Enabling Grid Interoperability at Workflow Level

Dagmar Krefting [1]*, Tristan Glatard [2], Vladimir Korkhov [3], Johan Montagnat [4]
Silvia Olabarriaga [3]

[1]Charité - Universitätsmedizin Berlin, Hindenburgdamm 30, 12200 Berlin, Germany
[2]CREATIS-LRMN, 21 avenue Jean Cappellle, 69621 Villeurbanne Cedex, France
[3]Academic Medical Center of the University of Amsterdam, Meibergdreef 9, 1105AZ Amsterdam, Netherlands
[4]I3S laboratory, CNRS, 930 route des Colles, BP 145, 06903 Sophia Antipolis Cedex, France

ABSTRACT

Motivation: In the last years various distributed computing infrastructures (DCIs) have been developed to support national and international research activities. Today several applications from diverse domains have been ported to them. For example, workflow-based grid applications for medical imaging have been developed in the Netherlands within the VL-e project, in France within the EGI biomed VO and in Germany within the German medical D-Grid projects. These applications are based on the resources and workflow systems provided by respective grid infrastructures, and researchers now face difficulties to exchange applications and data across the DCIs. This would be important to obtain access to additional resources and to enable sharing of applications and methodology. Unfortunately today the mobility across DCIs at application level is hardly supported.

Results: The European project SHIWA - Sharing Interoperable Workflows for large-scale scientific simulations on Available DCIs[1] aims to realize interoperability at workflow level. This will allow domain researchers to share and reuse their scientific workflows across DCIs. Different use cases are identified which result in a two-fold approach: coarse-grained and fine-grained workflow interoperability. First results are presented for two pilot applications - neuroimaging and chemistry - and two workflow systems - MOTEUR and GWES. We analyze the similarities and differences between these systems, and show implementation strategies for easily combine and translate scientific workflows. These results enable sharing and reusing workflows and grid services between EGI, D-Grid, and Dutch Grid infrastructures

Availability: The workflow managers GWES and MOTEUR are both open source and free for academic use. The mentioned workflows will be published within the SHIWA platform.

Contact: dagmar.krefting@charite.de

1 INTRODUCTION

Several distributed computing infrastructures are in production now, shared by scientists from different research domains. Examples are the European Grid Initiative (EGI)[2], which is now restructured into a federation of national Grid initiatives (NGIs); the US-American TeraGrid[3]; and many national or regional grid projects like NGS[4], D-Grid[5], or NorduGrid[6]. They differ in the adopted middleware, for example gLite in EGI, Globus Toolkit in NGS, Teragrid and D-Grid, Arc in NorduGrid and Unicore in D-Grid. Even when the basic middleware is the same, the DCIs differ in data management, authentication and authorization, security policies and middleware versions. This makes it very difficult for domain researchers developing their application for a certain DCI to benefit from other implementations available in other DCIs. Grid workflow systems enable a high level of virtualization and simplify grid integration of complex processing pipelines [4, 5, 9]. Whereas the process knowledge is contained within the workflow description, a lot of knowledge about the underlying infrastructure is provided by the workflow engine. System-specific failure recovery is often found within the workflow engines, and application-specific error handling is present within the workflow description. This is important because domain researchers - the prospected end users of the academic research infrastructures - need to use these workflows as services out of the box without expertise in the grid technology. Workflow interoperability is important to facilitate user-friendly exchange of existing workflows between users, workflow systems and DCIs.

1.1 Cases for workflow interoperability

Sharing the process and system knowledge by sharing grid workflows is of high interest for scientists to fully exploit Grid infrastructures and concentrate on their research topics. Workflows not only contain domain-specific methodology and expertise, but they also encapsulate knowledge about porting the applications to a particular DCI reliably. By executing a pre-defined workflow on a new set of data, the user is actually reusing all this valuable knowledge, and takes benefit from the DCI more effectively. In practice existing workflow implementations are limited to a particular workflow system and an underlying DCI, which are often determined by external factors such as available expertise close by, funding, or existing infrastructure. Potentially, these

*to whom correspondence should be addressed

[1] http://shiwa-workflow.eu

[2] http://www.egi.eu/

[3] https://www.teragrid.org/

[4] http://www.ngs.ac.uk/

[5] http://www.d-grid.de/

[6] http://www.nordugrid.org/

implementations/workflows could be exchanged between groups to prevent duplication of effort in both organizations, or they could be combined and extended to implement more complex studies. Different use cases can be identified.

