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.*3*

Study of domain semantic data and workflow description

Due date of deliverable: 30/08/2014 Actual submission date: 30/08/2014

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

3 Status and Change History

Status:	**Name:**	**Date:**	**Signature:**
Draft:	Johan Montagnat	21/06/2014	n.n. electronically
Reviewed:	Tamas Kiss	12/08/2014	n.n. electronically
Approved:	Gabor Terstyanszky	15/08/2014	n.n. electronically

Table 1. Deliverable Status

Version	**Date**	**Pages**	**Author**	**Modification**
0.1	04/11/13	All	N. Cerezo	Creation from template and detailed outline
0.2	27/11/13	Section 5	N. Cerezo	Introduction
0.3	30/11/13	Section 5	J. Montagnat, N. Cerezo	Revised introduction and outline
0.4	16/03/14	Section 6	N. Cerezo	State of the Art – Workflow descriptions
0.5	30/03/14	Section 7	N. Cerezo	Recommendations – Workflow descriptions
0.6	28/05/14	Sections 5 to 7	J. Montagnat	Completed sections
0.7	11/06/14	Sections 6 and 7	A. Benabdelkader, S. Olabarriaga	Added “provenance” sections
0.8	17/06/14	All	J. Montagnat	Final edits before internal review

Table 2. Deliverable Change History

4 Glossary

CGI	Coarse-Grained Interoperability
DCI	Distributed Computing Infrastructure
EGI	European Grid Infrastructure
FGI	Fine-Grained Interoperability
FITS	Flexible Image Transport System
FMA	Foundational Model of Anatomy
IVOA	International Virtual Observatory Alliance
LOINC	Logical Observations Identifiers Names and Codes
MoSGrid	Molecular Simulation Grid
MSML	Molecular Simulation Markup Language
NIFSTD	Neuroscience Information Framework Standard Ontologies
OPM	Open Provenance Model
OWL	Web Ontology Language
PROV	W3C specification for PROVenance on the Web
RDF	Resource Description Framework
RDFS	RDF Vocabulary Description Language
SNOMED-CT	Systematized Nomenclature of MEDicine - Clinical Terms
SSP	SHIWA Simulation Platform
SPARQL	SPARQL Protocol and RDF Query Language
UCD	Unified Content Descriptor
URI	Uniform Resource Identifier
URL	Uniform Resource Link
VObs	Virtual Observatory
W3C	World Wide Web Consortium
WP	Work Package
XML	Extensible Markup Language

Table 3. Glossary

5 Introduction

The ER-flow project aims to build a European research community by leveraging and extending the features of the SHIWA Simulation Platform[1] (SSP). Our goal is to study two distinct but somewhat intertwined subjects:

- How **semantic data** could help tackle **scientific workflow interoperability**.
- How **workflow descriptions** impact **scientific workflow interoperability.**

The three main motivations calling for scientific workflow interoperability are:

- **cross-system collaboration**: when two different scientific communities, who use different frameworks, collaborate;
- **cross-system preservation**: when a community switches to a different framework and wants to keep using old workflows; or
- **cross-infrastructure usage**: when a community needs to access resources which are not supported by their framework of choice.

The SHIWA project[2], which preceded ER-flow and resulted in the creation of the SSP, identified two types of scientific workflow interoperability:

Coarse-grained interoperability is achieved when a workflow framework is able to run non-native sub-workflows as black boxes, as illustrated on Figure 1. The SSP implements it by packaging non-native sub-workflows with the corresponding non-native workflow enactor into a WS-PGRADE [1] *job* and composing them in WS-PGRADE *meta-workflows*.

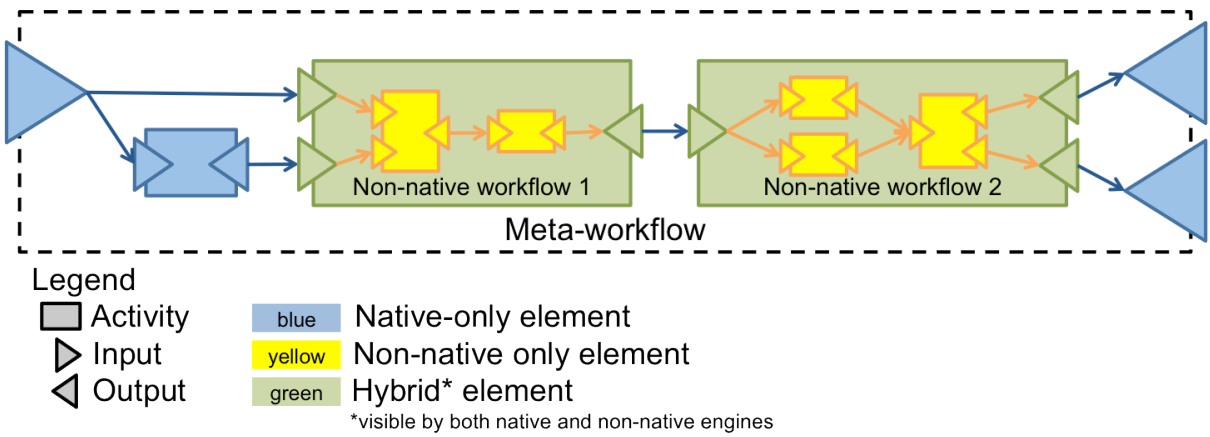

Figure 1. Coarse-Grained Interoperability

Fine-grained interoperability is achieved when a workflow framework is able to interpret non-native workflow languages, either directly or through translations, as illustrated on Figure 2. The SHIWA project designed IWIR [2] as a pivot language which compatible frameworks can interpret and export to.

[1] SHIWA Simulation Platform: http://ssp.shiwa-workflow.eu

[2] SHIWA Project: http://www.shiwa-workflow.eu/project

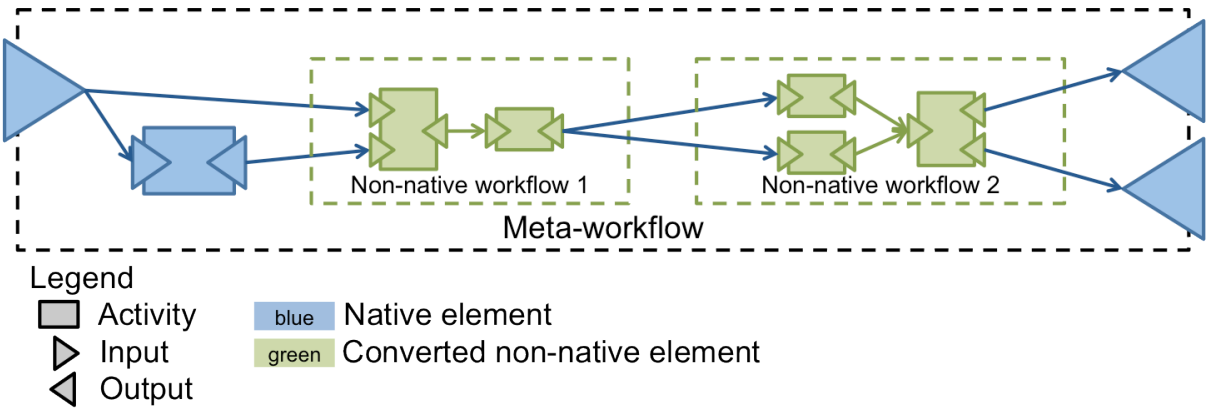

Figure 2. Fine-Grained Interoperability

Both **coarse-grained interoperability** and **fine-grained interoperability** imply solving **data interoperability issues**. Indeed, each scientific workflow framework has its own built-in data types and its own ways to store, retrieve, transfer and, in some cases, visualize data. Some of those specificities, like data types, are tied only to the scientific workflow language. Others, like data retrieval, are also tied to the target Distributed Computing Infrastructure (DCI). Manually creating all the related conversion and data management activities in a meta-workflow is a tedious and error-prone task, which significantly raises the entry barrier for meta-workflow designers. The first deliverable of WP4 (D4.1) specifically studied data interoperability issues arising across scientific workflow frameworks, while its first milestone (MS4.1) marked a proposal for a related data transfer service specification.

Since it does not rely on black boxes, **fine-grained interoperability** must also manage **workflow description discrepancies**, which may, in some cases, amount to format conversion, but generally involve complex model transformations.

5.1 Semantic Data

5.1.1 Raw Data

Data is the first-class citizen in most simulations. Many scientific research communities, notably the astronomical community, even refer to workflows as *pipelines*, thereby emphasizing their functions of data transformation and data transfer.

For a workflow to run and perform meaningful transformations, the operations performed on data must be compatible with the **data type** – i.e. how data is formatted – as well as the **data nature** – i.e. what data represents. Indeed, mismatches in data type (e.g. if an operation expecting an image file is given a compressed archive) will most often break the workflow, whereas mismatches in data nature (e.g. if an algorithm tailored for brain models is given a heart model) will most likely produce meaningless results.

Most data formats carry information about their associated **data type** (e.g. a JPEG file has a header specifying it is a compressed image file among many other things). However, automated conversion between identified data types is not always possible – how does one convert an image into an integer? – and can lead to loss of data, through compression, resizing or truncation.

