for enabling work have been selected with respect to the following categories:

- Enabling work that facilitates code instrumentation via an application plug-in
- Enabling work that requires code optimization and eventually restructuring
- Further enabling work, eventually in the domain of workflows

Naturally, code integration into the materials science portal would be the preferred task for JRA1, while code optimizations and workflow set-ups should be preferentially addressed by eDEISA eSA4-T1+T2.

According to the workplan established in the 30 month deliverable, the codes that have been dealt with in this JRA include LAMMPS, NAMD, GROMACS and the PWscf / QUANTUM ESPRESSO package.

### 2.1 NAMD

NAMD, recipient of a Gordon Bell Award (2002), is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems. (<http://www.ks.uiuc.edu/Research/namd/>).

NAMD is a parallel molecular dynamics code for the simulation of large biomolecular systems. It is designed for high-performance computing systems and received the Gordon Bell Award in 2002.

For the simulation preparation four files are needed: a CHARMM force field; an X-PLOR structure file in the PSF format describing the molecular structure; the initial coordinates of the molecular system as a PDF file; the NAMD configuration file.

NAMD is capable of reading CHARMM and X-PLOR parameter files in the CHARMM19, CHARMM22 or CHARMM27 versions.

The file containing the structure information can be generated with the help of the psfgen tool, shipped with NAMD. The structure is derived from PDF files containing the atom/molecule positions, which suffices to determine the overall structure in most cases. Hence this tool has been integrated into this plugin. The user alternatively has the possibility to upload a structure file. Coordinate files are specified as PDB files.

NAMD outputs DCD trajectory files. These files have a binary form and are not endian-ness aware. Therefore, one has to take care of a conversion between the different endian formats, e.g. big-endian on Power under AIX and little-endian on a Intel or AMD based Linux box. The plugin has an option for the conversion of trajectory files from one endian-ness to the other employing conversion programs shipped with NAMD.

#### 2.1.1 Input file interpretation

While the configuration files for CPMD, CP2K, and PWscf are organized hierarchically, a NAMD (and LAMMPS) configuration file is flat, and hence cannot be as elegantly described in terms of a XML Schema definition. Nevertheless, there are

attributes which have to precede others, and some may occur multiple times. The static part of the grammar of a NAMD configuration file was synthesized in an XML schema definition. The idea is to convert an ASCII NAMD configuration file with predetermined conversion rules into XML and then check this XML file against the schema definition. Possible validation errors will be traced back to the actual location in the original configuration file.

The great advantage of this approach is that supporting multiple input file versions is as simple as teaching the ASCII to XML converter to support a new version. The latter is achieved by simply adding an appropriate XML schema definition for syntax and partial semantic checking, and an XML style sheet for converting an XML representation back into the ASCII version understood by NAMD. The latter is only needed when the user does not provide ASCII in the first place as in the case of the NAMD Portlet application.

### 2.1.2 NAMD EJBs and Web Service

As in the cases of CP2K and CPMD, an EJB has also been developed for NAMD which encapsulates the methods offered by the input file interpretation code.

In order to increase the flexibility for the user and in order to enhance and widen the usability of the JMEA application, Web Service Interfaces to the EJBs were introduced which allow a simple usage of JMEA over wide area networks.

As for the JMEA job management interface itself, it is oriented on the work of OGF's SAGA-WG and OGSA-BES-WG which define a simple API for grid applications (SAGA) and basic execution services (BES). Hence, it offers to the user a factory service for the creation of stateful web service instances, a management service for performing management tasks such as starting or stopping the acceptance of service creation, and of course the interface to the stateful service which changes the state of a particular service instance.

The Factory service method definition follows a current BES draft. They allow the creation of activities (the stateful web service instances), explicit termination of those, and retrieving their status. Furthermore, methods for describing activities as well as the factory service itself have been introduced. Currently, their implementation does not offer more than just returning a list of activities created by this user and a textual description of the factory service. In future, they could be used in conjunction with Service Level Agreements.

