Project Number: **RI-312579**

Project Acronym: **ER-flow**

Project Full Title:

Building an European Research Community through Interoperable Workflows and Data

Theme: **Research Infrastructures**
Call Identifier: **FP7-Infrastructures-2012-1**
Funding Scheme: **Coordination and Support Action**

Deliverable D4.1
Virtual Data Objects specification

Due date of milestone: 31/01/2013
Actual submission date: 30/04/2013
Start date of project: 01/09/2012
Duration: 24 months

Lead Contractor: University of Westminster
Dissemination Level: PU
Version: 1.0

ER-flow is supported by the FP7 Capacities Programme under contract no RI-312579

1 Table of Contents

2 List of Figures and Tables

2.1 Figures

2.2 Tables

3 Status and Change History

Status:	Name:	Date:	Signature:
Draft:	Johan Montagnat	17/04/13	n.n. electronically
Reviewed:	Richard Grunzke	26/04/13	n.n. electronically
Approved:	Gabor Terstyanszky	29/04/13	n.n. electronically

Table 1. Deliverable Status

Version	Date	Pages	Author	Modification
0.1	28/03/13	all	Montagnat	Document layout
0.2	2/04/13	Section 5	Cerezo	Virtual Data introduction
0.3	8/04/13	Sections 6 and 7	Montagnat Cerezo	Data interoperability
0.4	12/04/13	Sections 8 and 9	Montagnat Cerezo Glatard	Specifications
0.5	16/04/13	Section 10	Montagnat	Added conclusion
0.9	17/04/13	All	Montagnat, Cerezo	Revised. Ready for internal review.
1.0	27/04/13	All	Montagnat	Included internal review changes

Table 2. Deliverable Change History

4 Glossary

CGI	Coarse-Grained Interoperability
CNRS	Centre National de la Recherche Scientific (French National Centre for Scientific Research)
DCI	Distributed Computing Infrastructure
DOI	Digital Object Identifier
EGI	European Grid Infrastructure
EMI	European Middleware Initiative
ETL	Extraction Transformation Load
EUDAT	European Data Infrastructure project
FGI	Fine-Grained Interoperability
FITS	Flexible Image Transport System
FMA	Foundational Model of Anatomy
GEMLCA	Grid Execution Management for Legacy Code Applications
IWIR	Interoperable Workflow Intermediate Representation
jSAGA	Java implementation of the SAGA API
LFC	LCG File Catalog
LSDMA	Large Scale Data Management and Analysis project
OAI	Open Archives Initiative
OPM	Open Provenance Model
ORE	Object Exchange and Reuse
OWL	Web Ontology Language
PDB	Protein Data Bank format
PROV	W3C specification for PROVenance on the Web
RDF	Resource Description Framework
RDFS	RDF Vocabulary Description Language
SAGA	Simple API for Grid Applications
SCI-BUS	Scientific gateways-Based User Support project
SRM	Storage Resource Manager
SSP	SHIWA Simulation Platform
SZTAKI	Magyar Tudomanyos Akademia Szamitastechnikai Kutato Intezete
UoW	University of Westminster
URI	Uniform Resource Identifier
URL	Uniform Resource Link
VDO	Virtual Data Object
VO	Virtual Observatory

W3C	World Wide Web Consortium
WP	Work Package

Table 3. Glossary

5 Introduction

5.1 Scientific data

In many scientific areas, improved digital acquisition capabilities combined with online availability of data led to augmented investigation opportunities. All scientific disciplines are witnessing a drastic evolution in the way scientific investigations are conducted, driven by digital data availability and diffusion:

- **Digital data**. Systematic acquisition of scientific data in digital formats combined with the availability of high-speed wide-area networks facilitate data diffusion. The need to uniformly and uniquely identify and access data objects worldwide became a stringent requirement tackled in the context of the *Web of Linked Data*[1].
- **Data volume and complexity**. In the same time, the improved digital data acquisition device capabilities combined with the increased number of data acquisition devices led to the acquisition of large data repositories which analysis is challenging. Scientific databases may contain a wealth of data, and often complex data sets with domain-specific data structures. Challenges related to this huge amount of data are tackled within the *Big Data*[2] research area.
- **In-silico experiments**. Finally, scientific data is also produced by the transformation of source data through computerized analysis steps and workflows, thus enriching existing scientific data repositories. *Scientific Workflows* provide formalized, algorithmic descriptions of the data transformation part in any scientific investigation.

Scientists may sometimes fear to publish data for various reasons [Nelson 09]. However, such reluctances are not only opposed by the emergence of ever-growing research communities and ever-larger scale scientific experiments enforcing the use of data distribution and sharing techniques. There are other strong incentives towards the wide spreading and globalization of scientific data:

- **Open data and translational research**. There is a growing trend to archive data in open databases freely accessible to researchers and more generally to all society with the idea of speeding up time-to-discovery. Open repositories are accessible to a large community, creating an opportunity for unexpected discoveries. Furthermore, open data create new opportunities for correlating complementary data acquired in different contexts. *Translational research*[3] exploits cross-factors analysis among data repositories that were not necessarily designed to be correlated.
- **Reproducibility, reanalysis and results checking**. It has been shown that publishing scientific data together with the scientific analysis methods and the results obtained from this data improves scientific results quality because it facilitates scientific experiment reproduction and results checking [Wich, 11]. Data reanalysis is also possible to draw new conclusions.

Scientific data therefore becomes widespread and available at a global-scale. Scientific disciplines encounter challenging issues related to large and/or complex data sets management, manipulation, sharing and processing which are tackled in modern e-Science platforms. Seminally restricted to the management of raw data stored in distributed files, many e-Science platforms were progressively enriched with complementary metadata annotating the raw data in order to manipulate, interpret and share scientific data (*e.g.* by providing information on the context of raw data acquisition, or provenance information of processed data). This momentum led to the search for new objects aggregating both raw data and all associated information relevant to describe the scientific investigation considered.

[1] Linked Data, http://en.wikipedia.org/wiki/Linked_data
[2] Big Data, http://en.wikipedia.org/wiki/Big_data
[3] Translational research, http://en.wikipedia.org/wiki/Translational_research

In particular, in the context of the myExperiment project[4] aimed at creating a social network of workflow developers and users, the notion of "data pack" was defined as an aggregate of digital scientific data and attached scientific resources (such as workflow(s) used to create or manipulate this data, scientific findings, authors of the scientific experiment, etc) [DRoure, 09]. This concept was later enriched in the Wf4Ever project[5] to create *Research Object* artefacts[6] which are not only data aggregates but also contain semantic annotations facilitating data interpretation and improving data sharing and reuse [Bech, 10a]. Research objects are meant to replace scientific publications by all digital scientific resources involved in a scientific investigation:

> "*A Research Object bundles together essential information relating to experiments and investigations. This includes not only the data used, and methods employed to produce and analyse that data, but also the people involved in the investigation. An association with a dataset (or service, or result collection, or instrument) is now more than just a citation or reference to that dataset (or service or result collection). The association is rather a link to that dataset (or service or result collection) that can be explicitly followed or dereferenced providing access to the actual resource and thus enactment of the service, query or retrieval of data, and so on. In addition a Research Object includes additional semantic information that will organize not just aggregate the resources.*" [Bech, 10a]

Research Objects should be created to make possible sharing and reuse of data, object repurposing, repetition of a study described in the object to reproduce its results, and replay of the experiment to trace and validate all its inner parts. There are several research object stereotypes proposed depending on the kind of information the research object is supposed to represent. It is proposed that the Research Objects implementation is based on Semantic Web standards for annotations, and the *Object Exchange and Reuse* (ORE) specification of the Open Archives Initiative[7], which defines a standard format for file bundles suited to build aggregates [Bech, 10b]. A Research Object specification draft is published[8], but the final Research Object format is not fully available yet.

