Wf4Ever: Advanced Workflow Preservation Technologies for Enhanced Science

STREP FP7-ICT-2007-6 270192

Objective: ICT-2009.4.1b — "Advanced preservation scenarios"

D2.2v1 Design, implementation and deployment of workflow lifecycle management components - Phase I

Deliverable Co-ordinator: **Sean Bechhofer**

Deliverable Co-ordinating Institution: **University of Manchester**

Other Authors: **Khalid Belhajjame**

This deliverable describes the first phase of delivery of workflow lifecycle management components. It includes a description of the Research Object Model, which facilitates interoperation between components; an initial Research Object Storage and Retrieval Service; RO Manager command line tool; and a definition of a model for workflow abstraction.

Document Identifier:	Wf4Ever/2012/D2.2v1/0.6	Date due:	July 31, 2012
Class Deliverable:	Wf4Ever FP7-ICT-2007-6 270192	Submission date:	July 31, 2012
Project start date	December 1, 2010	Version:	0.6
Project duration:	3 years	State:	Final
		Distribution:	Public

Wf4Ever Consortium

This document is part of the Wf4Ever research project funded by the IST Programme of the Commission of the European Communities by the grant number FP7-ICT-2007-6 270192. The following partners are involved in the project:

Intelligent Software Components S.A. (ISOCO) – Coordinator Edificio Testa, Avda. del Partenón 16-18, 1^o, 7^a Campo de las Naciones, 28042 Madrid Spain Contact person: Jose Manuel Gómez Pérez E-mail address: jmgomez@isoco.com	**University of Manchester (UNIMAN)** School of Computer Science Oxford Road, Manchester M13 9PL United Kingdom Contact person: Carole Goble E-mail address: carole.goble@manchester.ac.uk
Universidad Politécnica de Madrid (UPM) Departamento de Inteligencia Artificial, Facultad de Informática. 28660 Boadilla del Monte. Madrid Spain Contact person: Oscar Corcho E-mail address: ocorcho@fi.upm.es	**Instytut Chemii Bioorganicznej PAN - Poznan Supercomputing and Netowrking Center (PSNC)** Network Services Department Ul Z. Noskowskiego 12-14 61704 Poznań Poland Contact person: Raul Palma E-mail address: rpalma@man.poznan.pl
University of Oxford (OXF) Department of Zoology South Parks Road, Oxford OX1 3PS United Kingdom Contact person: Jun Zhao, David De Roure E-mail address: jun.zhao@zoo.ox.ac.uk david.deroure@oerc.ox.ac.uk	**Instituto de Astrofísica de Andalucía (IAA)** Dpto. Astronomía Extragaláctica. Glorieta de la Astronomía s/n, 18008 Granada Spain Contact person: Lourdes Verdes-Montenegro E-mail address: lourdes@iaa.es
Leiden University Medical Centre (LUMC) Department of Human Genetics Albinusdreef 2, 2333 ZA Leiden The Netherlands Contact person: Marco Roos E-mail address: M.Roos1@uva.nl	

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to the writing of this document or its parts:

- iSOCO
- OXF
- PSNC
- UNIMAN
- UPM

Change Log

Version	Date	Amended by	Changes
0.1	02-07-2012	Sean Bechhofer	Initial Version
0.2	17-07-2012	Sean Bechhofer	Updates and Examples
0.3	18-07-2012	Sean Bechhofer	Adding PURLs
0.4	19-07-2012	Khalid Belhajjame	Adding decay section
0.5	19-07-2012	Esteban Garcia	Adding abstraction section
0.6	20-07-2012	Sean Bechhofer	Editing Executive summary. Addition of figure.

Executive Summary

This deliverable describes the first phase of delivery of workflow lifecycle management components. These components are focused around the Wf4Ever Research Object Model (RO model), which provides descriptions of workflow-centric ROs – aggregations of content. This model is used to structure and describe ROs which are then stored and manipulated by the components of the Wf4Ever Toolkit.

The RO Model provides a framework for describing aggregations of content along with annotations of the aggregated resources, a vocabulary for describing workflows, and a vocabulary for describing provenance. We provide here a summary of the RO Model and its accompanying documentation (the RO Primer), and highlight the services developed for creating and management of Research Objects: the Research Object Storage and Retrieval Service and a command line tool. These components and services are described in greater detail in D1.2v2 and D1.4v1.