The Factory Service methods (following a BES draft) are:

- CreateActivity
- TerminateActivity
- GetActivityStatus
- GetActivityDocuments
- GetFactoryAttributesDocument

The management service allows a suspension and resumption of the factory service. By doing this, service components can be upgraded without making the service completely unreachable for a potential user. The same holds for the case of a DoS attack situation.

The Management Service (following a BES draft) are:

- StartAcceptingActivity
- StopAcceptingActivity

The methods which change the state of each activity deal with applications like NAMD directly and are augmented by methods from WS-BaseNotification in order to allow users to be informed about state changes such as job submission and status updates.

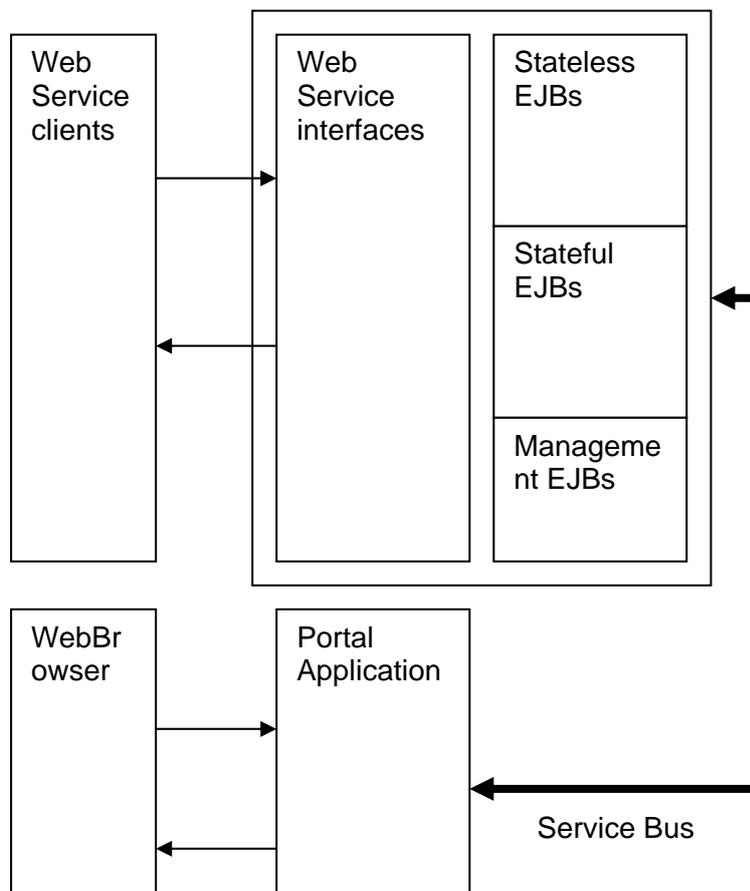
The Stateful Service methods (Application Specific + WS-BaseNotification) are:

- SetInput
- Validate
- Submit
- Cancel
- Subscribe
- Unsubscribe

Figure 1 sketches the architectural aspects of the described implementation. In addition, it outlines the connection of the NAMD Portlet application to this infrastructure employing a bus system. This bus system is currently under development. As described below, the Portlet application currently uses the web services directly with the exception of the aspects of job management where the sketched service bus is already in production.

The creation of the XML schema definition file which contains the syntax description of the XML translation of the NAMD ASCII input file has to be carried out manually, since the only source of information is the documentation (and, in principle, the program itself). Hence, great care has to be taken and extensive tests have to be carried out in order to ensure that everything was translated correctly.

While the functional aspects of all methods presented above are NAMD inherent, the architecture can be applied to every application from that domain. Hence, as described in the next sections, this architecture was applied to LAMMPS and, with some, limitations also to GROMACS.



**Figure 1: Sketch of Architecture and Integration of the Web Service enhanced JMEA implementation with NAMD application components.**

### 2.1.3 NAMD Portlet Application

The NAMD Portlet application provides a versatile graphical user interface for accessing the set of services which allows creation, submission, and monitoring of NAMD jobs. Implementing the web application as a Portlet application allows an easier redeployment of the application in other context, such as other Grid projects. For the NAMD Portlet application we completely adhere to the JSR-168 standard. Gridsphere is used as the current Portlet container. In principle, the application is deployable also in containers like Liveray, Apache Portals, eXo, jPortlet, JBoss Portal, and many others. Note that some of these may require the creation of portal container specific deployment descriptors.