Using other workflows as services. Someone has developed a successful grid workflow to implement an application for the so-called *target DCI*. Users from another DCI (native DCI) want to use this application, but they are not familiar with the workflow system or target DCI. Instead they want to use the application through their usual environment, so the workflow is executed as a service on the target DCI, assuming that the user has access to the application and the DCI.

Transferring a workflow to the native DCI. Sometimes the user cannot employ the target DCI, for example, due to missing authorization or security constraints that impede the data transfer outside the native DCIs. In such cases both the workflow and the application need to be transferred to the native DCI. Only if the same workflow system is available for the user, the workflows can be directly used. In most cases this is not a realistic assumption, therefore translation of the original workflow description to a workflow language compatible with an available workflow system is required.

Increasing scalability by using different DCIs. For some users, the native DCI does not offer the necessary resources to run the computation. For example, when running large-scale simulations, runtime reduction is an important factor for the scientist to speed-up research. In such case the scalability of the system is critical and could be increased by running parts of the simulation on different DCIs. This also requires the workflow and application to be translated to the available workflow systems.

Combining heterogeneous workflows into a meta-workflow. A researcher may be interested in using an existing workflow as part of a new and more complex workflow that implements a more comprehensive application. In this case the workflow could be called as a service in the target DCI or be translated into the native DCI.

We illustrate these cases in two scientific domains shortly described below. Note that there are also other approaches to cross DCI borders by enabling interoperability on other system layers, for example middleware, as aimed by the SAGA project[7], or scheduling, as investigated in the D-Grid project DGSI[8]. However these efforts do not support the application level directly.

1.2 Workflows in Surface Chemistry

Catalytic surface reactions as e.g. CO oxidation on platinum surfaces are of high importance to the chemical industry, but as highly nonlinear systems, they show complex spatiotemporal dynamics, like self-sustained oscillations, chemical turbulence, pattern formation and chaos. In the past years, such so-called diffusion reaction systems have been shown to be an important model systems for extended nonlinear systems, as they are also been found in fundamental physics, meteorology and complex living systems. Numerical simulation is an important tool to study and predict nonlinear systems. Both long-term simulations and ensemble studies using different initial conditions are required to cover the phase space. In particular, a realistic numerical model of CO oxidation on Platinum surfaces is employed for studies on extended nonlinear systems and chaos control [8]. However, its adoption has been limited by shortage of computing and storage resources. Many important features as unsteady states and long-term evolution are therefore not yet investigated. Realization of the simulations as scientific workflows is necessary to handle parameter variation, simulation results and subsequent analysis. Exploitation of different DCIs would allow the realization of large simulation experiments and furthermore the access to potential DCI-specific and workflow-specific benefits for the application (e.g. support of parameter sweeps or GPU-based data analysis). Furthermore, it would be interesting to combine the simulation with workflow-based image and signal processing pipelines.

1.3 Workflows in NeuroImaging

Leading-edge research on neuroscience relies on the combination and integration of the diverse sources of information provided by different methods. An example is a recent Dutch study about the effect of the ecstasy drug in the human brain, where 400 subjects were imaged with various modalities (fMRI, DTI, structural and other) [3]. Besides the large amount of data, each of the involved data analysis methods is in itself complex and often computationally expensive. Grid and workflow technology has been considered by various groups to address challenges for data analysis in Neuroscience. The expert knowledge is captured into the processing pipeline modelled as a workflow, which allows full exploitation of the grid infrastructure by automated execution. Several workflow implementations are already available to perform analysis of neuroimage data with automated multithreaded processing on distributed resources. For example, the AMC and Charité have both developed a variety of workflows for analysis of MRI data [14, 11]. Potentially, these implementations/workflows could be exchanged between groups to prevent duplication of effort in both organizations, or they could be combined and extended to implement more complex studies. Unfortunately the workflows available are not compatible because they were built for different workflow systems and grid infrastructures. As a consequence, great effort is needed today to reuse the different workflows and combine them manually [13].