We also discuss preliminary work in defining a model for workflow abstraction, with the aim of supporting reuse of workflows (or their constituent parts), and present an initial characterisation of workflow decay, identifying causes of workflow decay based on an analysis of existing workflows in the myExperiment repository.

This deliverable should be read in tandem with D1.4v1, D1.2v2, D3.2v1 and D4.2v1 in order to provide a complete picture of the state of the Wf4Ever Phase I components.

Contents

List of Tables

List of Figures

1 Introduction

This deliverable describes aspects of Phase I of the design, implementation and deployment of the Wf4Ever components that will support workflow lifecycle management. The document should be read in tandem with other Month 20 deliverables, in particular D3.2v1 [GC12b] and D4.2v1 [GC12a] which adress complementary aspects of the overall wf4ver architecture and components.

According to the Description of Work, *This prototype will include the following functionalities: an initial Research Object model, implemented by means of an ontology network, and basic management functions (storage and access), validation functionalities based on RO provenance, and definition of semantic overlays and workflow provenance matching techniques for abstraction.*.

These requirements are addressed in the following way:

Sections 2, 3 and 4 discuss the Research Object Model defined within Wf4Ever along with a Primer document providing an introduction to that model and a collection of example Research Objects.

Sections 5 and 6 describe the initial Research Object Storage and Retrieval Service and Command Line Manager. Both of these tools use the Research Object Model to structure the objects that they produce and consume. The RO Model is thus the "glue" that joins together the components and enables interoperation.

Section 7 discusses an initial model for workflow abstraction, while Section 8 presents a characterisation of workflow decay.

Note that this document represents the results from Phase I of the project – as a result, some areas are not yet complete and we expect updates, changes and extensions to be reported in Phase II of the project, due for completion in M32. For example, we expect that RO models reported here will be subject to change following further usage and experience, both within and outside the project.

2 The Research Object Model

The Wf4Ever Research Object model defines vocabularies that describe Workflow-centric Research Objects within Wf4Ever. A complete, functional RO based ecosystem also requires a number of different services for creation, storage, manipulation, recommendation, visualisation etc. of Research Objects. These services are not considered as part of the model (although implemented services are described in this deliverable in Sections 5 and 6). Nor does the core model describe the evolution of Research Objects (ROs) – this is discussed in D3.2v1 [GC12b].

Narrative text describing the model is published online[1] and the ontologies themselves also available[2]. We will not reproduce all this narrative text here, but provide an overview of the model and rationale.

A simple schematic of an RO is shown in Figure 1. The RO contains a workflow, input data and results along with a paper that presents the results and links to the investigators responsible. Annotations on each of the resources (and on the RO itself) provide additional information and characterise, e.g. the provenance of the results (the results were obtained by executing the workflow on the input data).

Research Objects play multiple roles. In the first case, they are *technical* objects. They provide access to the resources that are needed to support execution of investigations and record the provenance traces of those executions. They encapsulate dependencies between resources and maintain versioning information about the lineage and evolution of those resources.

At the same time, they are *social* objects. They encapsulate reusable protocols and know-how. They record best practices and support reproducibility. They are citable artifacts that can be referred to and quoted. They record and represent information about the people involved in investigations – those who create, use, extend and curate the objects.

These roles bring requirements on the representation structure and vocabularies used to describe Research

[1] http://wf4ever.github.com/ro/
[2] https://github.com/wf4ever/ro/

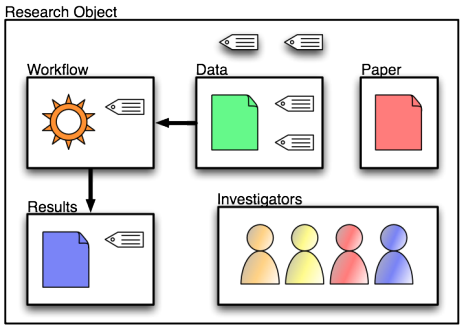


Figure 1: RO Schematic

Objects. The specification of the RO model focuses on the technical aspects. It describes the core Wf4Ever Research Object vocabularies that provide container structures and vocabulary for describing workflow objects. Additional vocabularies covering evolution, lifecycle, versioning and other social aspects will be covered elsewhere.

ROs allow for the aggregation of resources along with annotations on those resources concerning their provenance, use, characteristics etc. Aggregation is supported through the use of the OAI-ORE vocabulary[3] and annotation is supported through the use of the Annotation Ontology[4]. Re-use of these existing vocabulary will facilitate third party tools in understanding and processing Research Objects described using the model.