Information about **data nature** is semantic by definition, since it relates to the meaning of data. Any data carrying such information can be construed as **semantic data**. It is fairly

common, though, to reserve the term for data formats designed to handle semantic information and enable its automated processing. Indeed, while any bundle composed of a piece of core data and a textual description of what that data represents is technically semantic data, it is rarely thought of as such, because it is impractical to handle (e.g. the description and core data could easily be separated) and analyze automatically.

Conflicts of type and nature must already be dealt with in a single scientific workflow framework. Interoperability does not create those issues, but it further complicates them, since different frameworks deal with them in different ways, making it even harder to identify and fix conflicts, whether manually or automatically.

5.1.2 Metadata

As outlined in D4.1, scientific workflow environments do not only manipulate the raw data (usually stored in large files and transferred for DCI for computing) but also an increasing amount of complementary metadata which provides information on raw data content, data description format, acquisition context, traceability, computational parameters, etc. In a multi-workflow environment system such as the SSP, the data interoperability problem extends to this metadata. Metadata description is usually more structured than raw data, which helps in its manipulation and interpretation, but there remain an extensive number of proprietary metadata formats in use.

An important type of metadata for e-scientists using workflow is data traceability information. Linking source data, processed data, computational processes, and computation parameters are important for the scientific analysis of data generated in the context of workflow execution. Hence, provenance metadata has attracted the attention of many scientific workflow system designers. Provenance traces are now often captured using semantic data description models. The strength of these model is to both allow the capture of detailed provenance information and link the technical execution traces with the scientific domain concepts, thus easing the connection between (low-level) infrastructure traces and (higher-level) scientific computational traces.

5.2 Workflow Descriptions

All **workflow languages** detail orchestration rules for given types of **activities**, such as web services, legacy programs and grid jobs. Some are control-driven, i.e. they focus on how control is passed from an activity to the next. Some are data-driven, i.e. they focus on how data is transferred between activities and deduce which activities can run from data availability. Most hybrids are mixes of control and data-driven, but some systems adopt completely different paradigms, e.g. Kepler [3] with its explicit and flexible *Models of Computation*. Regardless of its approach to do it, a workflow language must somehow specify how activities will be orchestrated during enactment.

What the vast majority of workflow languages do not detail is why activities are orchestrated the way they are. Much like the nature of data is often left for humans to devise or documented in natural language, most scientific workflows carry little to no explicit information about what they do, only how they do it. One way to put it is that most workflow languages specify **methods** and not **goals**.

This poses some issues even outside the context of interoperability. Discovery, sharing, reuse and repurposing are all obviously hindered if it is impossible to tell what a workflow aims to achieve. It is also considerably harder to make sense of provenance traces if they remain at a purely technical level and feature no references to domain tasks whatsoever. If no automation is attempted for any of those tasks, then the issues can be solved with appropriate documentation. Automating them, however, will require **explicit semantics** to either be incorporated into the workflows themselves or somehow be associated with them.

As with the aforementioned data interoperability issues, scientific workflow interoperability further complicates those problems, because each framework deals with them in its own fashion.

The solution adopted by the SHIWA project, to design a low-level workflow language (i.e. IWIR) as a kind of assembly language for participating frameworks, is one way to overcome workflow description discrepancies preventing execution, but it does not explicit or leverage semantics in any way and thus does nothing for discovery, sharing, reuse and so on. It could also be argued that its extreme expressivity, which is a result of its design as a low-level pivot language, has somewhat reduced its accessibility.

Ideally, fine-grained interoperability should not only execute meta-workflows composed of non-native sub-workflows, but also represent them in a way that is both **accessible** (so users can understand what the workflow does and how it works) and **explicitly semantic** (so functions like discovery can be automated with minimal waste). One of the biggest hurdles before that vision is that there is no standard scientific workflow description.

5.3 Outline

In the next Section 6, we will give an overview of semantic data as it is leveraged by scientific communities who answered the survey organized by the ER-flow project as well as an overview of workflow descriptions in the field of scientific workflows at large. We will then give our recommendations for both semantic data and workflow descriptions, as they relate to scientific workflow interoperability, in Section 7. Section 8 concludes this study with future directions and all references are listed in Section 9.

6 State of the Art

6.1 Semantic Data

Semantic information is complementary to raw data. It can provide both technical information (*e.g.* data format, connection between different data items, data provenance...) and domain-specific information (*e.g.* production context, precise nature of data…). Semantic information may be implicit, known only from experiment designers and programmers who use this knowledge in the design of the experience support environment. However, the need to represent semantic data in computer-legible ways and to automatically process these annotations is increasingly recognized. In particular, explicit semantic data may be used for:

- Data encoding and data format: to transform data in a different format or to validate the compatibility of data with processing tools. In workflows, this can be used both at design-time (designer assistance) and at runtime (computation validity checking).
- Data precision and validity range: to check the validity or the coherence of data processing actions ordered.
- Nature and role of data: to validate the proper use of data in a computational process.
- Links between data items: to enrich data search. Data provenance is a specific kind of data link commonly captured at workflows execution time that helps scientists in understanding data products.

In the reminder of this section, we will particularly focus on the use of semantic data in the context of workflow interoperability. Section 6.1.1 focuses on domain-specific annotations used to enrich scientific data, especially in the application domains represented in ER-flow and Section 6.1.2 discusses the capture of provenance data in workflows.

Many kinds of meta-data (or annotation) mechanisms can be used to describe data semantics. The most advanced and the most widely adopted set of specifications to describe and manipulate semantics are undoubtedly the ones developed in the context of the *Semantic Web*[3] by the *World Wide Web Consortium*[4] (W3C). The Semantic Web makes a particular focus on the relations between different data items (a.k.a. *Linked Data*[5]). The basis for the Semantic Web is the *Resources Description Framework*[6] (RDF), which makes it possible to uniquely identify any data resource available over the Web and relate it to other data resources. RDF entities are typically composed by triples defining a source data entity, a relation, and a target data entity. The resources involved (source, relation and entity) are described by unique identifiers, which unambiguously refer to physical data artefacts or concepts described in a well-defined vocabulary (or ontology).

Several vocabulary definition languages, such as the *RDF vocabulary description language*[7] (RDFS) and the *Web Ontology Language*[8] (OWL) are specified to organize data. Vocabularies are both used to define terms within an area of concerns and classify them, characterizing possible relationships and defining possible constraints on using those terms. Vocabularies are therefore related to the type of reasoning that can be made upon entities

[3] Semantic Web, http://www.w3.org/standards/semanticweb/

[4] World Wide Web Consortium (W3C), http://www.w3.org

[5] Linked Data, http://en.wikipedia.org/wiki/Linked_data

[6] Resource Description Framework (RDF), http://www.w3.org/TR/rdf-mt/

[7] RDF Vocabulary Description Language (RDFS), http://www.w3.org/TR/rdf-schema/

[8] RDF Vocabulary Description Language (RDFS), http://www.w3.org/TR/rdf-schema/

described through these vocabularies. Different vocabulary description languages imply different reasoning abilities and different reasoning complexity.

RDF data sets can represent large databases of annotations. A set of RDF annotations can be seen as a graph composed of one or more connected components; nodes are the RDF triple sources and targets, and edges are the RDF triple relations. To retrieve relevant information in RDF graphs, the W3C defined the RDF-specific *SPARQL Protocol and RDF Query Language*[9] (SPARQL). SPARQL can be used to search for specific sub-graphs that match some search criterion (graph search pattern) in an RDF data set. Beyond its advanced pattern selection capabilities, SPARQL can also modify existing RDF triples or insert new data in RDF graphs through specific clauses. SPARQL is therefore a powerful and multipurpose query language.

Semantic technologies developed in the context of the extension of the Web of data towards richer and better-documented information are based on a rich set of widely adopted standards published by the W3C that are general enough to serve many purposes. In addition to this specification work, semantic Web technologies also benefit from a large tooling set implementation that reflects the level of adoption of these standards.

6.1.1 Use of Semantic Metadata in ER-flow Application Domains

A survey on scientific communities requirements and practices concerning semantic technologies was reported in ER-flow deliverable D5.3 [29]. This survey shows that the classical use of semantic technologies anticipated (content description, capturing information on the acquisition context, data provenance and data traceability for data-related needs) are very well covered by a majority of ER-flow communities.

Semantic data description, and in some cases semantic-aware search of data, have been widely adopted in many scientific areas. Some communities such as Astronomy & Astrophysics or Life Sciences are heavily relying on semantically enriched data. The globalization of scientific data and the trend towards on-line publication of open source data strongly pushes international-scale consortia to agree on standards and data models to archive and search data sets. The primary concerns raised are the sharing of data across sub-communities (understanding data content and converting data formats), the indexing of multiple and heterogeneous data sources, and advanced data search capabilities. Others, secondary use of semantic information, e.g., for data quality checking or long-term preservation, are sometimes mentioned but they are not considered as priorities yet.

Some of the data models developed are inspired by, or based on, semantic Web standards and technologies. The complete set of W3C standards, including the SPARQL semantic query and inference language, is rarely used though. Data search capabilities are therefore often ad-hoc and database-specific.

There is little use of machine-readable semantic annotations through semantic-aware processing tools yet, except for data format conversion. Semantic data models are often designed for human operators to non-ambiguously annotate data and reinterpret data produced by others.