5.2 Workflow data

Scientific data is usually represented through various means:

- Some structured data is accessible through databases with a specific query interface, especially relational databases.
- Large amounts of raw data are stored into opaque files.
- Metadata, which is increasingly needed to annotate the raw data with content description, complementary information on the data acquisition context, and/or provenance information, is attached to raw data through various means (joint to the raw data files in structured file containers, as separate data files, into relational or RDF[9] databases, or attached to file catalogues).
- Few processing parameters may be specified; either through configuration files or as program command line arguments.

Scientific workflow systems consume and produce data sets composed of parameters (primitive type values) and raw data (usually stored into files). Each workflow system manipulates primitive data types using its system-specific representation. The use of array data structures is common among scientific workflow systems, which aim at processing large amounts of scientific data, but the representation of arrays of values is system-dependent.

[4] myExperiment project, http://www.myexperiment.org
[5] Wf4Ever project, http://www.wf4ever-project.org
[6] Research Object, http://www.wf4ever-project.org/wiki/display/docs/Research+Object+model
[7] Object Exchange and Reuse (ORE) standard, http://www.openarchives.org/ore/
[8] Research Object specification draft: http://wf4ever.github.io/ro/
[9] Resource Description Framework (RDF), http://www.w3.org/TR/rdf-mt/

Workflow systems are usually not considering the content nor the format of raw data stored in files, which are manipulated as opaque entities. It is the case for all workflow systems considered in ER-flow in particular. Workflow systems need to get access to these files though, and each system depends on different file symbolic naming schemes that are most often inherited from the underlying Distributed Computing Infrastructure (DCI) on which the system operates. Most workflow systems are not aware of domain-specific database structures neither their query interface. Data queries are not exposed to the workflow systems: they happen either before workflow execution (to assemble input data sets) or as part of some of the workflow activities, in user business-code. It is the case for all workflow systems considered on ER-flow and the access to parameters stored in domain-specific databases will not be considered in this study.
The focus of this study will therefore be put on the interoperability of data sets composed by files and parameters.

5.3 Virtual data

In a distributed computing context, workflows are composed of two parts [Plank, 11]:

- The **logical part** is the workflow program, described through the workflow language (*e.g.* a DAG description, a workflow script, etc). The logical part refers to a number of external activities that need to be invoked to complete the workflow execution.
- The **concrete part** is the set of programs invoked from within workflow activities. These programs are independent components orchestrated by the workflow logical part. They may be reused in different contexts. Consequently, they are usually not directly described as a part of the workflow. Instead, the workflow logical part references these programs through an invocation specification.

Workflows create a link between source data consumed as input, algorithmic transformation process of data, and resulting data produced as output. A particularity of workflows is that their logical part constitutes a formal description of the data transformation process while their concrete part constitutes an implementation of this process. Research Objects, in particular Workflow-Centric Research Objects[10], aim at aggregating source data, scientific workflow logical parts, and produced data among other things (such as production traces), with the aim of enabling reproducibility and reanalysis of scientific investigations. Practice shows that replaying a workflow execution to reproduce a scientific data product is a complex enterprise though. It requires that all workflow dependencies are resolved, in particular the workflow concrete part, and that the programs orchestrated by the workflow are replayable (which does not involve that they are reproducible, as they may be non-deterministic processes). Research Objects are more focussed towards the scientific investigation description, and therefore they often contain the workflow logical part, and sometimes execution traces, but not necessarily the workflow concrete part. It makes sense that produced data is part of the research object, not only because it may take a long time to generate the data but also because there is no guarantee that resources stored in a research object are sufficient to ensure regenerating this product data.
Conversely, a complete executable workflow description format combining source input data, logical and concrete parts of workflows, and a detailed execution context description could be considered as an alternative to produced data, as it enables regenerating it on-demand. In the SHIWA project[11], workflow description bundles were developed to specify a workflow including both its logical and its concrete parts [Harris, 10], and an execution platform was delivered to make workflows (re-)execution possible. The format adopted for SHIWA bundles is the ORE specification, which defines a standard format for file bundles including packed files and metadata. Such a bundle can be considered as a *Virtual specification for scientific Data Objects*. We will refer to **Virtual Data Objects** (VDOs) in the reminder of this document to mention a workflow execution description precise enough to generate data by executing a

[10] Workflow-centric Research Objects: http://www.wf4ever-project.org/wiki/pages/viewpage.action?pageId=2065079
[11] SHaring Interoperable Workflows for large-scale Applications (SHIWA), http://www.shiwa-workflow.eu

workflow on identified input data. A VDO has a strong relation with the Data Object to which it is a generative process description: the Data Object is a physical instantiation of the VDO that can be seen as a cache on disk of the VDO value. It should be noted that scientific workflows typically consume and produce data sets composed by multiple data entities. A VDO is therefore able to generate a set of data entities, which logical connection lies in the fact that they were produced through a same workflow execution. Compared to the Research Objects, VDOs and their instantiation are two facets of data that may (or may not) be found in research object aggregates. In addition, VDOs do not include all sort of related information such as data analysis conclusions and publications.

6 Data interoperability

ER-flow WP4 aims at studying scientific data interoperability in the context of workflow processing and generation of data. Data Interoperability is a broad area arising from the need of widespread user communities to share and reuse data acquired and stored in different places. In general, several levels of interoperability of different information systems through data may be distinguished (from the highest to the lowest coupling):

- Using open data stores to share data among several information systems. This solution involves that standard data representation models understood by all information systems are in use.
- Using pivot data models to exchange data. The information systems only need to share pivot models specification to transform data back-and-forth between their internal representation and the pivots.
- Constructing intermediate data stores between different data representation domains through Extraction-Transformation-Load (ETL) techniques. This technique may be completely non-invasive for the information systems as third party data generation tools can be used to produce data stores compatible with the information system internal data representations.

A semantic description of data models underlies all techniques mentioned above. Semantic description is explicit and part of the model when using open data stores, so that information systems can manipulate the data with precise recognition of its content and meaning. This is the case for instance when using Semantic Web standard to expose Linked Data over the Internet. The data specification may be more specific to a narrower domain in which fall the concerned information systems though. Similarly, pivot data models bear a precise semantic that is used by information systems to transform data from there internal representation to the pivot and vice versa. Finally, ETL techniques do not expose the information systems to the semantic of data but they make the semantic of data implicit, hidden in the third party transformation tools.