The NAMD Portlet application supports the functionality of the NAMD Web Service at large. Furthermore, it allows direct interaction with the JMEA job submission, diverse data services and discovery services based on UDDI (currently not present in DEISA) and Globus MDS v4.

Figure 2 and Figure 3 give snapshots of the Portlet applications. The snapshots show the configuration summary for accessing one of the several available Web Services for using NAMD.

Portlets can have several views. In the standard view, most portlets of the application present summary views of information. The edit view is used for presenting forms for modifying configuration data. In the help view, information for assisting the user in choosing the right configuration option is displayed.

The user has similar options to create and modify the configuration as in the case of the CP2K plugin which has been described in an earlier deliverable. The configuration file editor again supports syntax highlighting and every logical section is represented by a form. Due to the absence of a hierarchical structure, groups are defined on a logical level, as indicated in the NAMD user manual.

Currently, the generation of forms is semiautomatic; hence the starting point for the creation, the XML schema definition, has to be created manually as already stated in the last subsection. Fortunately, the number of keywords here is much smaller than in the case of CP2K and CPMD.

Dynamic constraints within the configuration file are currently not checked for all possible cases since they have to be interpreted from the documentation or the source code itself. Obvious ones, like having either Amber or Gromacs force fields configured, are however implemented. In future, it may be desirable to have a Force Field editor where a user can select existing force fields or create new ones. The assistance of the creation of new force fields has been postponed since it is not clear if this is of scientific relevance.

The Portlet application is available within the materials science portal at <https://mas.deisa.org/>.