Detailed report on each community use of semantic data can be found in [29].

6.1.2 Provenance Metadata

Data provenance plays a major role in addressing the emerging challenges in today's and future scientific environments, where proper methodologies adopted by the scientists need to guarantee that all the steps are correctly recorded and that they can be traced back to

[9] SPARQL Protocol and RDF Query Language (SPARQL), http://www.w3.org/standards/techs/sparql

facilitate reproducibility of scientific results. Data provenance refers to the capability of determining the origin and history, or lineage, of a certain piece of data [18]. Therefore, its importance is rapidly increasing in a connected digital world where open sources of data are becoming available for everyone [15].

In the domain of e-science, the scientific workflow management systems developers' community was among the first interested in using and deploying provenance toolkits and frameworks. This is due to the step-wise design approach used for composing and executing workflows, which suits the main ideas behind the data provenance approach (being able to capture provenance data automatically and at fine granularity [35, 36]). Examples of workflow management systems with provenance capabilities include Pegasus [17], Kepler [3], Taverna [20] and MOTEUR [16]. Typically, each of the systems used its custom terminology for defining and capturing data provenance.

Since the emergence of provenance as a standard (OPM [31] in 2007 followed by PROV [32] in 2013), scientists and researchers have increased their efforts in exploiting data provenance facilities. Such interest is motivated by the ability of the provenance standard to document the data generation process and to provide useful means for the scientists to better understand the way they perform their experiments and to trace, reproduce and explain the data analysis process.

Our motivation for considering the data provenance is two-fold: Firstly, it provides a set of documents (called PROV family of documents) that specifies the mechanisms for provenance data exchange and interoperability between heterogeneous systems; Secondly, it provides a flexible data model (PROV-DM) to describe the data flow and the processing steps with additional means to describe the processes and part of their semantics in a controlled manner.

Figure 3 illustrates the organization of PROV components and the dependency between them. PROV-DM is the core conceptual Data Model that defines a common vocabulary and concepts used to describe provenance, to which a set of constraints apply as defined by PROV-CONSTRAINTS [32]. Documents in the PROV family include:

- The *PROV OWL2* ontology defines the mapping of the PROV data model to RDF (*PROV-O*);
- *PROV-XML* is an XML schema for the PROV data model;
- *PROV-DC* is a mapping between Dublin Core and PROV-O;
- *PROV-SEM* is a declarative specification in terms of first-order logic of the PROV data model;
- *PROV-AQ* describes how to use Web-based mechanisms to locate and retrieve provenance information;
- *PROV-DICTIONARY* is a set of constructs for expressing the provenance of dictionary style data structures;
- *PROV-LINKS* is an extension to PROV to enable linking provenance information across bundles of provenance descriptions; and
- *PROV-N* is a human-readable notation for the provenance model.

The major improvements introduced in PROV, particularly the PROV family of documents, have advanced the provenance standard to a level that attracted a large scientific community and increased the number of efforts in adapting to, and implementing PROV.

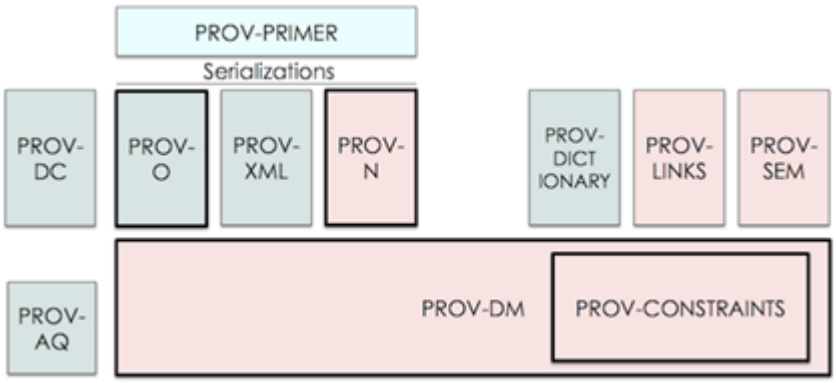

Figure 3. ***Organization of PROV according to [32] showing the core conceptual data model (PROV-DM), the family of documents it provides, and their dependencies. Bold bordered boxes denote W3C Recommendations, and regular bordered boxes denote Working Group Notes. The colors classify the audience for each document, namely: Users, Developers, and Advanced.***

Data provenance is described in PROV by the use and production of *Entities* by *Activities*, which may be influenced in various ways by *Agents*. *PROV-DM* is the core conceptual data model that defines a common vocabulary and concepts used to describe provenance. In brief, PROV-DM consists of:

- Core data types (Entity, Activity, and Agent);
- A set of Relations between the core data types as defined in PROV (16 in total);
- A set of Attributes that can be defined for each of the core data types and Relations; describing their properties as key-value pairs; and
- A Document grouping all the above.

Figure 4 illustrates a subset of the entity-relationship (ER) diagram of the PROV-DM core data types and their Relations. Note that the complete ER diagram would be too complex to display because it would include all optional Attributes that can be defined for the core data types and Relations.

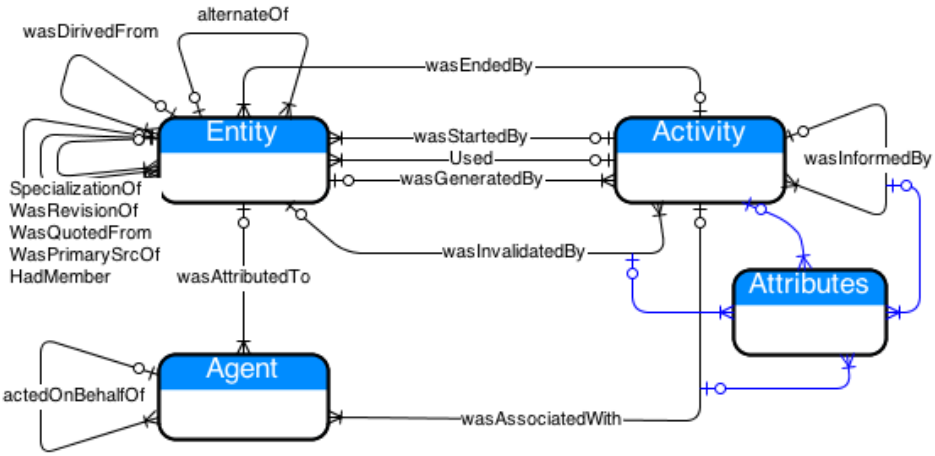

Figure 4. PROV-DM core data types with their prominent relationships. For readability reasons, only a subset of the relationships to the Attributes (highlighted in blue) is presented.

Relations in PROV-DM are always defined between the three core data types (*Entity*, *Activity*, and *Agent*). Their richness provides a strong mechanism to describe and express semantics of data. In addition, *Attributes* allow for further description of the core data types and their relationships.

6.2 Workflow Descriptions

While there is a *de facto* **standard language** for business workflows, called the Business Process Execution Language (BPEL), and while some works suggest using it or adapting it for scientific experiments [4]–[8], there is no such standard in the field of scientific workflows, as of yet. It is not easy to compare existing scientific workflow languages without a common basis, but though there are no widely accepted standards, there are **clear trends** in the field.

In analysing those trends, we focused on the notion of **abstraction level**, which is of the utmost importance for **accessibility**: the closer a workflow model is to the user domain, the easier it will be for that user to read, design and reuse workflows; conversely, the closer a workflow model is to the underlying infrastructure, the higher the entry barrier will be for scientists trying to use it for their scientific experiments.