In the scientific domain, data is increasingly distributed and shared at a large-scale as detailed in Section 5. e-Science platforms, used to speed up time-to-discovery, increase the need for data interoperability at a large-scale. ETL techniques can hardly be applied in this context as the number of systems to interoperate may be large (thus requiring for many ad-hoc internal format to internal format converters) and the volume of data manipulated often makes materialization of transformed data stores impractical. The use of pivot models and/or standard data models should be considered instead. In addition, e-Science platforms are tightly coupled to their underlying Distributed Computing Infrastructure (DCI). Interoperating several infrastructures implies **publishing data** (making it known and accessible externally), giving **means of interpretation** (documenting data semantics so that it can be exchanged between different actors), and enabling **data transfers** across different resources. This high-level definition of data interoperability covers many different challenges (*e.g.* the use of interpretable data formats, standard data access protocols, possibly data access control…), some of which are addressed in this document more specifically focussed on ER-flow objectives.

The SHIWA project[12] has set up a multi-workflow systems management environment aiming at making these systems interoperable. It operationalized its concept through the SHIWA Repository[13], which enables executable workflows sharing, and the SHIWA Simulation Platform[14] (SSP), which enables multi-workflows execution over multiple Distributed Computing Infrastructures. However, SHIWA put little emphasis on data interoperability issues. In particular, there is no data repository, no common data specification interface nor means to transfer data across different DCIs within the SSP. As a result, exchanging data

[12] SHIWA project: http://www.shiwa-workflow.eu
[13] SHIWA repository: http://repo.shiwa-workflow.eu/
[14] SHIWA Simulation Platform: http://ssp.shiwa-workflow.eu/

across workflow systems or underlying infrastructures remained difficult, and data interoperability proved to be a substantial showstopper towards workflow interoperability. The aim of this document is therefore to analyze requirements of the supported research communities in scientific data interoperability and their technical implications.

6.1 Data interoperability over the Web

Web technologies have been pioneering challenges related to data interoperability. To ease data exchanges and processing over different servers, the W3C[15] developed many standards related to data interoperability, including data representation (*e.g.* XML and RDF structuring languages), data indexing (*e.g.* URLs and URIs), data transfers (*e.g.* HTTP protocol), and data processing (*e.g.* Web Services), with particular emphasis on text-based data. Many of these technologies are highly relevant for addressing general data interoperability challenges.

Web technologies nowadays also address metadata description and manipulation challenges. The evolution of the *Web of Data* towards a *Semantic Web* led to Linked Data specifications that ease data interlinking and leverage data (re-)usability through unique identification, referencing means, and rich description of data objects.

6.2 Data security

An important particularity of e-Science platforms is that they may manipulate sensitive data for which appropriate access control and privacy preserving rules are needed, while the Web of Linked Data often targets open data sources, putting little emphasis on data protection. On Distributed Computing Infrastructures, data access control mechanisms may often restrict data exchange over different information systems if proper authorization mechanisms are not implemented.

6.3 Scientific data interoperability

In the context of scientific data consumed and produced by scientific workflow systems (data sets), data interoperability issues usually refer to exchanges of data over different workflow systems and the associated underlying Distributed Computing Infrastructure on which the data sets are stored. Data interoperability challenges may arise from:

1. Different **data representations** in use (different data types and data sets specification, different encodings). Data representation applies both to data values stored into files or sent as parameters to workflow systems.
2. Different **file formats** for a same type of data.
3. Different **data storage and indexing** means (files, databases, and even data sets defined as a result of a workflow computation in some cases).
4. Different **data exchange** means (different data transfer protocols, different I/O parameters passing modes).

Each scientific domain is making use of its specific scientific data representation formats. Some formats become widely accepted and facilitate the exchange of data between different domain applications. In ER-flow for instance:

- Computational Chemistry uses the XYZ format coordinates[16] and the Protein Data Bank (PDB) format[17] to three-dimensional structure of molecules for instance.
- Life Sciences uses different image formats, among which DICOM[18], as well as several various radiology metadata formats.

---

[15] World Wide Web Consortium (W3C), http://www.w3.org
[16] XYZ coordinates format, http://en.wikipedia.org/wiki/XYZ_file_format
[17] Protein Data Bank format (PDB) http://en.wikipedia.org/wiki/Protein_Data_Bank_(file_format), http://www.wwpdb.org/documentation/format33/v3.3.html
[18] Digital Image and Communication in Medicine (DICOM), http://en.wikipedia.org/wiki/DICOM, http://dicom.nema.org

- Astronomy & Astrophysics adopted the Flexible Image Transport System (FITS) format[19] for representing astronomical observations.

All the formats cited above aim at exchanging data between domain business applications sharing the same file format. File format converters are used for dealing with different data representation formats. Scientific workflow systems manipulate data files in a domain-agnostic manner, without consideration of the actual content, format, nor coherency of these files. Data files representation are managed in workflows implicitly through format-aware scientific data processing codes or explicitly, through format conversion activities. In both cases, it is up to the workflow designer to consider file format support by each workflow activities. This information is considered too domain-specific and not accessible from the workflow management system. Direct access to data stored in remote databases is possible for specialized workflow activities, but these are also considered as opaque from the workflow engine perspective and not addressed in this study as explained in Section 5.2.
While files formats are not handled in workflow systems, file names are manipulated through their symbolic file identifiers though. Data storage is predominantly organized through file hierarchies in Distributed Computing Infrastructures. Hence, only data file exchanges will be considered in this study. The focus is put on **data representation** (objective 1) and **file exchange** means (objective 2) applicable to Distributed Computing Infrastructures in this document.

6.4 File symbolic referencing and catalogues

While file formats are not handled in workflow systems, files themselves are manipulated through their logical identifiers. In Distributed Computing Infrastructures, scientific data files are stored on possibly heterogeneous storage resources and referenced to through an infrastructure-specific symbolic file name mechanism. File catalogues are used to structure file sets. Existing file catalogues range from simple file system-like file hierarchy views over stored files, to complex databases including file references and rich associated metadata that enable advanced search over file data sets.
The Astronomy community probably has the more elaborated distributed file catalogues. The Virtual Observatory[20] (VO) is an international-scale effort to publicly share astronomy data. It hosts a registry of regional astronomy data registries over which it provides a homogeneous view and data location services. It standardizes data access and data transfer across catalogues. The catalogues are based on relational databases through which advanced data search facilities are provided. Data sets are finally identified through lists of URLs that enable direct access to files through HTTP. In the context of EGI-InsPIRE project Work Package 6, an effort is in progress to create a bridge between the VO catalogues and applications (including workflows). However, this is still a work in progress.
Through the Molecular Simulation Grid portal[21] (MoSGrid), the Computational Chemistry community similarly accesses a repository covering the 3 sub-domains covered (quantum physics, molecular dynamics, and molecular docking). The MoSGrid repository in based on indexed files stored in the XtreemFS file system[22]. These files are directly accessed by the UNICORE DCI. Various chemical file formats are transformed to and from the Molecular Simulation Markup Language (MSML) which acts as the pivot data format to enable seamless executions of applications with different formats in a workflows..
Finally, the Life Sciences community mostly makes use of the LFC file catalogue[23] maintained as part of the European Middleware Initiative (EMI). LFC is a relational catalogue mapping logical file names (URIs) to physical replicas of data files. It has a limited capability for storing user-defined metadata associated to each logical file. It provides a hierarchical

[19] Flexible Image Transport System (FITS), http://en.wikipedia.org/wiki/FITS, http://fits.gsfc.nasa.gov/
[20] Virtual Observatory (VO), http://www.ivoa.net
[21] Molecular Simulation Grid (MoSGrid), https://mosgrid.de
[22] XtreemFS fault-tolerant distributed file system, http://www.xtreemfs.org
[23] LCG File Catalogue (LFC), http://www.eu-emi.eu/products/-/asset_publisher/1gkD/content/lfc-3

view of files stored in virtual folders. Some specific catalogues may also be in use, such as the XNAT data management system[24] used in neuroradiology. A WS-PGRADE wrapper portlet has been implemented for XNAT. It enables input data pre-staging and output data post-staging in the XNAT catalogue.