2 APPROACH

The approach taken by SHIWA is to enable workflow interoperability in two different ways, so-called coarse-grained and fine-grained interoperability. The basic idea of *coarse-grained interoperability* is to leave a workflow in its original language and execute it with the respective workflow engine [10]. Workflow interoperability is realized by nested workflows, where workflows can be enacted on an appropriated workflow engine by service calls. The basic idea of *fine-grained interoperability* is to translate the workflow into another representation that can be executed by other workflow engines. A common representation will be proposed by SHIWA

[7] `http://saga.cct.lsu.edu/`

[8] `http://dgsi.d-grid.de/index.php?id=553`

Table 1. Workflow interoperability cases and requirements

Case	User Access to other DCIs	Data Transfer to other DCIs	Data Transfer between DCIs	Software Transfer to other DCIs
Run foreign workflow	+	+	-	-
Import a workflow	-	-	-	+
Export a workflow	+	+	-	+
Run multi-DCI data generating workflows	+	+	-	-
Run multi-DCI data processing workflows	+	+	+	-

(IWIR: Interoperable Workflow Intermediate Representation[9]) and implemented by supporting workflow engines.

Both approaches are relevant for the described applications. In neuroimaging, coarse-grained interoperability allows to call and combine analysis pipelines accessing diverse DCIs, and fine-grained interoperability brings the workflow-based application to the native or other DCIs. In surface chemistry, coarse-grained interoperability will allow combination with workflow-based analysis pipelines, while fine-grained interoperability increases scalability for large-scale simulations.

Note, however, that not only workflow descriptions have to cross the infrastructures, but also software, data and user privileges. Firstly, users need access to the resources. On DCIs based on the Grid Security Interface (GSI), it implies having a valid X.509 user certificate and becoming a member of a VO supported by the respective DCI. Secondly, most workflows require the processing of input data or at least generate some result files, so using other DCIs also requires access to storage resources which are or can be connected to the respective DCI. Additionally, data transfer between DCIs may be needed in meta-workflows embedding components from different DCIs. Automated data transfer between DCIs require that the user authentication is trusted in both environments. Finally, when a workflow is transferred from a DCI to another, the underlying software invoked by the workflow needs to be transferred and adapted to the new environment and - if workflow translation is required - to the new workflow system interface. See table 1 for the various workflow interoperability cases and requirements.

3 METHODS

The analysis presented here refers to interoperability between the Grid Workflow Execution Service (GWES) used by D-Grid's medigrid VO, and MOTEUR, used by EGI's biomed VO and Dutch-Grid's vlemed VO.

3.1 Application Workflows

Prototypes were realized for the numerical simulation of a chemical surface reaction and neuroimaging data processing on diffusion tensor images. The test workflows are simple but show typical characteristics.

[9] `http://www.ogf.org/gf/event_schedule/index.php?id=2097`

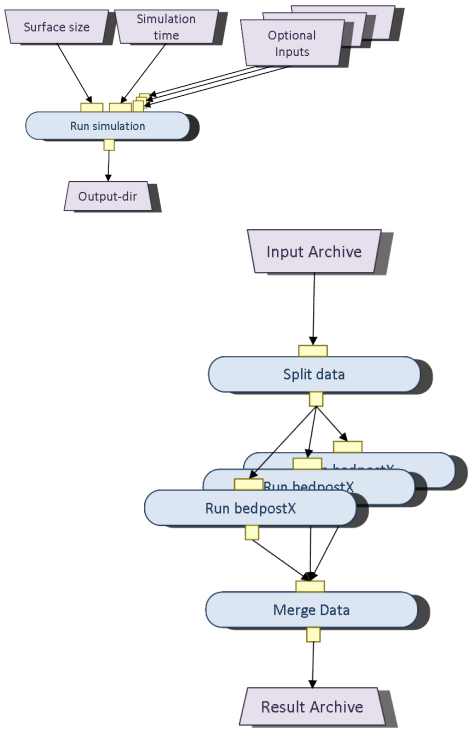

Fig. 1. Semiformal descriptions of the investigated workflows. Top: CO oxidation simulation, Bottom: Diffusion tensor estimation.