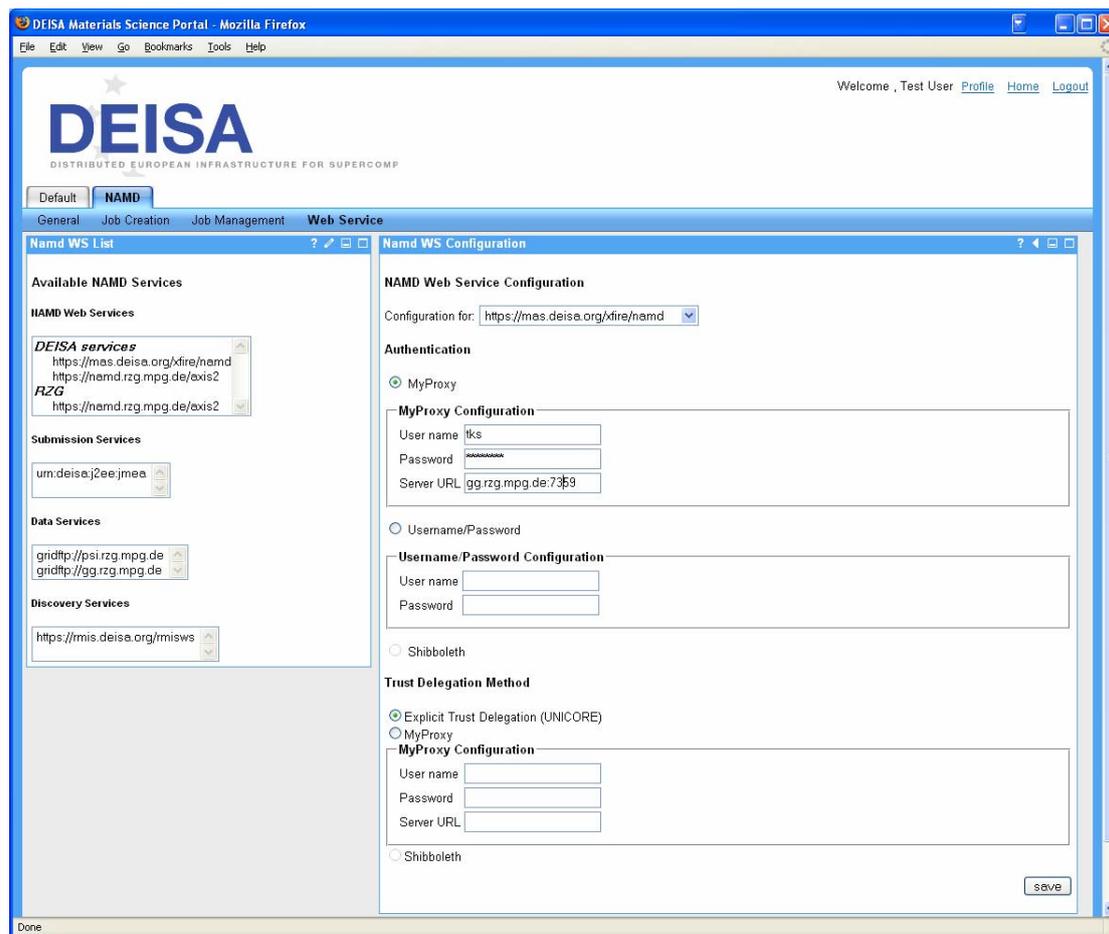


Figure 2: Snapshot of the NAMD Web Service configuration in edit view.

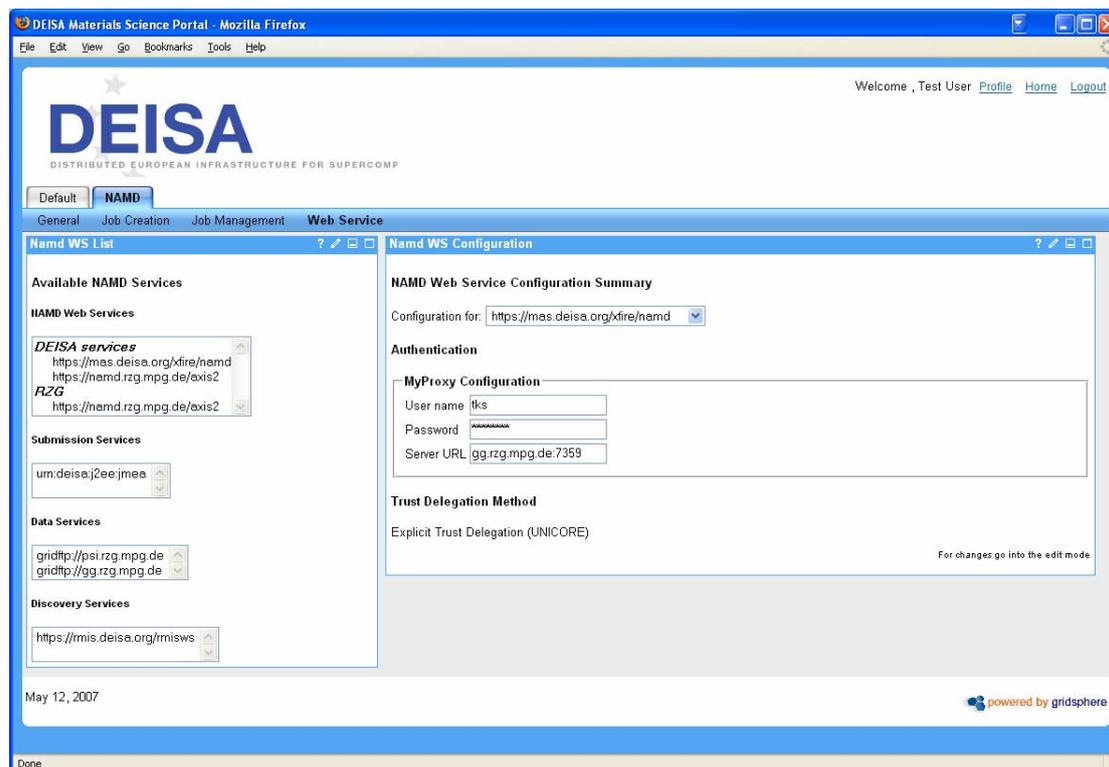


Figure 3: Snapshot of the NAMD Web Service configuration in standard view

## 2.2 LAMMPS

LAMMPS is a classical molecular dynamics simulation code designed to run efficiently on parallel computers (<http://www.cs.sandia.gov/~sjplimp/lammps.html>). It was developed at Sandia National Laboratories, a US Department of Energy facility, with funding from the DOE. It is an open-source code, distributed freely under the terms of the GNU Public License (GPL).