The specific catalogues in use within each user communities are currently considered too domain-specific for workflow management level, although some initiatives show a growing interest for repository-aware workflow management systems (VO catalogue bridge and XNAT wrapper for WS-PGRADE for instance). All computing infrastructures underlying the workflow systems supported by ER-flow currently use URIs to reference files. URIs are rich identifiers which usually contain target file server identification, file access protocol and server-specific file identification name. These URIs ensure the uniqueness of file names across distributed resources and facilitate their retrieval.

Ensuring file data sets interoperability across multiple workflow systems based on different Distributed Computing Infrastructures usually requires copying files across the different infrastructure data management systems and replacing file symbolic identifiers. File transfers across DCIs is a notoriously difficult problem due to the different file access / transfer protocols in use and the need for a common authentication & authorization framework among the target DCIs [Korkh 11]. Several tools tackle this issue though, such as the jSAGA library[25] or the SCI-BUS Data Bridge service[26]. A secondary concern is the limitation of the current coarse-grained interoperability mechanism implemented in the SSP (with WS-PGRADE master workflow engine), which only supports single file exchange for each sub-workflow I/O. The use of a file archive format, such as the ORE format, is recommended to bundle several files when needed.

6.5 Other data interoperability initiatives

The problem of data interoperability is not new [Kahn, 95] and several initiatives studying various aspects of data interoperability challenges have been conducted. The W3C, driving Web development over the past decades, is undoubtedly the largest and best recognized institution dealing with data interoperability in a very wide context. W3C only produces specifications and recommendations though. The technical implementations of W3C specifications may be heterogeneous and more or less conforming the standards established.

Other projects in the context of distributed computing have focussed more specifically on remote file storage and file transfers. The Globus toolkit[27] is a pioneer and *de facto* widely adopted middleware for distributed computing. In particular, Globus provides a foundational public key-based security infrastructure (GSI), which guarantees interoperability between different systems adopting it at the lowest level. Its wide adoption among various middleware development initiatives it the key to multiple infrastructures interoperability, especially file exchange capabilities. The Globus Toolkit also provides the GridFTP[28] high-performance data transfer protocol, which became a *de facto* standard for file transfers within and across different computing infrastructures. More recently, Globus developed the Globus online data transfer service[29], a third-party service for managing file transfers over any compliant hosts. Globus online currently accounts for more that 12 PB of data transfers over the Internet.

The European Middleware Initiative[30] borrows from the Globus toolkit its foundational security infrastructure and data transfer capabilities and from earlier middleware initiatives (gLite, UNICORE) to build on top distributed file management services, in particular the LFC

[24] Imaging Informatics Software Platform (XNAT), http://xnat.org/
[25] Java Simple API for Grid Applications (jSAGA), http://grid.in2p3.fr/jsaga/
[26] Scientific Gateway-Based User Suport (SCI-BUS), https://www.sci-bus.eu
[27] Globus toolkit, http://www.globus.org/toolkit
[28] GridFTP, http://www.globus.org/toolkit/docs/latest-stable/gridftp/
[29] Globus online, https://www.globusonline.org
[30] Eureopean Middleware Inititive (EMI), http://www.eu-emi.eu

File Catalog, Storage Element interfaces complying to the SRM standard[31], and the GFAL/LCG utilities for files transfer and replication. The Java implementation of the Simple API for Grid Applications[32] (jSAGA) also proposes a plugin-based extensible middleware that includes access to various file catalogues and file transfer using various file transfer protocols in use on Distributed Computing Infrastructures (such as HTTP, SRM, FTP, GridFTP, local files…). The SCI-BUS European project[33], which develops gateway technologies to facilitate access to various distributed computing infrastructures, is currently developing a cross-infrastructure file transfer tool on top of jSAGA.
Recently, the European Data Infrastructure project[34] (EUDAT) was started to tackle the specific challenges of data management in Distributed Computing Infrastructures. It follows a preliminary requirement study identifying the need for a coherent approach to data access and preservation [Koski, 09]. The EUDAT implementation of the challenges identified in the area of distributed data management among various user communities is the delivery of 5 data management-related services, namely:

- Safe Replication: replication of data in selected data centres.
- Dynamic Replication: stage data between EUDAT resources and computing resources.
- Metadata: joint open metadata domain for all data stored by EUDAT centres.
- Simple Store: data upload, storage and sharing.
- AAI: Authentication and Authorization Infrastructure.

EUDAT services are still under specification and development.
Even more recently, the German Large Scale Data Management and Analysis project[35] (LSDMA) was started to develop community-specific Data Life Cycle Laboratories, especially in the “Earth & Environment”, “Energy”, “Health”, and “Structure of Matter” areas. The Data Services Integration Team[36] has the mission to provide solutions for the uniform access to computing and storage resources of the LSDMA participating centres, including access to high performance storage, replication, organisation and archival of data. This work is still in an early development phase.
Many tools and services mentioned above can be considered as a basis for a data interoperability solution. In particular, they tackle the problems of file indexing (through global-scale file catalogues) and cross-infrastructure file transfers (through multi-protocols file access APIs), *i.e.* **objective 2** as identified in the Section 6.3. The problem of cross-infrastructure users authentication and authorization is a cross-concern that received little attention though. More generally, the problem of managing non file-based data sets and interfacing data with workflow systems received very little attention.

[31] Storage Resource Manager (SRM), http://en.wikipedia.org/wiki/Storage_Resource_Manager
[32] Java Simple API for Grid Applications (jSAGA), http://grid.in2p3.fr/jsaga/
[33] Scientific Gateway-Based User Suport (SCI-BUS), https://www.sci-bus.eu
[34] European Data Infrastructure project (EUDAT), http://www.eudat.eu
[35] Large Scale Data Management and Analysis (LSDMA), http://www.helmholtz-lsdma.de
[36] LSDMA Data Services Integration Team (DSIT), http://www.helmholtz-lsdma.de/63.php

7 Data interoperability in ER-flow

7.1 ER-flow user communities and workflow systems usage

The ER-flow project involves four pilot user communities: Computational Chemistry, Life Sciences, Astronomy & Astrophysics, and Heliophysics. In addition, it aims at servicing other user communities making use of distributed computing workflow systems, in particular through collaboration with the EGI.eu organization. The pilot user communities already have experience with some workflow systems, in particular:

- WS-PGRADE and UNICORE workflow engine in Computational Chemistry.
- WS-PGRADE and MOTEUR in Life Sciences. The use of Taverna is also being considered.
- WS-PGRADE in Astronomy & Astrophysics. The use of UNICORE workflow engine for access to HPC resources is considered.
- Taverna in Heliophysics.

Consequently, ER-flow most stringent needs relate to WS-PGRADE, UNICORE workflow engine, Taverna and MOTEUR. Many other workflow engines are supported by the SSP platform (*e.g.* Triana, Askalon, Pegasus…), which could be of interest for other communities as well. Integrating the UNICORE workflow engine is in the ER-flow roadmap.