Simulation of CO oxidation The workflow is an atomic workflow that simulates the spatio-temporal evolution of the carbon monoxide and oxygen coverage on a platinum (110)-surface. The simulation is written in Matlab and compiled for Linux systems using the Matlab Compiler. Apart from a number of optional simulation parameters, the surface size and the simulation time are required input parameters (more information about the model in [7]). The simulation creates an output directory containing a number of result files (see Fig. 1).

The goal is to run large-scale parameter sweeps of the simulation, exploiting both D-Grid and EGI.

Estimation of diffusion tensors The workflow implements the parallel execution of an FSL method *bedpostX* for estimation of diffusion tensors from DTI images. FSL is a C++ based open-source toolbox with several neuroimaging analysis tools. The grid implementation has been realized within D-Grid [11]. The workflow consists of three steps: (1) Split the image volume in slices; (2) Run bedpostX; and (3) Merge results back to a 3D volume. This is a typical fork/join scheme for parallelized data processing on a DCI. Input data is an archive containing the manually prepared DTI files. Output data is an archive containing the result files. A semiformal description of the workflow is given in figure 1.

The goal is to make the workflow available for users at the AMC. For anonymized research data, the D-Grid implementation can be used. For sensible data, where the possibility of the subject's reidentification cannot be excluded, an implementation on the Dutch Grid is required.

Both cases, despite coming from completely different research domains, have similar requirements for exporting the workflow to another system, and accessing to a foreign workflow. The common steps to realize the scenarios are:

1. Deploy the software on the respective infrastructures

2. Create the interfaces for the respective workflow engines
3. Create workflow descriptions for the respective workflow engines
4. Invoke the application consistently on workflow engines.

3.2 Analysis of the Workflow Systems

GWES (Grid Workflow Execution Service) is a workflow manager designed for grid applications [6]. The core of the GWES is the Grid Workflow Description Language (GWorkflowDL), a Petri-Net based standard for describing workflows using XML [1]. GWES implements high level Petri nets (HLPN) for workflow description, as they can be used directly in order to model transfer and storage of input and output data as well as control data (e.g. the exit status of a workflow step). GWES descriptions may be realized at several abstraction levels, which are then concretized during runtime. GWES provides basic resource brokering and scheduling - using the information provided by the D-Grid Resource Description Language (D-GRDL). The implementation of XPATH enables built-in evaluation and manipulation of tokens [2].

MOTEUR is a workflow engine that was initially developed as an application-porting framework to the EGEE [5]. It currently uses the Gwendia language [12] and can invoke Java Beanshells, Web-Services and GASW activities producing grid jobs that can be submitted to the EGI directly or through a pilot-job system. GASW activities wrap executables using a simple command-line description format [15]. Resource brokering and scheduling are provided by the underlying middleware layer. Data are transferred by grid jobs when they reach computing resources. Applications (executables and libraries) are installed on the fly.

Workflow Execution To make the deployment of the application (including software and execution scripts) as simple as possible, a high level of similarity between the two instances of the workflow-based application is envisioned. In the optimal case, only the workflow description has to be translated.

In a first step, the typical processes for deploying a workflow-based application on the two infrastructures are analyzed.

GWES-applications in D-Grid A typical GWES-based Grid application consists of the following components:

- Software (libraries and executables)
- Wrapper scripts (bash scripts)
- Resource descriptions (D-GRDL documents)
- Workflow descriptions (GWDL documents)

Typically, the software and the wrapper scripts are stored on the headnodes of the D-Grid clusters, while the resource description is stored within a dedicated database. The workflow description is created by the user and uploaded.