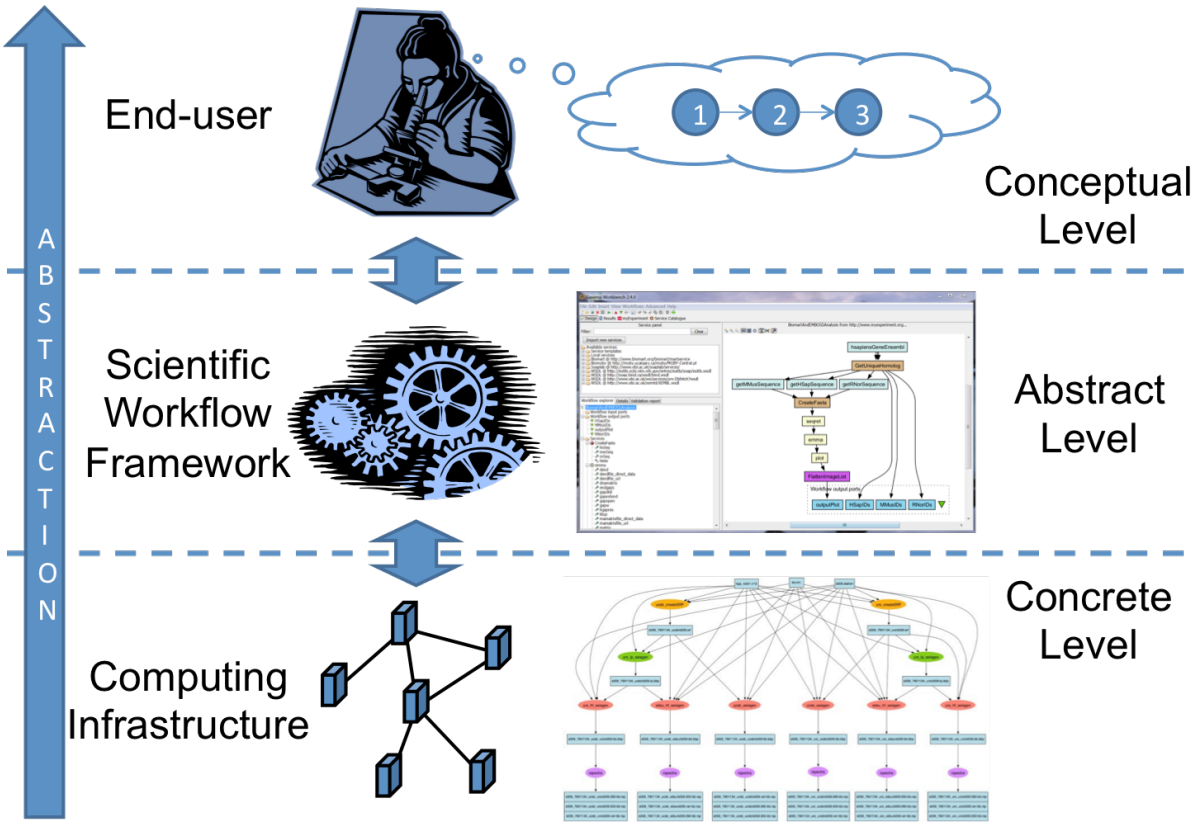

Figure 5 - Scientific Workflow Abstraction Levels

The distinction between the **Concrete Level** of the enactment and the **Abstract Level** where most scientific workflow models lie is commonly made in the field [9], but there is no consensus for the name of the highest level of abstraction (i.e. the user domain level): we chose to call it **Conceptual Level** to contrast it with the level below it and to emphasize its semantic nature, but it can even be called *Abstract* in some works, e.g. [10], and hence might confuse the unsuspecting reader.

While most scientific workflow models lie at the Abstract Level, their levels of abstraction do vary quite sensibly. Plenty of factors may contribute to elevate the abstraction level of scientific workflow model. We identified and chose to focus on the following four such factors (the convention is that the level of abstraction is higher if answers are positive):

- **Annotations**: Can the scientific workflows and/or their components be annotated with Semantic Annotations? With curated keywords? With informal tags?
- **Composition**: Does the system automatically compose scientific workflows? Does it provide the user with suggestions of edges or nodes? Does it check existing edges for potential mismatches?
- **Flexibility**: Is there any structural flexibility in the scientific workflow model? Can the same scientific workflow instance represent or lead to (via generation/transformation) structurally different processes (e.g. a sequence of 3 tasks vs. 4 parallel tasks)? Is there flexibility in the data representation (e.g. multiple files can be represented by a single input parameter)?
- **Indirection**: Is there indirection between the specification of a task and the technical execution thereof? Can a given activity represent multiple web services? Multiple programs? Multiple processing units?

We analysed 15 of the most well-known and widely used scientific workflow frameworks. TABLE shows how well each fulfils our four criteria:

- **Supported** means the feature is natively supported by the framework.
- **Marginally present** means that some effort was made towards fulfilling the criterion, but more is still required.
- **Third-party project** means that we found some published works detailing how to extend the framework with the feature, but it seems somewhat unlikely that the effort will be integrated into the main project.
- **Essentially absent** means that we have found no trace of the feature, though effort might be on-going towards implementing it or published works might have escaped our notice.

Framework	Annotations	Composition	Flexibility	Indirection
ASKALON [11]	Marginally present	Third-party project	Essentially absent	Essentially absent
Galaxy [12]	Supported	Essentially absent	Essentially absent	Marginally present
GWES [13]	Supported	Essentially absent	Essentially absent	Supported
Java CoG Kit [14]	Essentially absent	Third-party project	Essentially absent	Third-party project
Kepler [3]	Supported	Marginally present	Marginally present	Essentially absent
KNIME [15]	Marginally present	Essentially absent	Essentially absent	Essentially absent
MOTEUR [16]	Marginally present	Marginally present	Essentially absent	Marginally present
Pegasus [17]	Essentially absent	Essentially absent	Essentially absent	Marginally present
SHIWA [18]	Essentially absent	Essentially absent	Essentially absent	Essentially absent
Swift [19]	Essentially absent	Essentially absent	Marginally present	Essentially absent
Taverna [20]	Marginally present	Marginally present	Essentially absent	Marginally present
Triana [21]	Essentially absent	Essentially absent	Essentially absent	Essentially absent
VisTrails [22]	Supported	Third-party project	Essentially absent	Essentially absent
WINGS [23]	Supported	Supported	Essentially absent	Supported
WS-PGRADE [1]	Essentially absent	Essentially absent	Essentially absent	Supported

Table 4. Scientific Workflow Frameworks and Abstraction Level Features

Multiple things can be derived from our analysis of the field. It seems that **Annotations** and **Indirection** are slowly becoming staples in the field, though not all systems opt for explicitly semantic annotations, and that automated **Composition** is a hot topic. Clearly the new frontier is now structural **Flexibility**: even WINGS [23], whose level of abstraction is clearly the highest in the field as of this writing, presents no such flexibility whatsoever.

That lack may be a legacy of business workflows. Indeed, while business processes evolve with time, they are nowhere near as variable as simulations, given the exploratory nature of science. It thus may seem reasonable to think of two structurally different business workflows as different and independent workflows, unlike scientific workflows where a given scientific protocol could and often is implemented in ways that significantly differ in terms of structure.

The need for structural flexibility is made acute by interoperability: for a workflow language to represent *meta-workflows* in an accessible way, it must not only accommodate the various types of underlying models (such as control-driven and data-driven), but also seamlessly handle multiple levels of abstraction.

7 Recommendations

7.1 Semantic Data

7.1.1 Domain Data Semantics

As reported in D5.3 [29], all ER-flow user communities expressed a clear interest in the use of semantic technologies to address their data management needs yet different communities have very different expertise and experience with semantic data manipulation tools.

Many communities make use of rich data formats where raw data can be annotated through a format-specific mechanism. In particular:

- The Astronomy community makes use of different data formats, notably the standard Flexible Image Transport System (FITS) format. FITS is too open to constitute a data model in itself (all metadata is optional), but it provides a strong basis to define well-accepted data models.
- The Computational Chemistry community uniformly uses the Molecular Simulation Markup Language (MSML) formal language to describe both data and computational processes.
- Medical images are often stored in Digital Image and Communication in Medicine[10] (DICOM) format in a clinical context. In the image analysis context, several other formats more directly addressing the image transformation needs are common (e.g. NifTi[11] or Analyze[12] in the neuroimaging domain) although no unique standard emerged.

Astronomy is probably the domain with the highest level of expertise and the community is already making use of semantic data indexing techniques to achieve long-term cataloguing of astronomical archives. Indeed, astronomy is an observational science and observed events can often not be replicated. Observational data has to be preserved with great care. The Astronomy and Astrophysics community has thus set up an international-scale meta-repository to access distributed, cross-institution and cross-instruments data repositories in the context of the International Virtual Observatory Alliance[13]. Semantic technologies are used to address the heterogeneity of indexed repositories. A community-wide Digital Object Identifier scheme ensures non-ambiguous designation of astronomical objects. Most common astronomical quantities are defined in the IVOA Unified Content Descriptor models (UCDs). Other kinds of data are described through narrower use vocabularies or even much more informal “folksonomies”. The IVOA progresses towards an ontological definition of astronomical objects. New vocabularies are being developed for different sub-domains such as High Energy Astrophysics, Radio-Astronomy and Planetology. Heliophysics in particular inherits from this investment on semantic technologies. Many vocabularies (IVOA UCD1+, IVOA Thesaurus, VO-Theory), ontologies (HELIO ontology[14], Space Physics Archive Search

[10] DICOM: http://en.wikipedia.org/wiki/DICOM

[11] NifTi: http://nifti.nimh.nih.gov

[12] Analyze file format: http://web.archive.org/web/20070927191351/http://www.mayo.edu/bir/PDF/ANALYZE75.pdf

[13] IVOA: http://www.ivoa.net

[14] HELIO project: http://www.helio-vo.eu/

and Extract[15], HEK) and other data models (IVOA Characterization and Observation Data Models, ESA-FOREST data model for heliophysics, etc) have been developed. There is also a clear push towards open data publication[16]. There are many structured data repositories open to the community, among which NASA CDAS, HELIO/DPAS, the VSO and the JSOC have been cited. There are tools to link scientific publications with data (ADS). Further semantic resources are being developed, especially in the context of the FOREST and the SOLARIS projects.