In ER-flow, data interoperability issues arise from the need to exchange data between:

- Different workflow activities.
- Different workflow management systems.
- Different distributed computing infrastructures, as different workflow systems may operate on different infrastructures.

Data sets may be exchanged between different workflow management systems in two scenarios at least. Firstly, data sets may be used as intermediate data objects, resulting from a (workflow-based) pre-computation and aimed at being post processed by a different workflow. Secondly, in the SSP multiple workflows may be combined in a meta-workflow as the system enabled coarse-grained workflow interoperability.

7.2 Data interoperability in the SSP

The SSP exploited in the ER-flow project is a multi-workflow systems platform operating over different Distributed Computing Infrastructures. Each workflow system consumes and produces data sets using its own data I/O interface, data representation and data access protocols. In ER-flow, the coarse-grained workflow interoperability technology implemented in the SSP allows for the design and execution of meta-workflows, where a master workflow embeds sub-workflows as some of its activities (see Figure 6.1 below). The master workflow receives input data from the SSP user interface and returns output data to the end user through this interface. Intermediate data sets also need to be exchanged between the master workflow system and the embedded workflow systems at sub-workflow input and sub-workflow output.

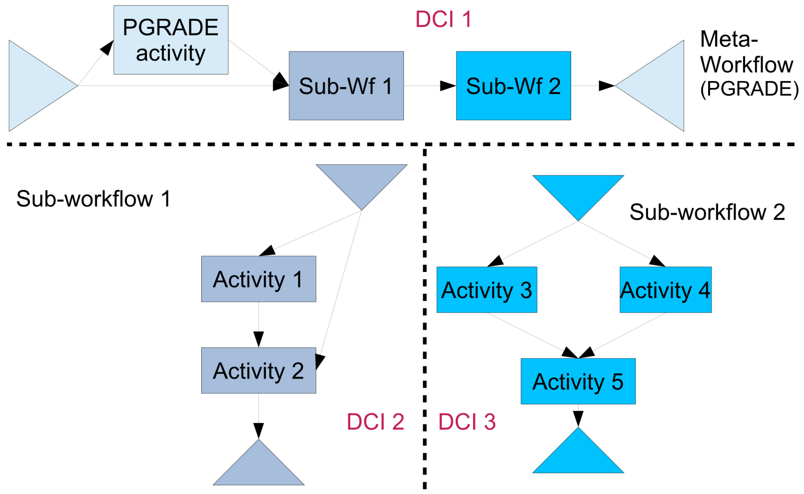

Figure 6.1. A typical meta-workflow as executed in the SSP: a master PGRADE workflow, executing on computing infrastructure DCI1, embeds a native activity and two sub-workflows as sub-activities. A potentially different workflow engine, potentially using a different computing infrastructure (DCI2 or DCI3), executes each sub-workflow. The inputs and outputs of the sub-workflows are chained with the master workflow process. The master workflow input and outputs are stored in files and received from / returned to the SSP user interface.

To execute the meta-workflow illustrated in Figure 6.1, several data exchanges need to be considered (see Figure 6.2 below, where orange arrows show data transfers explicitly). Data received from the user interface ① and data produced by the master workflow ② may be sent to sub-workflows. Data produced by a sub-workflow may be consumed by another sub-workflow ③ or by the master workflow ④.

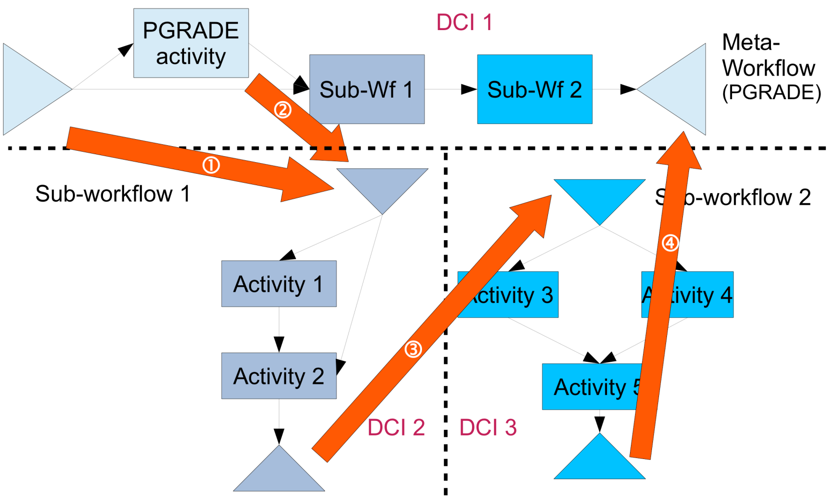

Figure 6.2. Data transfers needed between the master workflow and the embedded sub-workflows. Sub-workflow 1 receives as input a mixture of data from the master workflow input and data produced by the native PGRADE activity. Sub-workflow 2 receives as input data produced by sub-workflow 1. Master workflow receives as output data produced by sub-workflow 2. The data exchanged may be exchanged by direct parameters passing, through a specific data management system (*e.g.* data records in a relational database) or through files. In case of files, transfer across different DCIs may be needed.

Currently, the input data is passed to the master workflow from the graphical interface and the end user retrieves output data from the master workflow as files. A file identifier may be specified for each input port associated to an input activity or each output port associated to an output activity of the master workflow. Similarly, data is exchanged between the master workflow and embedded sub-workflows as file name identifiers (file names sent to the input ports of the activity wrapping a sub-workflow are transferred to the sub-workflow engine and file names generated by sub-workflow engines are mapped to output port of the wrapping activity). There is no explicit management of the file transfers between different infrastructures, nor ability to express non-file parameters. Data is always exchanged as a single file name per input/output port, regardless of the input/output interface of the embedded sub-workflow engines.

Ideally, the data interoperability mechanism should ensure:

- Interoperable data representation (**objective 1** in Section 6.3):
  - Transformation of non-file parameters taking into account different data representations that may be in use.
  - Adaptation of input / output data sets description to the workflow engine data interface.
- File exchange (**objective 2** in Section 6.3):
  - Input / Output data transfers between different workflow engines / DCIs, whether data is stored into files or represented as non-file parameters.
  - Transfer of Input / Output files across DCIs, taking into account the discrepancies between file access control systems and file transfer protocols that may exist.
  - Transfer of Input / Output files between the platform and external machines (*e.g.* the user's machine).

The current SSP platform does not manage workflow data sets representation. It only enables the passing of a single file for each master workflow input and output port, and for each sub-workflow input and output. It does not deal with the different sub-systems I/O representation discrepancies, letting to the workflow designer the task of transforming data when needed. It does not recognize non-files parameters nor the structure of data sets. It does not deliver a cross-infrastructure file transfer service either. The remainder of this document provide recommendations regarding the way to achieve this data interoperability level and study the impact on the execution platform.

8 Workflow data pivot format

As explained above, the use of a pivot data representation is strongly encouraged to solve the data representation challenge of inter-workflow data sets interoperability. This representation should be accompanied with a standardized in-file representation (serialization and deserialization process) to ease data exchanges across different systems and DCIs.

The pivot data representation should enable the description of workflow data sets, including primitive parameter values (numerical values, text strings…) and symbolic file identifiers. Scientific workflows usually manipulate large data sets made of lists or arrays of data values or files. The construction of such data sets should therefore be supported.