MOTEUR-applications in EGI A typical MOTEUR-based Grid application consists of the following components:

- Software (libraries and executables)
- Wrapper scripts (bash scripts)
- GASW descriptors (xml documents)
- Workflow descriptions (GWENDIA or Scufl documents)

Typically, the software, the wrapper scripts and the GASW descriptors are accessible from the EGI file catalog (a.k.a the Logical File Catalog - LFC). The workflow description is created by the user. Fig. 2 shows the schematically the execution process of a Grid application usind MOTEUR (left) and GWES (right). Subsequent steps might be processed several times in parallel.

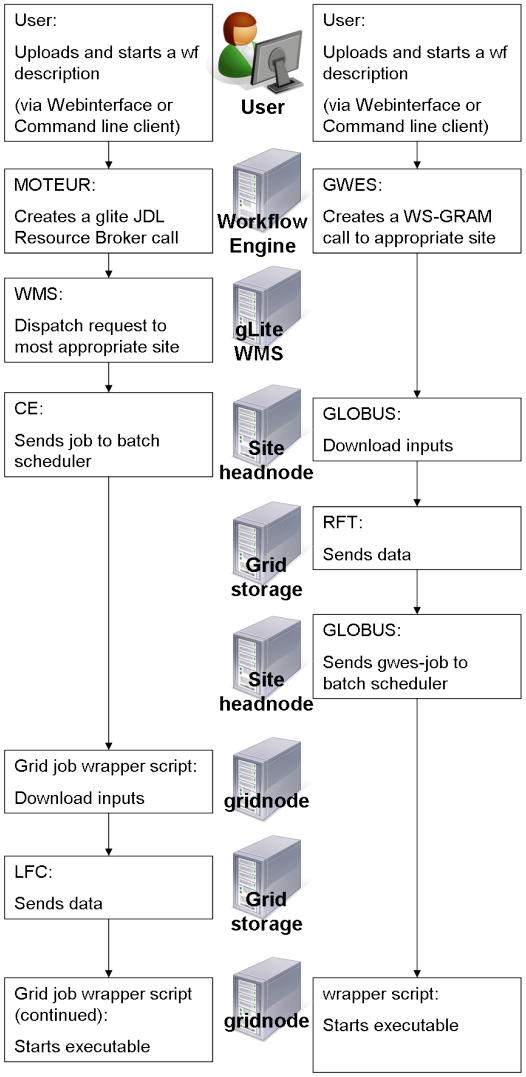

Fig. 2. Flowchart of processes and resources involved during the workflow based submission of a Grid job. *Left:* MOTEUR on EGI. *Right:* GWES on D-Grid.

Comparing the program files involved in the workflow-based Grid application, as shown in Fig. 3, strong similarities between the functions of the different files are found. But the location and the methods to call them are different. Note that the wrapper scripts play a central role here, as they are the interface between the various workflow systems and the underlying software.

In order to make the migration of an application from one workflow system to the other as easy as possible, the modification of a minimum number of files is desired. This can be achieved if a common wrapper script is created. Then the workflow migration requires only translation of the actual workflow and resource descriptions, as depicted in Fig. 4. As the information provided by the gwdl workflow description is spread over the gwendia workflow description and the GASW descriptor, no further reduction of files to be modified can be achieved. While the workflow translation is envisioned to be realized automatically within the course of

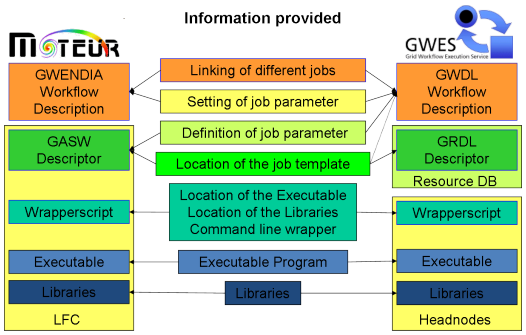

Fig. 3. Comparison of files required for workflow-based applications.