The creation of a LAMMPS Web Service and a LAMMPS Portlet application was carried out analogous to that of NAMD, since LAMMPS and NAMD are similar enough in terms of user interaction. The set of keywords is similar but not identical, so that although the procedure was the same as in the case of NAMD, every keyword again had to be treated manually. In addition, LAMMPS supports its own way of specifying force fields. A treatment of this case also had to be added,

## 2.3 GROMACS

GROMACS is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles (<http://www.gromacs.org>). It is primarily designed for biochemical molecules like proteins and lipids that have many complicated bonded interactions, but since GROMACS is extremely fast at calculating the non-bonded interactions (that usually dominate simulations) many groups also use it for research on non-biological systems such as polymers.

GROMACS is a highly efficient code with limited scalability. A job submission service was provided in the materials science gateway based on a script job submission. An investigation has been started if and how an application plug-in for the materials science gateway could be realized.

## 2.4 PWscf

**PWscf** (<http://www.pwscf.org>)

Discussions with an author of the PWscf package have started concerning the support of PWscf in the materials science Gateway. Design and implementation of a plug-in has been envisaged as a next step.

## 3. Enhancements of the portal

Enhancements of the materials science gateway were focused on the introduction of GridSphere as the web application hosting framework, and on improvements of the layout design and functionalities.

GridSphere has been selected as the new Framework for hosting web applications. It supersedes the cocoon framework employed previously. This step has been done upon request of several users especially from ETH Zurich and Italian Universities interested in embedding the CP2K plugin and the planned PWscf plugin in their Portlet Containers.

Enhancements of the look and feel were introduced by employing the Yahoo user interface library, a javascript library with user interface components which can interact with a server side in an AJAX fashion. Currently, work is being carried out not only to port old cocoon applications to Portlet applications but also to unify the look and feel using Yahoo widgets in combination with the standard HTML widgets.

## 4. Interoperability demo DEMO with AHE at SC06

The recently developed interface of the *Application Hosting Environment (AHE)* of UK eScience to the materials science gateway and the UNICORE package was successfully used for interoperability demos during Supercomputing Conference 2006 in Tampa, US. Via AHE, series of jobs had been prepared and submitted to target machines both in DEISA and in Teragrid, and results were automatically retrieved after successful calculation in Pittsburgh Supercomputing Centre and at RZG in Garching, Germany.

## 5. Contributions by HLRS

The work of HLRS was focused on the provision of the CP2K simulation package. CP2K has been compiled and installed on the NEC SX-8 at HLRS. For the successful installation several modifications have been necessary as the source code was not compiling as it was provided. Test cases have been successfully executed, although large production cases still cause problems which are under investigation.

## 6. Workplan Update

In the last project year, the work of JRA1 will be continued at the following two levels:

- Enabling work on one or more further important materials science simulation codes, in particular on PWscf;
- Improvements of the materials science gateway: updates, enhancements and integration into the DEISA production infrastructure.

Besides updates, general enhancements and integration into the DEISA production infrastructure, improvements of the materials science gateway will include an extension of the functionality for supporting online data analysis in terms of standard tasks including 2D plotting of simulation results, standard post-processing tasks, case dependent automatic creation of restart jobs (until month 42).

The JRA1 partner HLRS will continue to enable, adapt and optimize relevant applications for the NEC SX-8 vector architecture at Stuttgart.

### Next deliverables:

**42 month deliverable:** DEISA\_D-JRA1-7

Progress report on enabling work of further important materials science codes and integration into the materials science gateway

**48 month deliverable:** DEISA\_D-JRA1-8

Final report on providing important materials science codes and integration into the materials science gateway