8.1 Primitive data types

Definitions of a wide variety of primitive data types as well as constructs to create complex data structures are standardized in the W3C XSD Datatypes document[37]. Built-in data types include numerical values, character strings, dates, times, and binary values among others. It is recommended to follow this standard for primitive data types representation. Yet, scientific workflows usually only make use of a subset of the primitive data types included in the W3C standard. The Interoperable Workflow Intermediate Representation (IWIR) defined in the context of the SHIWA project is a pivot workflow language only considering the following primitive data types for instance: booleans, integers, doubles, strings and file identifiers. Also, all file management systems in DCIs accessible from the SSP are using URIs for file identification. It is therefore recommended to restrict the XSD primitive data types to those six:

- boolean, with two-values space {true, false}.
- long, a mathematical integer number between -9223372036854775808 and 9223372036854775807.
- double, an IEEE double-precision 64-bit floating point datatype[38]
- string, an XML character string.
- anyURI, an International Resources Identifier Referent (IRI) used to identify a file (local or remote).
- base64Binary, arbitrary binary data encoded using the Base64 Encoding[39].

8.2 Data arrays

Homogeneous arrays of either primitive type atomic values (simple arrays) or sub-arrays (nested arrays) are needed to represent workflow data sets. Arrays may be defined in XSD using the Complex Type Definition Schema Component and restricting the underlying sequence of items to all have the same data type (note that List data types, deriving from the Simple Type Definition Schema, could be used to represent arrays of primitive types but not nested arrays).

8.3 Mapping primitive data types to workflow inputs and outputs

Scientific workflows typically consume and produce multiple inputs and outputs (also named “sources”/”sinks” or “workflow input/output ports” depending on the workflow system considered). The input data sets, described as arrays of primitive types, need to be mapped to the corresponding workflow inputs. Similarly, the data arrays produced by workflow execution need to be associated to the corresponding workflow output for further use.

[37] XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, http://www.w3.org/TR/xmlschema11-2/

[38] IEEE double-precision floating point, IEEE standard 754-2008, http://dx.doi.org/10.1109/IEEESTD.2008.4610935

[39] Base64 Data Encoding, http://www.ietf.org/rfc/rfc3548.txt

Hence, a common representation for mapping input and output data sets to workflow inputs and outputs is needed.

8.4 Pivot file representation

Workflow data is exchanged within the ER-flow platform through files. A pivot file specification is needed to enable describing workflow data to be consumed as input or data produced as a workflow output.
Using XSD makes primitive data sets description in pivot file standard, conforming to the XSD schema and using XML serialization. The mapping of data sets to the workflow input or output interface should be represented using the same XML representation. Workflow input and outputs are typically identified by names (encoded as XSD strings). Two XML tags containing the values to be mapped as workflow inputs or output and containing the corresponding values to be consumed or produced are sufficient:

```
<input name="source name" type="primitive type"> … </input>
<output name="sink name" type="primitive type"> … </output>
```

The input (respectively output) tag is parameterized by a `type` attribute specifying the primitive type of data produced or consumed by the corresponding workflow input (respectively output). The value of the `type` attribute is one of the 6 recognized XSD types. From the valued data structure described inside the input (respectively output) tag, the array structure may be inferred: the input (respectively output) may contain a single value (scalar case), an array of values, or an arbitrary number of nested arrays.

8.5 Pivot files manipulation in the SSP

A pivot file containing an arbitrary number of input tags may be consumed by a workflow if it contains exactly the same number of named input as the workflow input interface. Conversely, a pivot file produced by a workflow will always contain exactly the same number of names output as the workflow output interface. Pivot files may be manipulated by the ER-flow platform (sliced and/or merged) depending on the meta-workflow specification, as illustrated in Figure 8.1 below.

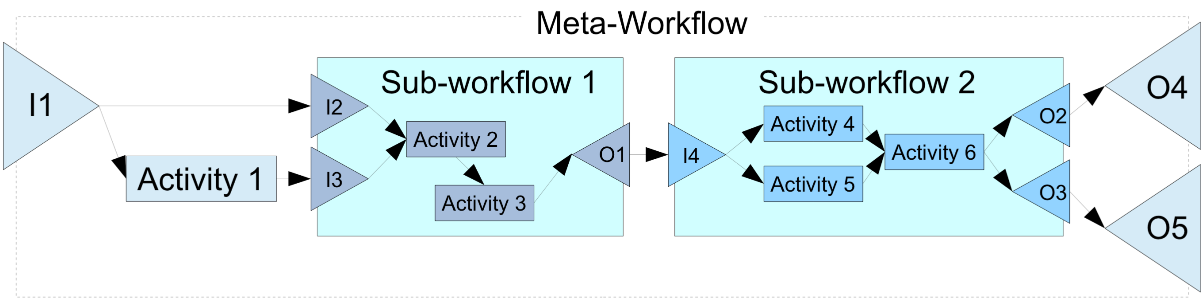

Figure 8.1. Pivot files manipulation within the ER-flow platform.

The master workflow specifies the data exchange between several sub-workflows. The meta-workflow illustrated in Figure 8.1 has one input (I1) and two outputs (O4 and O5). It contains two sub-workflows. Sub-workflow 1 has 2 inputs (corresponding to two input ports of the meta-workflow activity) named I2 and I3, and one output (output port of the meta-workflow activity) named O1. Sub-workflow 2 similarly has one input (I4) and two outputs (O2 and O3). The meta-workflow structure describes how data is exchanged between workflows: Sub-workflow 1 receives data in I2 coming from I1 and data in I3 coming from Activity 1. It produces data in O1 that is transferred to input I4 in Sub-workflow 2. Sub-workflow 2 produces data in O2 finally transferred to master workflow O4 and data in O3 transferred to O5. In this example it can be seen that Sub-workflow 1 input is composed by data coming from two branches of the master workflow, which need to be merged. The corresponding input file will contain two input tags which values are provided by different

activities of the meta-workflow. Conversely, Sub-workflow 2 produces two outputs that will be sent to different meta-workflow outputs. Its output file will be sliced into two independent outputs to be piped in the meta-workflow process upon Sub-workflow 2 completion. The intermediate pivot file transferring data from O1 to I4 do not require changes (except that Sub-workflow 1 output becomes Sub-workflow 2 input). Any type of input/output file slicing and merging can be needed, depending on the master workflow graph structure.
It should also be noted that this intermediate file enforces data types compatibility: a workflow output can be piped into another workflow input only if data types of the corresponding input/output match. Similarly, the nesting level of arrays produced and consumed by workflow should match. A mismatch will produce a platform runtime error.