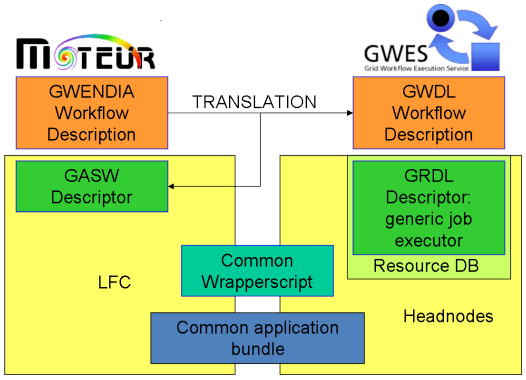

Fig. 4. Scheme of files involved in workflow translation

the SHIWA project, a common underlying application is required to run the translated workflows successfully.

3.3 Results

Workflow Translation For the described simulation task, a common wrapper script could be achieved by moderate modification of the original wrapper script, basically by adding some more input parameters regarding the software location. For this task, the workflow translation has also been done manually. Screenshots of the workflow executed within the different systems are given in Fig. 5 a) and b). The workflows themselves look different, in particular the GWES workflow has more input and output places. This is because the location of the program bundle (release pack) and the wrapper script to be used are set within the GASW descriptor implicitly called by MOTEUR. The additional output places *stdout* and *stderr* have monitoring purposes. The same information can be obtained on the MOTEUR web interface. To have a maximum similarity between the two workflow implementations, the program files and the wrapper script are not deployed to every grid cluster in D-Grid but are stored on a certain location within the Grid and are transferred at runtime.

For the DTI analysis, again the wrapper scripts could be reused with slight modifications, but workflow translation required more effort, as many built-in activities have been used in the original GWES workflow. Screenshots of the workflow executed within the different systems are given in Fig. 5 c) and d).

Workflow execution In the current implementation the user has to invoke the workflows on the different workflow systems. Both workflow systems have own user interfaces. The groups using MOTEUR mainly employ the VBrowser for Workflow execution and monitoring, while GWES is mainly accessed through the MediGrid portal. Both workflow systems

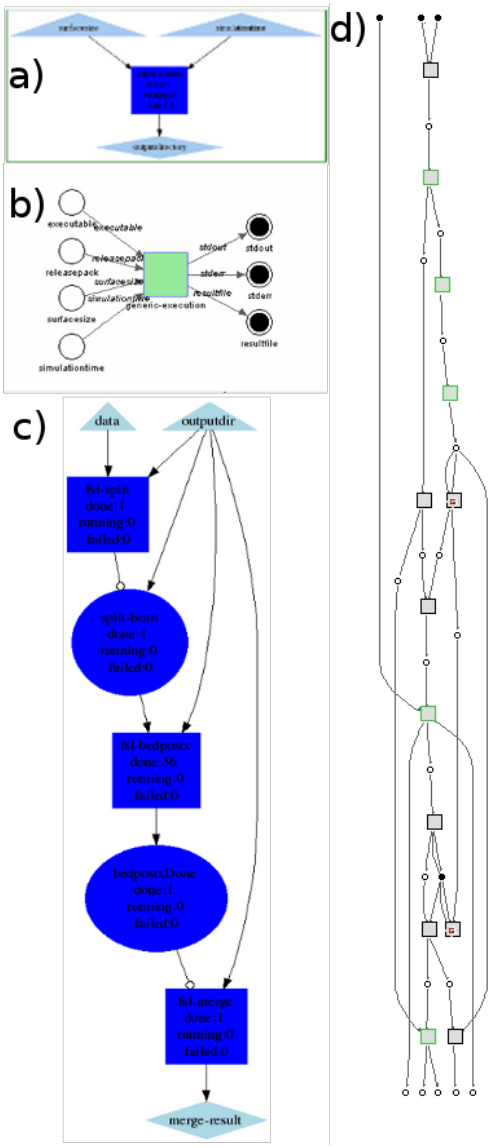

Fig. 5. Screenshots of the translated workflows. CO oxidation simulation as a) MOTEUR workflow on EGI and b) GWES workflow on D-Grid. Diffusion tensor estimation as c) MOTEUR workflow on Dutch-Grid and d) GWES workflow on D-Grid.