In Life Sciences, ontological resources are numerous to the point that it may be difficult to identify most appropriate data models for a specific purpose. The lack of widely accepted standards reflects the different possible uses for biomedical data (medical, physio-pathological, radiological, biological, experiment setup data, etc) and the scattering of the community. There are several international-scale organizations that maintain websites and Web services for finding data, methods and vocabularies. Some of the most commonly used taxonomies and ontologies are ConceptWiki, SNOMED Clinical Terms (SNOMED-CT), Convergent Medical Terminology (CMT), NCBI taxonomy, RADLex ontology of radiology terms, Foundational Model of Anatomy (FMA), NeuroLex (formerly BIRNLex), Logical Observations Identifiers Names and Codes (LOINC), Neuroscience Information Framework Standard Ontologies (NIFSTD), OntoNeuroLOG ontology, etc. Structured and documented data-repositories are common in all sub-domains (Bioinformatics: GenBank, KEGG, Pathway databases, UniProt, etc; Neurosciences: MRI Atlases, PhysioNet, ADNI, OASIS, etc; Structural Biology: wwPDB), although they usually provide semantic annotations that are not necessarily in a machine-readable format. To face the scattering of resources, creating links between entities stored in different databases as well as links between data items and scientific publication is highly relevant.

In computational chemistry, the MoSGrid repository was developed to enrich data files (stored in XtreemFS) with additional metadata. It facilitates advanced data search through metadata analysis. Similarly, the HydroMeteorology community uses numerous data models and repositories at a local scale (e.g. Climate and Forecast standard names vocabulary), in absence of a clearly accepted standard.

7.1.2 Lack of Generic Semantic Information Framework

It can be seen from the variety of examples mentioned above that the systematic use of semantic annotations encounters two strong limitations:

- Most communities show an early adoption stage. Few standards have emerged among the plethora of early experiments and proposals. The field is not mature enough to rely on a few widely accepted formats except in very specific subdomains.
- Data models are inherently complex and different data usage scenarios usually lead to different data modelling results. Consequently, it may become difficult to identify and reuse proper data models even inside a given community. The modelling effort to produce “universal” data models should not be underestimated.

Data integration and data interoperability is already a challenge within each community participating to ER-flow. Implementing data interoperability even at the most basic level (data format considerations) seems rather intractable in the context of a generic and domain-agnostic platform such as the SSP. Shared data models and annotation models are very unlikely to be adopted if they have not been co-opted within a community and there is little chance that manageable data models can be designed in a broad context. There are two

[15] SPACE: http://www.spase-group.org

[16] Helophysics Data Environment: http://hpde.gsfc.nasa.gov

aspects of workflow-related semantic data interoperability that should not be neglected though: (i) the universality of semantic data manipulation standards established by the W3C and (ii) the link between data and data manipulation tools.

The strengths of Semantic Web standards proposed by the W3C are to be completely domain-agnostic, open, and already well accepted in a large Internet community. They are versatile languages that can apply to most data modelling tasks. Tools already exist to store, search, align, and reason upon data graphs. The adoption of these standards is vital to ensure interoperability of emerging semantic data management initiatives in all domains with future models. Furthermore, interoperability between different models is strongly enforced by the adoption of a common semantic data description framework. There are known limitation, in particular in terms of performance, but the state-of-the-art is quickly evolving and production quality software complying with these standards is emerging. This is thus a strong recommendation for all communities to investigate these standards and to workflow management environment to consider adopting these for their data processing tasks.

Representing the link between data manipulated by workflows (workflow activity inputs and outputs) and data transformation tools (workflow activities) is also highly relevant in scientific disciplines as discussed in Section 6.1. This link can be captured at a technical level (fine-grained traces of workflow activities execution) or at a domain level (usually coarser-grained traces linking input/output data with the data transformation function of workflow activities). The technical level is usually completely domain-independent as the purpose is to link domain-agnostic activities with data pieces. It is further discussed in Section 7.1.3. The domain level requires binding the (domain-specific) activities functions with the transformed data, and therefore requires adapted ontological resources and expertise to bind the technical data processing artefacts with the ontology-defined domain concepts. It is not necessarily implementing a one-to-one relation between activities and concepts since a specific data transformation process may require executing several activities (sub-workflow) or a single activity might implement several data transformation functions. This aspect is further detailed in Section 7.2.

7.1.3 Provenance Metadata

The proper strategy for provenance data collection would be better achieved at the workflow execution level. Automating provenance traces capture in workflow engines makes provenance data collection systematic, reliable and cost-effective. In larger workflows, generated traces can represents millions of annotations. Additionally, provenance data has to be presented according to a standard format, to better facilitate the data exchange and interoperability between heterogeneous systems. Thus, achieving a common understanding of the data format and its semantics. PROV has become the *de facto* standard and is widely adopted these last years.

However, all workflow management systems are not instrumented to capture provenance information. In that case, the workflow management system log files and the Distributed Computing Infrastructure job execution logs provide useful information to reconstruct provenance. Figure 6 describes the architecture of a provenance data collector for non-instrumented workflow management systems. For each workflow execution, the collector captures data related to the workflow jobs, their inputs and output results, users in charge of the experiments, and dependency relationships among these data. Additionally, the collector organizes the provenance information according to their execution context. The collector thus analyzes both the workflow management system database/logs and the log files generated by the jobs on the Distributed Computing Infrastructures to reconstruct provenance traces complying to the PROV data model.

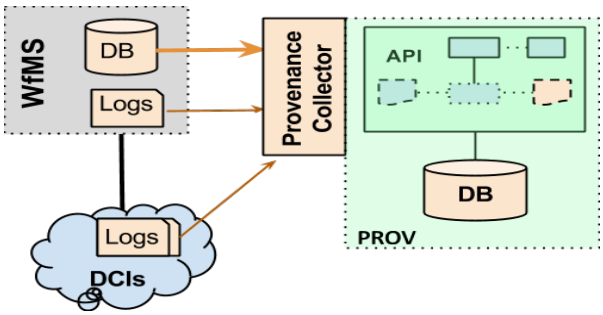

Figure 6 – Architecture of the provenance data collector.

Table 5 illustrates the mapping of gUSE [1] executed workflow data to PROV concepts. The mapping is straightforward: each workflow maps to a *Document*, jobs are mapped to *Activities*, input/output data to *Entities* and users are mapped to *Agents*. The most important *Relations* linking the input data to the output results in each experiment are *used* and *wasGeneratedBy*.

gUSE Concept	PROV Counterpart	Description
executed workflow	Document	executed workflow
Job	Activity	executable code
Input Port	Entity	input data of jobs
Output Port	Entity	output results of jobs
User	Agent	workflow user
Job -> Input Port	Relations: Used	Job's input data
Output Port -> Job	Relations: wasGeneratedBy	Job's output data
User -> Job	Relations: wasAssociatedWith	User executing the job

Table 5. gUSE-PROV concept mapping: mandatory data

Additionally, descriptive details documenting the properties of the core data types and relationships are mapped into PROV as *Attributes,* such as format, location, and size of input/output data; hostname of computing nodes where the jobs are executed; operating system on the computing nodes; the version of the software tools; etc.

Two main challenges could be faced during the data collection and organization according to PROV. The first relates to accessing the log files on the DCIs, where the logs are only kept for a short period of time after the job execution. We therefore must configure the provenance collector to be triggered as soon a workflow terminates execution. For this reason, for most workflows executed in the past it will not be possible to collect details such as start and end time of jobs and computing nodes on which they run. Job start and end time are mapped as direct members of an *Activity* in PROV; however, the final status of a job had to be mapped as an *Attribute* of that *Activity*.

The second challenge relates to reconstructing the full dependencies between data and jobs in a workflow from the various scattered information sources of the workflow management system and DCIs job logs. In particular, various operations are needed to correctly link all

jobs to their proper input and output data in the context of the workflow. The full dependencies could be made possible by identifying the jobs that consume the output generated by other jobs.

Recent developments, following the above guidelines, addressed the design and implementation of the provenance framework using an optimal database schema to store provenance for scientific experiments. These developments aims at enhancing scientific environments and platforms with provenance capabilities and include:

(1) A core component provenance data collector for workflows defined and executed under the WS-PGRADE/gUSE framework [37].

(2) A core component to automatically gather provenance data from existing grid workflow enactments services, based on MOTEUR [40, 41].

(3) An integrated provenance data collector within the WS-VLAM workflow system [38, 39].

The deployment of the provenance framework within the workflow management systems will enable the automatic collection of provenance information in interoperable format, whenever scientists use the platform to analyze and process their data.

7.2 Workflow Descriptions

Among works that best illustrate the potential of combining semantic data with scientific workflows are those surrounding the WINGS [23] framework, which was built from the start as a semantic framework meant to focus on user domains. Indeed, most of the works around WINGS deal with leveraging semantic data ontologies to ease and assist the design and sharing of scientific workflows. Notably:

- [24] describes a framework built on top of WINGS to automatically transform user queries into scientific workflows;
- [10] describes an approach to publish “abstract workflows” (i.e. workflow templates with undetermined Activities) and “executable workflows” (i.e. abstract workflows as defined in Section 6.2) as Open Linked Data through an extension of the Open Provenance Model [31];
- [25] focuses the framework on the state of the art in data mining pipelines and obtains great results on automated composition and increased accessibility; And
- [26] mines provenance data to detect “abstract templates” and thus elevate the abstraction level of WINGS workflows automatically.