8.6 Files indexing scheme and manipulation

Pivot data sets are meant to be used as intermediate files between multiple workflow invocations. In some cases (*e.g.* meta-workflow) these files are only transferred by the SSP between different workflow engines for invocation. In other cases (*e.g.* master workflow output), the result should be stored for later reuse. Pivot data sets should therefore be accessible uniformly regardless of the storage technology used in the various Distributed Computing Infrastructures underlying their production. A unique logical identifier should be associated to each pivot file and a service should implement pivot file manipulation (listing, upload, download, deletion, access rights control).
It is recommended that the pivot file manipulation service implements a de-referencing scheme similar to the Digital Object Identifiers (DOIs) used for scientific publications, and standard protocols for physical files identification (*i.e.* URLs) and transfer (*i.e.* HTTP). Pivot files can then be identified through cross-platform unique logical identifiers and accessed using commonly adopted file transfer mechanisms. The pivot file management service should expose a simple HTTP-based API for accessing all functionalities programmatically.
In addition, pivot files may refer to scientific data files that are part of workflow inputs and outputs. The file indexing scheme is also useful to name the files referenced, as anyURI XSD data types. The use of a cross-DCI file indexing scheme makes pivot files valid, independently of the target computing infrastructure. Yet, it should be noted that when transferring a pivot file across different DCIs for meta-workflows execution, transferring the referenced files might also be needed.

8.7 Pivot input data files and VDOs

Input data files conforming to the pivot file format defined in this section may be used as input of any workflow to be executed on the ER-flow platform. Such files define the input parameters required for triggering workflow executions. Together with the workflow logical and concrete parts, they specify Virtual Data Objects. Using an archive format such as ORE, it is possible to bundle all these elements together, to specify a VDO through a single archive file identified in the ER-flow platform through the same indexing scheme as the one used to specify pivot data files.
VDOs may either refer to an execution specification archive, or be instantiated as a set of data items produced by workflow execution (workflow output pivot file). Consequently, the de-referencing scheme should be able to identify which type of data object is available (execution archive and/or physical file instances) and return the corresponding file(s) to the requester. Depending on the context, the data volume, and the time needed to generate data through workflow execution, VDOs can be physically generated and cached on disk, or conversely generated on-demand. Since a single physical data artefact may be replicated several times on different storage spaces in a distributed system for performance and reliability reasons, a VDO may reference several instantiated file sets, or even an archive description and physical instances.
Note that VDOs may play an important role in case of re-execution of a meta-workflow after failure. When a meta-workflow is partially executed, some intermediate results may be

generated by sub-workflows and store in the SSP as VDO files. Upon re-execution of the meta-workflow, searching for existing VDO files may avoid useless re-execution of part of the sub-workflows.

9 Implementation of data interoperability in the SSP

The implementation of the concepts and specifications described in the previous section has an impact on several components of the SSP infrastructure that are discussed below.

9.1 SSP architecture

The following components of the SSP are involved in the management of workflow data:

- The SSP portal is used for designing master workflows and configuring executable workflows. In particular, it includes:
  - The PGRADE workflow designer through which master workflow input and output ports are defined;
  - The executable workflow configuration interface through which files are associated to inputs and outputs; and
  - A proxy manager through which user credentials are uploaded.
- A MyProxy server is used to store medium-lived user X509 proxies.
- Various workflow management systems that can be used for sub-workflows execution are embedded. The GEMLCA[40] legacy application wrapper is used as a common invocation interface between the PGRADE master engine and embedded workflow systems.

As outlined in the previous Section, a data interoperability solution requires using a common data representation and enabling cross-DCI file transfers. The common data representation management will at least have an impact on:

- The SSP executable workflow configuration interface, so that non-file parameters and/or multiple I/O files may be specified. If the current interface based on a single exchange file is kept, it could be use to specify a meta-file containing all information on parameters and files which constitute the data set though. Similarly, the PGRADE workflow designer may be impacted in case a data set composed with multiple files is mapped to a single workflow activity port. To preserve the current designer, a single meta-file could be used. Meta-file specification may follow the Object Exchange and Reuse (ORE) specification defined by the Open Archives Initiative[41], which defines a standard format for file bundles.
- The I/O interface of all workflow system embedded in the platform (including the PGRADE master system), so that I/O data sets can be exchange between different systems. Potentially, this will have an impact on the GEMLCA wrapper.

In addition, cross-DCI transfer is not available in the SSP. A dedicated service will be needed to deliver this functionality (e.g. one of those identified in Section 1.3). This service needs to be synchronized with the platform proxy management system, potentially requiring an adaptation of this component.

9.2 Adapters

The data sets that workflows consume as inputs or produce as outputs are composed of parameters (*e.g.* simple values such as an integer, etc) and/or data files. Data files are identified through symbolic file identifiers (*e.g.* URIs), which can be considered as textual parameter values. All I/Os will therefore be considered as parameters in the remainder.

The workflow systems embedded in the SSP all have a specific interface to describe their I/O parameters. They may use:

- Input and/or Output parameters described on the command line;
- Input (resp. Output) parameters in system-specific Input (resp. Output) files;

[40] Grid Execution Management for Legacy Code Applications (GEMLCA), http://www.cpc.wmin.ac.uk/cpcsite/index.php/Gemlca

[41] Object Exchange and Reuse (ORE) standard, http://www.openarchives.org/ore/

- Input parameters read from the process standard input and/or Output parameters written to the process standard output stream;
- Or any combination of the above.

Invoking a workflow engine therefore requires adapting to its specific interface. Given that any workflow system may exchange data with any other in the SSP (see Figure 6.2), the number of adaptors needed grows as the square of the number of workflow engines supported. Alternatively, the use of a common data sets description format, known from all embedded systems, reduces the number of adaptors to be developed significantly, as it can be used as a pivot representation and a single pair of adaptors (two-ways conversion between the workflow internal format and the pivot format) is then sufficient for each workflow system. It should also be noted that the adaptors might be integrated:

- Invasively, by modifying the workflow systems to make them aware of this pivot format; or
- Non-invasively, by developing a two-ways wrapper that receives an input pivot format file describing the workflow input data set, adapts it to the native workflow invocation interface, retrieves the workflow output in its native format, and convert the result in an output pivot format file.

Given the nature of the SSP platform, which aims at facilitating the integration and exploitation of existing workflow engines, a non-invasive approach is preferred. This does not prevent some workflow systems to adopt the pivot representation in their code base though. A single invocation interface recommendation is preferable in this latter case.

The master workflow system plays a specific role as it triggers the invocations of embedded workflow systems. It should adopt the pivot data representation itself to ease data exchanges, and it should also make sure that intermediate pivot I/O files are transferred between itself and the workflow systems embedded. In the context of the SSP, the GEMLCA wrapper that shields the master system from the idiosyncrasies of embedded systems can be used for handling these I/O files.

The I/O pivot file may reference data files stored in different DCIs. It might be needed to copy the files referenced from one DCI to another. File transfers may be handled either non-invasively by a third party service, or be integrated invasively in the embedded workflow (augmenting the target workflow with data transfer activities). The former, non-invasive solution is preferable. The availability of cross-DCI data transfer tools (see Section 6.5) should help in its implementation. It should be noted that cross-DCI transfers require proper management of user credentials over both the source and the target DCI: the user requesting files transfer should be both recognized on these two DCIs and authorized to access the corresponding files. This potentially implies the management of multiple credentials per users, or the use of robot certificates if these are accepted by the DCI usage policies.

9.3 Pivot files manipulation service

As explained in Section 8.6, a file manipulation service is needed to properly and easily manage files stored on multiple DCIs. This service should offer:

- A multi-DCI file indexing and de-referencing mechanism.
- A cross-DCI data transfer facility.

File indexing will make it possible to identify any file stored on one of the supported DCIs through a platform-wide unique URI. The URIs generated by the file manipulation service non-ambiguously identify scientific data files and can be used as logical file identifiers to name referenced files (anyURI data type) in pivot files for instance.