come with additional web-based interfaces. Within the SHIWA project, a common portal for workflow execution is under development: the SHIWA simulation platform. Anyhow, researchers generally prefer to use the interfaces they are familiar with. Coarse-grained interoperability would solve this in an elegant way, reducing the effort for the researcher down to define the foreign workflow execution call as a transition within his or her native workflow system. Both systems can be accessed with a webservice, and both systems can call external webservices, enabling coarse-grained interoperability. But the invocation of a MOTEUR workflow on EGI from GWES was not possible, as the respective web service can only be accessed via SSL connection, which is currently not supported by GWES's web service activity. Submission to the MOTEUR instance on the Dutch-Grid didn't succeed neither, presumably due to some data format problems. A MOTEUR workflow on Dutch-Grid could finally invoked through GWES

using a dedicated web service client installed on D-Grid, but it has limited functionality and cannot monitor the submitted workflow. On the other hand, attempts to submit a GWES workflow to D-Grid from the Dutch-Grid MOTEUR instance failed due to AMC's firewall regulations, as the GWES runs on a non-standard http port.

4 DISCUSSION

The presented workflow interoperability analysis shows that workflows can be translated to the other system by only translating the workflow documents themselves, while keeping the underlying application scripts and program files. However, this is only possible because both infrastructures are based on similar or at least compatible Linux versions and hardware architectures. Different hardware or OS may require various binary bundles. Creation of a common wrapper script should be always possible, but it is easier when input parameters are always passed as key-value pairs, as it is required by GWES. It might be difficult to consider all possible environment settings on all systems, so the addition of further input parameters might be necessary. While on D-Grid and Dutch-Grid software packages might be pre-installed on the grid clusters, this is unusual in the EGI's biomed VO. For the CO oxidation application the binaries are about 200 MB. No significant delay due to additional data transfer could be measured on D-Grid. Transferring the program files at runtime significantly eases the software deployment process and guarantees that all clusters use the same software version. But on the other hand it might be more error-prone, as data transfer is known to be a significant source of job failures. The FSL library encompasses 1.4 GB. For employment of the EGI, only tools that are required for the respective task should be sent, which implies a corresponding application analysis. This is not an easy task, because many scripts and executables of FSL have dependencies. More difficult than the plain translation of Grid execution activities is the translation of built-in features of the workflow systems. As already mentioned, GWES allows for XPATH-expressions, while MOTEUR enables Java Beanshells. Here a fine-grained approach might get very complex and coarse-grained interoperability, where only the DCI-related parts of the workflow are invoked on the other workflow system might be the better choice. But also coarse-grained interoperability, which theoretically can be easily accomplished by combining web services, requires certain modifications of the current systems, or has to be realized in a very complicated way with several security and stability flaws. So even if web services are supported from the workflow systems, the shown examples demonstrate that a well defined solution - as to be developed within SHIWA - is necessary to facilitate coarse-grained interoperability.

5 CONCLUSION

Enactment of scientific workflows on other Grid infrastructures could be realized by translation of the workflow to another workflow system, while the application files could be reused unmodified. All DCI specific tasks are then taken over by the respective workflow system. Access to scientific workflows on other Grid infrastructures could be realized by sub-workflow activities, where foreign workflows are submitted to the respective workflow engine's web services. Both solutions require today a lot of manual effort and good knowledge of the involved workflow systems and the DCIs. To reduce this efforts and enable interoperability between further workflow systems and DCIs is the goal of the SHIWA project.

ACKNOWLEDGMENTS

We thank the whole SHIWA consortium, Andreas Hoheisel from Fraunhofer FIRST for the support on GWES, and the support team of the AMC e-BioInfra, Mark Santcroos and Shayan Shahand, for support on the MOTEUR web service.