There is very little doubt that WINGS is the most well-known and furthest developed Conceptual Level scientific workflow framework to date. It would nonetheless be a poor fit as a meta-workflow model, for interoperability purposes, for two main reasons:

- The WINGS approach is to close the world of possibilities by modelling every available tasks and matching abstraction levels so that every element from the Conceptual Level corresponds to one or more elements from the Abstract Level. That closed-world approach has allowed the team behind WINGS to achieve great results for the automation of the scientific workflow design process, but it does considerably raise the barrier for community growth.
- Like the overwhelming majority of scientific workflow models, WINGS features absolutely no structural flexibility. That is problematic in at least two ways: on the one hand, it means that even the most technical steps in a workflow have to be modelled at every abstraction level, “polluting” the higher levels; on the other hand, it makes

the model unsuitable for fine-grained interoperability, since it cannot account for the variety of structural constraints of non-native workflow languages.

Ideally, a meta-workflow language, used to describe meta-workflows composed of sub-workflows pertaining to different scientific workflow models and DCIs, would fulfil all four criteria for high abstraction level we identified in 6.2:

- Explicitly semantic **annotations** would not only leverage semantic models, but allow for better handling of semantic data;
- Automated **composition** would make the modelling of meta-workflows less tedious and error-prone and thus lower the entry barrier to scientific workflow interoperability as a whole;
- **Indirection** would be vital to handle a great variety of essentially incompatible DCIs;
- And structural **flexibility** would let workflow designers model non-native sub-workflows in the meta-workflow language in a cohesive and transparent way.

While there are no perfect candidates for such a role, as of this writing, it is important to note that the WINGS project is evolving in this direction, as it is currently working towards adding more flexibility to the model and approach.

As further proof that the criteria described here are not too far-fetched to be practical, we will now describe Conceptual Workflows [27], [28], [30], a scientific workflow model which fulfils all four of the criteria highlighted here.

7.2.1 Conceptual Workflows

An overview of the Conceptual Workflows model is given below. This model aims at providing a representation of scientific workflows both at the abstract and the conceptual level (referring to the abstraction levels introduced in Section 6.2). Its UML representation is shown in Figure 7. It can be decomposed in three main parts:

- The Conceptual part describes the workflow elements at the conceptual level. It is at this level that domain-specific data transformation functions are defined in particular. This level is further described in this Section.
- The Abstract part describes the workflow activities. As discussed in Section 7.1.2, there is not necessarily a one-to-one relation between data transformation functions and workflow activities. This level is further described in Section 7.2.2.
- The Semantic part describes the binding between conceptual or abstract elements and domain-related semantic annotations. This level is further described in Section 7.2.3.

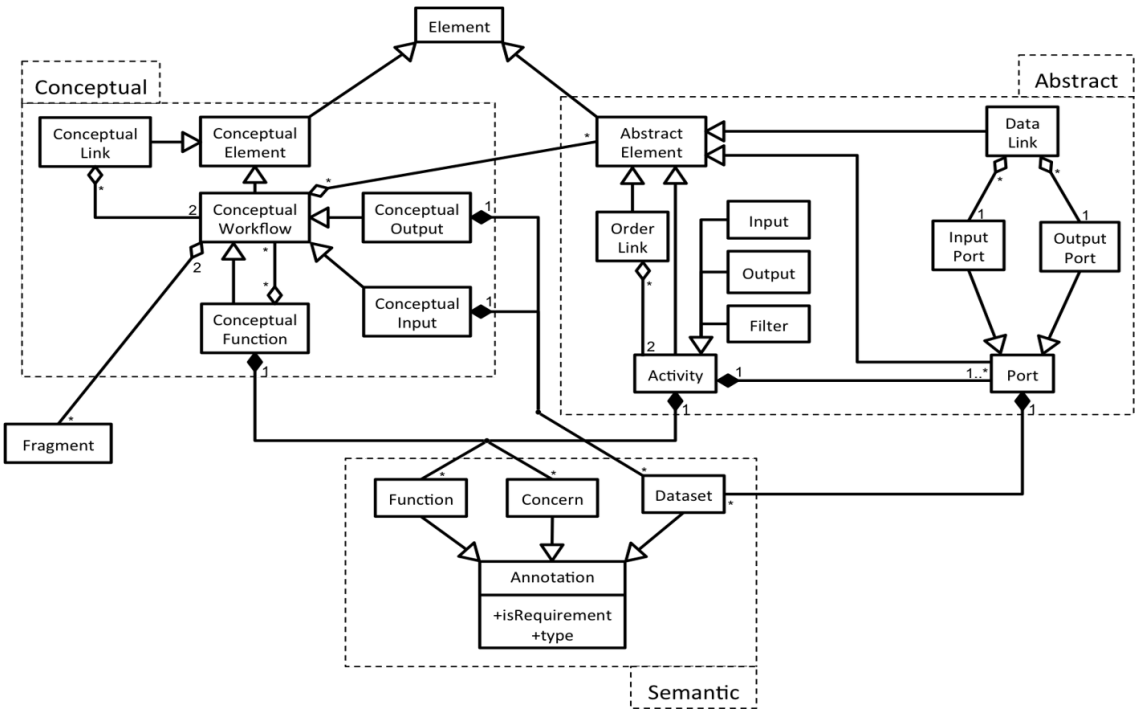

Figure 7 - Conceptual Workflows Meta-Model

Conceptual Workflows are modelled through nested directed cyclic graphs. Direct graphs have been adopted because they are at the base of the majority of scientific workflow frameworks. Their nesting allows modelling multiple levels of abstraction as well as encapsulation. Workflow input data is modelled by Conceptual Inputs, data analysis steps are modelled by Conceptual Functions, workflow output products are modelled by Conceptual Outputs, and dependencies between those elements are modelled by Conceptual Links. Conceptual inputs, outputs, functions and links are used to describe direct workflow graphs as illustrated in Figure 8 (left). The workflow in Figure 8 (right) illustrates the workflow nesting capability of this model: the image spatial alignment function implemented in this workflow can be decomposed into two steps: spatial transformation estimate (a process called “registration” in this community) and spatial transformation application (transformation process). Conceptual functions can contain both nested conceptual functions and abstract elements representing workflow activities. Similarly, the conceptual link may represent any type of dependencies between the workflow activities (*e.g.* data dependencies or control dependencies).

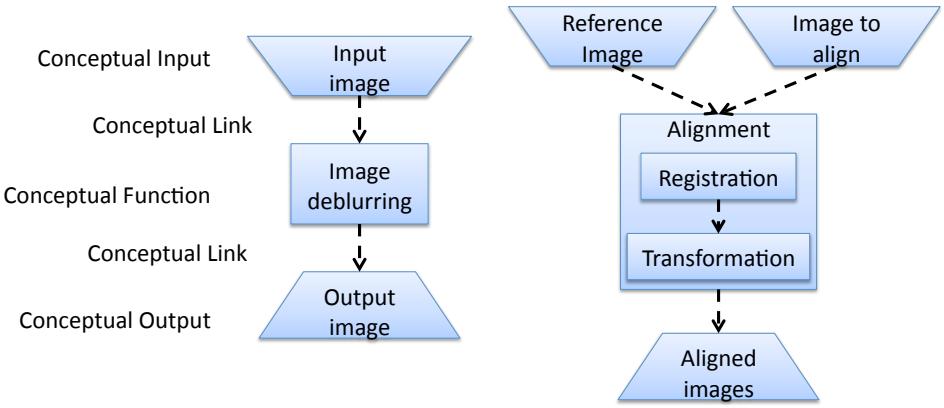

Figure 8 – Example conceptual workflows. Left: conceptual elements. Right: workflow nesting example.

7.2.2 Link with Workflow Activities

From the viewpoint of the Conceptual Workflow that embeds it, an Activity is a black box representing an executable artefact (*e.g.* a web service, a grid job or a legacy program). The arguments of the underlying artefact are modelled by Input Ports and its products by Output Ports associated with the Activity. Each Activity has at least one Port; otherwise it would be impossible to connect it to the rest of the workflow.

In addition to regular Activities, the Conceptual Workflow Model defines the following special ones:

- Inputs are Activities with at least one Output Port and no Input Port;
- Outputs are Activities with at least one Input Port and no Output Port; and
- Filters are special Activities implementing conditional constructs: they have one Input Port, two Output Ports: then and else and a logical condition called a Guard. Whenever a piece of data d is transferred to a Function, the associated Guard is evaluated: d is passed along the then branch if the Guard is True, along the else branch otherwise.

In practice, Inputs and Outputs are most often data constants or references to files, but they may also be executable artefacts (such as web services) that either only produce or only consume data.

Finally, there are two types of flow in workflows and, accordingly, there are two types of links in the Abstract part of the Conceptual Workflow Model:

- Data Links represent data flow, *i.e.* data transfers from a source to a target, with the target waiting for the data to execute; and
- Order Links represent control flow, *i.e.* control transfers - which can be seen as order constraints, hence the name - from a source to a target, with the target waiting until the source finishes to execute.

As they represent data transfers, Data Links connect Output Ports to Input Ports, whereas Order Links connect two Activities directly.