File de-referencing addresses several requirements:

- Discover the DCI-specific file identifier.
- Generate a transfer URL when files should be retrieved through the standard HTTP protocol.

- Map a VDO either to its corresponding execution specification archive or to one of its physical instantiations (set of files stored on disk).

Cross-DCI file transfer is needed both by the SSP when transferring data across workflow systems and by end user when uploading input data sets to the platform or retrieving data produced by workflow runs. Cross-platform data transfer tools such as jSAGA or the SCI-BUS DCI Bridge should be considered for implementing this functionality. A cross platform data transfer API is needed to access the service programmatically, *e.g.* by the SSP to transfer files referenced in pivot data files across DCIs during meta-workflows execution. An easily accessible HTTP interface is also needed for end users to access file upload / download functionality. It should enable file transfer through the HTTP protocol, by providing an intermediate storage space where data may be sent by HTTP, or retrieved from the connected DCI for HTTP transport towards the end user. In that case, the file index de-referencing functionality can be invoked to transfer file from DCIs to the intermediate storage space and return an accessible URL.

10 Conclusions

10.1 Summary

Data interoperability is a broad concept covering data publication, data interpretation, and cross-platforms data transfers. Scientific data interoperability challenges may arise from different data representations in use, different file formats for a same type of data, different data storage and indexing means and different data exchange means. In the context of ER-flow the focus is more specifically put on workflow parameter files representation and transfer across DCIs.
It is recommended to:

- Follow the W3C XSD specifications and use a dedicated XML schema introducing workflow input and output concepts to deliver a pivot format representation.
- Explore existing multi-DCI file transfer tools such as jSAGA or the SCI-BUS Data Bridge to support cross-DCI data exchange.
- Provide a file index and manipulation service, universally accessible through a standard HTTP API. This service shall include a de-referencing layer enabling the identification of files stored on any supported DCI, and the implementation of Virtual Data Objects.

Enriching the SSP with data interoperability would then imply to:

- Write adapters for each supported workflow system to map pivot data sets to the system-specific data interface;
- Make use of these adapters through the sub-workflow system invocation wrapper;
- Integrate the cross-DCI file manipulation service; and
- Potentially make use of an archive format such as ORE to adapt to the current PGRADE portal and workflow engines, which implement communication with embedded workflow systems through a single file (otherwise the PGRADE engine interface and the SSP I/O specification GUI need to be updated accordingly).

10.2 Future work

Data semantic description is considered as a key towards data interoperability nowadays. Data semantic technology arose from the emergence of the *Semantic Web*[42] and the need to ease data interlinking and interpretation. In this context the *World Wide Web Consortium* (W3C) defined multiple standards to explicit the semantics of data, including vocabulary definition languages (e.g. RDFS[43] and OWL[44]) and semantic annotation formats (e.g. RDF). The formal definition of data semantics through a vocabulary facilitates the alignment of heterogeneous data sources onto a shared reference. It thus primary addresses the challenge of data interoperability at the level of data representations. In addition, e-Science platforms increasingly use semantic description resources to link the semantics of computations (as contained in workflow programs) and the semantics of data, *e.g.* through the production of provenance traces upon data generation. Semantic information therefore becomes a vector to facilitate data reuse and data generation reproducibility.
Among the pilot scientific communities involved in ER-flow, there is only a limited use of semantic technologies so far. Metadata may be extensively used to describe and make data searchable, *e.g.* in the Virtual Observatory catalogues, but there are no widely accepted reference semantic vocabulary, neither in Astronomy nor in Computation Chemistry where pivot format files (PDB, FITS) are widely adopted as data sharing means. In Life Sciences though, and most particularly in medicine and in radiology, many ontologies have been developed. Some *de facto* standards emerged such as the XNAT data model or the

---

[42] Semantic Web, http://www.w3.org/standards/semanticweb/
[43] RDF Vocabulary Description Language (RDFS), http://www.w3.org/TR/rdf-schema/
[44] Web Ontology Language (OWL), http://www.w3.org/standards/techs/owl

Foundational Model of Anatomy[45] (FMA). The use of semantic technology goes beyond data description and is used *e.g.* for data provenance (using OPM[46] or PROV[47] models).
The impact of semantic technologies will be studied in depth in the future of WP4 activity within ER-flow (Task 4.4).

[45] Foundational Model of Anatomy (FMA), http://sig.biostr.washington.edu/projects/fm/
[46] Open Provenance Model (OPM), http://openprovenance.org
[47] PROV specification for provenance on the Web, http://www.w3.org/TR/prov-primer/

References

[Bech, 10a] Bechhofer S., De Roure D., Gamble M., Goble C. and Buchan I., “Research Objects: Towards Exchange and Reuse of Digital Knowledge”. In: The Future of the Web for Collaborative Science (FWCS 2010), April 2010, Raleigh, NC, USA.

[Bech, 10b] Bechhofer S., Ainsworth J., Bhagat J., Buchan I., Couch P., Cruickshank D., Delderfield M., Dunlop I., Gamble M., Goble C., Michaelides D., Missier P., Owen S., Newman D., De Roure D. and Sufi S., “Why Linked Data is Not Enough for Scientists”. Sixth IEEE e-Science conference, December 2010, Brisbane, Australia.

[DRoure 09] D. De Roure, C. Goble, and R. Stevens. “The Design and Realisation of the myExperiment Virtual Research Environment for Social Sharing of Workflows”. Future Generation Computer Systems, 25:561–567, 2009.

[Harris, 10] A. Harrison, D. Rogers and I. Taylor, “SHIWA Desktop”, SHIWA deliverable D5.2, 2010. http://www.shiwa-workflow.eu/documents/10753/626f809c-7853-40ce-a3b2-eb41a29a9ecd

[Kahn, 95] Kahn R. and Wilensky R., “Digital Object Architecture”, Technical Report cnri-dlib/tn95-01, Corporation for National Research Initiatives, 1995. http://www.cnri.reston.va.us/k-w.html.

[Korkh, 11] Korkohv V., Krefting D., Kukla T., Terstyanszky G., Caan M. and Olabarriaga S., “Exploring workflow interoperability tools for neuroimaging data analysis”, Proceedings of the 6th workshop on Workflows in support of large-scale science (WORKS’11), Seattle, USA, November 2011, pp 87-96, ACM. http://dx.doi.org/10.1145/2110497.2110508.

[Koski, 09] Koski K., Gheller C., Heinzel S., Kennedy A., Streit A., Wittenburg P. “Strategy for a European Data Infrastructure”. White Paper, 2009. http://www.csc.fi/english/pages/parade.

[Nelson, 09] B. Nelson. “Data sharing: Empty archives”. Nature, 461(7261):160–3, 2009. http://www.nature.com/news/2009/090909/full/461160a.html.

[Plank, 11] Plankensteiner K., et al., “IWIR Specification”, SHIWA deliverable D6.1, 2012. https://www.shiwa-workflow.eu/documents/10753/55350/IWIR+v1.1+Specification.

[Wich, 11] Wicherts JM, Bakker M, Molenaar D, “Willingness to Share Research Data Is Related to the Strength of the Evidence and the Quality of Reporting of Statistical Results”. PLoS ONE 6(11), 2011. doi:10.1371/journal.pone.0026828