This work has been supported by the EC within the 7th framework programme under grant agreement no. 261585 SHIWA, the German Federal Ministry of Education and Research (BMBF) within the D-Grid project PneumoGrid (011G09002A-H) and the Dutch NWO within the BiGGrid project.

REFERENCES

[1]M. Alt et al. A grid workflow language using High-Level Petri Nets. In R. W. et al., editor, *Proceedings of the 6-th Intl. Conf. on Parallel Processing and Applied Mathematics PPAM'2005, Poznan, Poland*, volume 3911 of *LNCS*, pages 715–722. Springer-Verlag Berlin Heidelberg, 2006.

[2]A. Berglund et al. Xml path language (xpath) 2.0, w3c recommendation, 2007.

[3]M. M. de Win, G. Jager, J. Booij, L. Reneman, T. Schilt, C. Lavini, S. D. Olabarriaga, G. J. den Heeten, and W. van den Brink. Sustained effects of ecstasy on the human brain: a prospective neuroimaging study in novel users. *Brain*, 131:2936–2945, Nov 2008.

[4]E. Deelman et al. Mapping abstract complex workflows onto grid environments. *Journal of Grid Computing*, 1:9–23, 2003.

[5]T. Glatard et al. Flexible and efficient workflow deployement of data-intensive applications on grids with MOTEUR. *Int. Journal of High Performance Computing and Applications (IJHPCA)*, 22(3):347–360, Aug. 2008.

[6]A. Hoheisel. Grid Workflow Execution Service – dynamic and interactive execution and visualization of distributed workflows. In *Proceedings of the Cracow Grid Workshop 2006, Cracow, Poland*, 2007.

[7]P. Kaira, P. S. Bodega, C. Punckt, H. H. Rotermund, and D. Krefting. Pattern formation in 4:1 resonance of the periodically forced CO oxidation on Pt(110). *Phys Rev E Stat Nonlin Soft Matter Phys*, 77:046106, Apr 2008.

[8]D. Krefting, P. Kaira, and H. H. Rotermund. Period doubling and spatiotemporal chaos in periodically forced CO oxidation on Pt(110). *Phys. Rev. Lett.*, 102:178301, May 2009.

[9]D. Krefting, M. Vossberg, and T. Tolxdorff. Simplified grid implementation of medical image processing algorithms using a workflow managment system. In *Proc. of the Workshop on Medical Imaging on Grids (MICCAI-Grid)*, pages 23–32, 2008.

[10]T. Kukla, T. Kiss, P. Kacsuk, and G. Terstyanszky. A general and scalable solution for heterogeneous workflow invocation and nesting. In *3rd Workshop on Workflows in Support of Large-Scale Scienc*, 2008. In conjunction with SC 2008.

[11]R. Luetzkendorf et al. Enabling of grid based diffusion tensor imaging using a workflow implementation of fsl. In *Healthgrid 2009*, 2009.

[12]J. Montagnat, B. Isnard, T. Glatard, K. Maheshwari, and M. Blay-Fornarino. A data-driven workflow language for grids based on array programming principles. In *Workshop on Workflows in Support of Large-Scale Science(WORKS'09)*, nov 2009.

[13]S. D. Olabarriaga, T. Glatard, A. Hoheisel, A. J. Nederveen, and D. Krefting. Crossing healthgrid borders: Early results in medical imaging. In *Healthgrid 2009*, 2009.

[14]S. D. Olabarriaga, A. J. Nederveen, J. G. Snel, and R. G. Belleman. Towards a virtual laboratory for FMRI data management and analysis. *Stud Health Technol Inform*, 120:43–54, 2006.

[15]J. Rojas Balderrama, J. Montagnat, and D. Lingrand. jGASW: A Service-Oriented Framework Supporting High Throughput Computing and Non-functional Concerns. In *IEEE International Conference on Web Services(ICWS 2010)*, , Miami, FL, USA, July 2010. IEEE Computer Society.