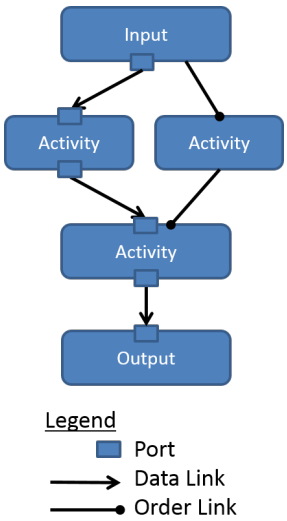

Figure 9 - Main Abstract Level Elements

7.2.3 Exploitation of Semantic Web Standards

In order to leverage Semantic Web technologies and to ensure maximum flexibility when it comes to Semantic Annotations, the Conceptual Workflow Model itself is captured in an ontology called COnceptual WORKflow (COWORK) [30]. Conceptual Elements and Abstract Elements are bound with domain concepts and non-functional concerns they model, as defined in external ontologies. As a result, many Conceptual Elements and Abstract Elements can bear semantic Annotations. Three things in the Conceptual Workflow Model define an Annotation, as illustrated in Figure 10:

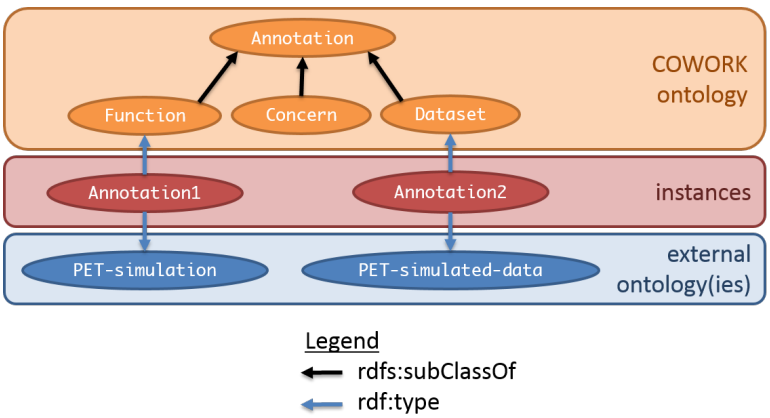

Figure 10 - Annotation System

- **Type**. Domain ontologies often contain extensive taxonomies of the concepts that the domain workflows handle. In order to exploit type inference, Annotations are simultaneously of the type cowork:Annotation and of a type defined in an external ontology.
- **Role**. At a computation-independent level of abstraction, Conceptual Elements do not yet achieve any goals or fulfil any criteria. Therefore, at that level, Annotations associating domain concepts and non-functional concerns with Conceptual Elements are Requirements: they represent the objectives of the Conceptual Elements they annotate, rather than what the Conceptual Elements do. Mapped Conceptual Elements, which embed sub-workflows and/or abstract workflows to fulfil their

Requirements, no longer need them. They are annotated instead with Specifications that describe the goals achieved and the criteria satisfied by the Conceptual Elements they annotate. Abstract Elements also bear Specifications describing the goals they achieve and the criteria they satisfy, so that they can be suggested as suitable candidates to embed in high-level Conceptual Workflows, to fulfil their Requirements. The graphical representation for distinguishing Requirements from Specifications in Conceptual Workflows is shown in Figure 11.

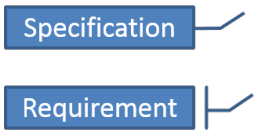

Figure 11 - Semantic Annotation Roles

- **Meaning**. Annotations are linked to external semantic concepts through their Type. For the purpose of modelling workflows, there are three categories of relevant concepts and we distinguish three different Meanings accordingly: Functions describe scientific process steps, Concerns describe non-functional criteria; and Datasets describe data content and/or format.

7.2.4 Application to Workflow Assistance Design

Conceptual workflows contain information on the workflow structure as well as semantic annotations describing the different workflow parts, both at the technical and the scientific domain level. Furthermore, they provide a dual view over workflows, both in terms of Conceptual (user domain) and Abstract (technical) levels. An annotated conceptual workflow can have several applications such as coherence checking, compatibility of input data checking, and provenance traces generation at the domain level. When stored in a knowledge base, conceptual workflows can also be used for workflow fragments search based on domain-defined workflow characteristics. Finally, conceptual workflows can be transformed using the Semantic Web SPARQL graph manipulation language, thus providing assistance for the workflow composition issue.

Conceptual workflows transformation is based on the use of annotated workflow fragments stored in a domain-specific knowledge base. Fragments are composed of two distinct Conceptual Workflows as illustrated in Figure 12: the Blueprint represents the content of the Fragment and the Pattern represents the context in which the Fragment is relevant. The Blueprint is in every way a regular Conceptual Workflow, but the Pattern is slightly different: its elements are interpreted as variables. For instance, a Conceptual Function *CF* annotated with a Function *F* in a stand-alone Conceptual Workflow or in a Blueprint will represent a specific instance, but the same pair in a Pattern will be interpreted as “any Conceptual Function annotated with *F*”. The names of elements in Patterns are thus disregarded when they are matched against a base workflow.

The example in Figure 12, is taken from the nuclear medical image simulation domain (PET imaging). The pattern represents a Conceptual Workflow that provides the domain-specific PET-simulation function and the non-functional SplitAndMerge optimization (parallel processing). The parallel PET-simulation functionality may not be implemented in any single workflow activity, but the Blueprint workflow fragment shows how a combination of a “sorteo_single” activity (which generates single beam rays) and the “sorteo_emission” activity (which simulates beam emission and interaction with the image body) can be composed. The Blueprint can be substituted to the Pattern without change in the workflow semantic after taking care of replacing, deleting and creating the workflow elements as indicated.

Pattern	Blueprint	Resulting Graph
✓	✓	Preserved ≈
✗	✓	Generated ✚
✓	✗	Deleted ⏏
✗	✗	Untouched

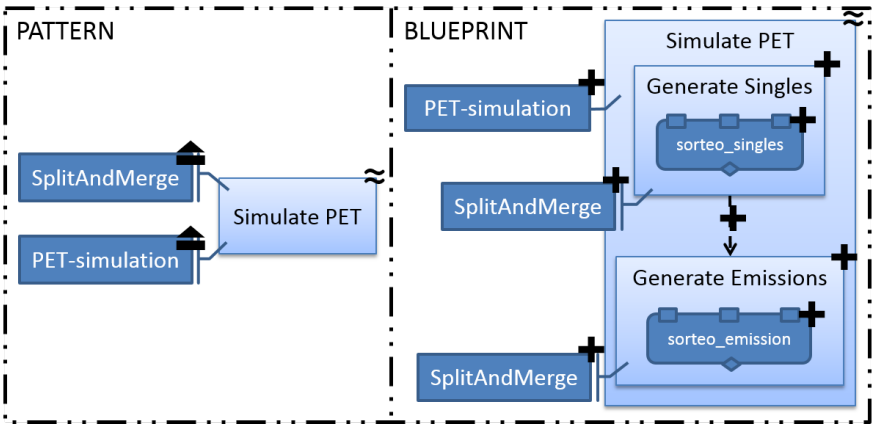

Figure 12 - Fragment Weaving

It is shown in [30] how a Conceptual Workflow can be transformed into an executable Abstract Workflow once all its Conceptual Elements have been resolved and substituted by Abstract Elements. The model thus represents a high-level abstract description of workflows that could be instantiated in different languages.

What the Conceptual Workflow model and associated framework lack most to be better matched for the needs of scientific workflow interoperability is an interoperability-ready Transformation Process (from the Conceptual Level to the Abstract Level) that would not only be able to convert into multiple target languages simultaneously, but also somehow guide the user in the choice of which system should handle which part of the overall workflow. For instance, the system could detect incompatibilities between target languages and the activities; target DCIs or language features the workflow designer wants to use.

8 Conclusions

The semantics of data and data transformation processes plays a critical role in scientific experiments. Initially completely in the hand of workflow designers, standard semantic description formats and semantic data manipulation tools now make it possible for workflow management environments to manipulate and take into account this information for assisting workflow designers and workflow system users. Semantic data can be used for many different purposes including scientific data transformation process documentation, produced data analysis, reproducibility of results, reuse of workflows, assistance to workflow design, and workflow interoperability.

This document describes the state of the art of semantic technology usage within ER-flow communities and by scientific workflow environments. Although the current usage remains rather low, all communities clearly have needs for more semantic-aware scientific workflow environments. An area in which semantic data representation is commonly used is the one of provenance traces description. However, most existing environments are considering low-level technical execution traces, which are mostly of interest for workflow designers, and not the higher-level domain-specific scientific data transformation traces that help scientists linking the data manipulated with the scientific objectives of the data transformation processes implemented as workflows.

Although there is no generic semantic data manipulation framework that is well established enough to be accepted by all ER-flow communities, this document provides recommendations and best practices for semantic data representation and usage in scientific workflow environments. The relevance of the Semantic Web standards in the context of interoperability of different systems and the pivotal role of semantic technologies in workflow provenance traces capture are outlined. A domain-agnostic model is proposed to link any scientific workflow transformation process to manipulated data sets.

9 References

[1] P. Kacsuk, Z. Farkas, M. Kozlovszky, G. Hermann, A. Balasko, K. Karoczkai, and I. Marton, "WS-PGRADE/gUSE Generic DCI Gateway Framework for a Large Variety of User Communities," *J. Grid Comput.*, vol. 10, no. 4, pp. 601–630, 2012.

[2] K. Plankensteiner, J. Montagnat, and R. Prodan, "IWIR: A Language Enabling Portability Across Grid Workflow Systems," in *Workshop on Workflows in Support of Large-Scale Science(WORKS'11)*, 2011.

[3] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, "Scientific Workflow Management and the Kepler System," *Concurr. Comput. Pract. Exp.*, vol. 18, no. 10, pp. 1039–1065, Aug. 2006.

[4] W. Emmerich, B. Butchart, L. Chen, B. Wassermann, and S. Price, "Grid Service Orchestration Using the Business Process Execution Language (BPEL)," *J. Grid Comput.*, vol. 3, no. 3–4, pp. 283–304, Sep. 2005.

[5] A. Akram, D. Meredith, and R. Allan, "Evaluation of BPEL to Scientific Workflows," in *CCGRID '06: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*, 2006, pp. 269–274.

[6] A. Slominski, "Adapting BPEL to Scientific Workflows," in *Workflows for e-Science*, Springer-Verlag, 2007, pp. 208–226.

[7] B. Wassermann, W. Emmerich, B. Butchart, N. Cameron, L. Chen, and J. Patel, "Sedna: A BPEL-Based Environment for Visual Scientific Workflow Modeling," in *Workflows for e-Science*, Springer-Verlag, 2007, pp. 428–449.

[8] M. Sonntag and D. Karastoyanova, "Model-as-you-go: An Approach for an Advanced Infrastructure for Scientific Workflows," *J. Grid Comput.*, pp. 1–31, 2013.

[9] J. Yu and R. Buyya, "A taxonomy of scientific workflow systems for grid computing," *ACM SIGMOD Rec.*, vol. 34, no. 3, pp. 44–49, Sep. 2005.

[10] D. Garijo and Y. Gil, "A new approach for publishing workflows: abstractions, standards, and linked data," in *Proceedings of the 6th workshop on Workflows in support of large-scale science(WORKS)*, 2011, pp. 47–56.

[11] T. Fahringer, R. Prodan, R. Duan, J. Hofer, F. Nadeem, F. Nerieri, S. Podlipnig, J. Qin, M. Siddiqui, H. Truong, A. Villazon, and M. Wieczorek, "ASKALON: a development and grid computing environment for scientific workflows," in *Workflows for e-Science*, Springer-Verlag, 2007, pp. 450–471.

[12] J. Goecks, A. Nekrutenko, and J. Taylor, "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences," *Genome Biol.*, vol. 11, no. 8, p. R86, 2010.

[13] F. Neubauer, A. Hoheisel, and J. Geiler, "Workflow-based Grid applications," *Futur. Gener. Comput. Syst.*, vol. 22, no. 1–6, pp. 6–15, Sep. 2005.

[14] G. von Laszewski, I. Foster, J. Gawor, and P. Lane, “A Java commodity grid kit,” *Concurr. Comput. Pract. Exp.*, vol. 13, no. 8–9, pp. 645–662, 2001.

[15] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel, “KNIME: The Konstanz Information Miner,” in *GfKl(GfKl)*, 2007, pp. 319–326.

[16] T. Glatard, J. Montagnat, D. Lingrand, and X. Pennec, “Flexible and efficient workflow deployement of data-intensive applications on grids with MOTEUR,” *Int. J. High Perform. Comput. Appl. Spec. issue Spec. Issue Work. Syst. Grid Environ.*, vol. 22, no. 3, pp. 347–360, Aug. 2008.

[17] E. Deelman, G. Singh, M. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz, “Pegasus: a Framework for Mapping Complex Scientific Workflows onto Distributed Systems,” *Sci. Program. J.*, vol. 13, no. 3, pp. 219–237, 2005.

[18] D. Krefting, T. Glatard, V. Korkhov, J. Montagnat, and S. Olabarriaga, “Enabling Grid Interoperability at Workflow Level,” in *Grid Workflow Workshop 2011(GWW’11)*, 2011.

[19] Y. Zhao, M. Hategan, B. Clifford, I. Foster, G. von Laszewski, I. Raicu, T. Stef-Praun, and M. Wilde, “Swift: Fast, Reliable, Loosely Coupled Parallel Computation,” in *IEEE International Workshop on Scientific Workflows*, 2007.

[20] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li, “Taverna: A tool for the composition and enactment of bioinformatics workflows,” *Bioinforma. J.*, vol. 17, no. 20, pp. 3045–3054, 2004.

[21] I. Taylor, M. Shields, I. Wang, and A. Harrison, “The Triana Workflow Environment: Architecture and Applications,” in *Workflows for e-Science*, Springer-Verlag, 2007, pp. 320–339.

[22] S. P. Callahan, P. J. Crossno, J. Freire, C. E. Scheidegger, C. T. Silva, and H. T. Vo, “VisTrails: enabling interactive multiple-view visualizations,” in *Visualization, 2005. VIS 05. IEEE*, 2005, pp. 135–142.

[23] Y. Gil, V. Ratnakar, K. Jihie, J. Moody, E. Deelman, P. A. Gonzales-Calero, and P. Groth, “Wings: Intelligent Workflow-Based Design of Computational Experiments,” *IEEE Intell. Syst.*, vol. 26, no. 1, pp. 62–72, Jan. 2011.

[24] Y. Gil, J. Kim, P. A. Gonzales-Calero, J. Moody, and V. Ratnakar, “A semantic framework for automatic generation of computational workflows using distributed data and component catalogues,” *J. Exp. Theor. Artif. Intell.*, vol. 23, no. 4, pp. 389–467, 2011.

[25] M. Hauder, Y. Gil, Y. Liu, R. Sethi, and H. Jo, “Making data analysis expertise broadly accessible through workflows,” in *Proceedings of the 6th workshop on Workflows in support of large-scale science(WORKS)*, 2011, pp. 77–86.

[26] D. Garijo, O. Corcho, and Y. Gil, “Detecting common scientific workflow fragments using templates and execution provenance,” in *Proceedings of the seventh international conference on Knowledge capture(K-CAP ’13)*, 2013, pp. 33–40.

[27] N. Cerezo, J. Montagnat, and M. Blay-Fornarino, “Computer-Assisted Scientific Workflow Design,” *J. Grid Comput.*, vol. 11, no. 3, pp. 585–610, Sep. 2013.

[28] N. Cerezo and J. Montagnat, “Scientific Workflow Reuse through Conceptual Workflows,” in *Proceedings of the 6th Workshop on Workflows in Support of Large-Scale Science*, 2011, pp. 1–10.

[29] J. Montagnat, N. Cerezo, S. Olabarriaga, " Requirements for domain semantic data and workflow description", ER-flow deliverable D5.3, August 2013.

[30] N. Cerezo, "Conceptual Workflows", PhD Thesis, University of Nice Sophia Antipolis, December 2013.

[31] L. Moreau, et al. "The Open Provenance Model Core Specification (v1.1)". Future Generation Computer Systems, vol. 27(6) pp.743-756, June 2011.

[32] PROV-Overview: http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/

[33] J. Zhao, C. A. Goble, R. Stevens, and S. Bechhofer, "Semantically Linking and Browsing Provenance Logs for Escience," in ICSNW, 2004.

[34] J. Widom, "Trio: A System for Integrated Management of Data, Accuracy, and Lineage," in CIDR, 2005.

[35] Davidson, S.B., Freire, J.: Provenance and scientific workflows: challenges and opportunities. In: SIGMOD Conference, pp. 1345–1350 (2008).

[36] Gil, Y., Deelman, E., Ellisman, M., Fahringer, T., Fox, G., Gannon, D., Goble, C., Livny, M., Moreau, L., Myers, J.: "Examining the challenges of scientific workflows". IEEE Computer 40(12), 26–34 (2007)

[37] Provenance for gUSE: http://www.sci-bus.eu/wiki/-/wiki/Public/PROV4gUSE

[38] Michael Gerhards, Sascha Skorupa, Volker Sander, Adam Belloum, Dmitry Vasunin, A. Benabdelkader. HisT/PLIER: A two-fold Provenance Approach for Grid-enabled Scientific Workflows using WS-VLAM. In the 12th IEEE/ACM International Conference on Grid Computing, 22-23 September 2011, Lyon, France, 2011. ICGC 2011

[39] Michael Gerhards, Sascha Skorupa, Volker Sander, Adam Belloum, Dmitry Vasunin, A. Benabdelkader. Provenance Opportunities for WS-VLAM: An Exploration of an e-Science and an e-Business Approach. Submitted to the 6th Workshop on Workflows in Support of Large-Scale Science, November 12-18, 2011, Seattle, 2011. - WSLSS 2011

[40] A. Benabdelkader,M. Santcroos, S. Madougou, A. H. van Kampen, S. Olabarriaga. A Provenance approach to trace scientific experiments on a grid infrastructure. In the 7th IEEE International Conference on e-Science, 05-08 December 2011, Stockholm, Sweden, 2011: 134-141. - e-science 2011

[41] Souley Madougou, Shayan Shahand, Mark Santcroos, Barbera D. C. van Schaik, Ammar Benabdelkader, Antoine H. C. van Kampen, Sílvia Delgado Olabarriaga: Characterizing workflow-based activity on a production e-infrastructure using provenance data. Future Generation Comp. Syst. 29(8): 1931-1942 (2013) - FGCS 2013