Finally, the RO Model provides a number of basic ontologies that are used within this aggregation/annotation framework to describe specifics of the Workflow-centric Research Objects. These are:

ro Provides basic structure for the description of aggregated resources and the annotations that are made on those resources.

wfdesc A vocabulary for the description of workflows. This provides an abstraction that can be mapped to different particular workflow systems. The ontology is intended as an upport ontology for more specific workflow definitions, and as a way to express abstract workflows, which could either be hand-crafted by users, or extracted from workflow definitions, for example Taverna's `t2flow` or Scufl2 formats. A prototype service that transforms workflows into Research Objects, using the wfdesc ontologies has been developed (See [PH12]).

wfprov A vocabulary for the description of provenance information. This provides an abstraction that can be mapped to different provenance vocabularies, for example PROV-O[5] as developed by the W3C Provenance Working Group.

3 Research Objects Primer

The Research Object Ontologies and Vocabularies Primer is a document targeted at users, providing an accessible introduction to the Wf4Ever RO Model. This will enable readers to understand *what* the RO Model provides and *how* the RO Ontologies and Vocabularies can be used to describe an aggregation object that represents scientific experiments in a structured format.

The document is published online[6]

http://purl.org/net/wf4ever/ro/repeatability http://purl.org/net/wf4ever/ro/reproducibility

[3] OAI-ORE is not a ratified standard produced by a body such as the W3C or IETF, but it is becoming widely used as a vocabulary for aggregation.

[4] Again, AO is not as yet a standardised vocabulary, but a W3C community group has been set up to oversee the drafting of a specification: `http://www.w3.org/community/openannotation/`

[5] `http://www.w3.org/TR/prov-o/`

[6] `http://wf4ever.github.com/ro-primer/`

4 Research Object Examples

A number of exemplar Research Objects have been described. These provide illustrative examples of how the model may be used to describe aggregations of content.

4.1 Astrophysical Quantities

This RO collects together several resources including input and output datasets, scripts, web services and other documents. These relate to various tasks and stages of the experiment including *gathering*, *propagation* and *comparison* with intermediate results being passed from one stage to another.

A screenshot of the RO within the RODL is shown in Figure 2. This shows the conceptual view of the RO, with folders containing Workflow Runs expanded.

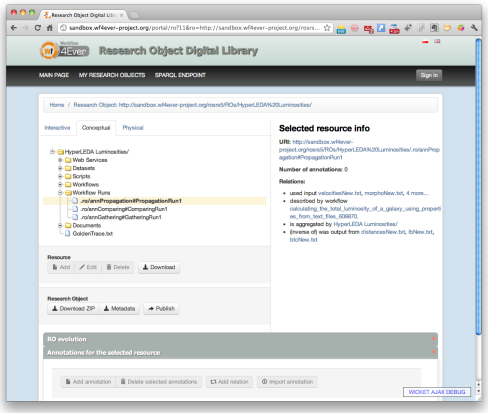


Figure 2: HyperLEDA Luminosities Example RO

We can see here the relationships between a particular workflow run aggregated in the RO and its input, the workflow executed etc. These relationships are described using the RO model vocabulary, and the portal allows for export/publication of this metadata.

The RO is available in the RODL portal[7]

4.2 InterProScan

This is an example of an RO built around a workflow taken from myExperiment. The workflow performs an InterProScan analysis of a protein sequence using the EBI's WSInterProScan service[8]. The workflow illustrates the issue of workflow *decay* as it can no longer be enacted. This is because the workflow involves EBI asynchronous services that were suspended as the EBI changed the way asynchronous services are handled.

Again, a snapshot of the RO in RODL is shown in Figure 3. Here, we can see annotation applied to the workflow within the RO, in particular a description discussing the problems with the workflow.

The RO is available in the RODL portal[9]

4.3 Repeatability and Reproducibility

Two example ROs illustrate how different levels of information can be recorded within the ROs in order to support a rerunning of an experiment. The left of Figure 4) includes a workflow (along with its abstract

[7] http://purl.org/net/wf4ever/ro/HyperLEDA_Luminosities
[8] http://www.ebi.ac.uk/Tools/webservices/services/interproscan
[9] http://purl.org/net/wf4ever/ro/InterProScan_RO1

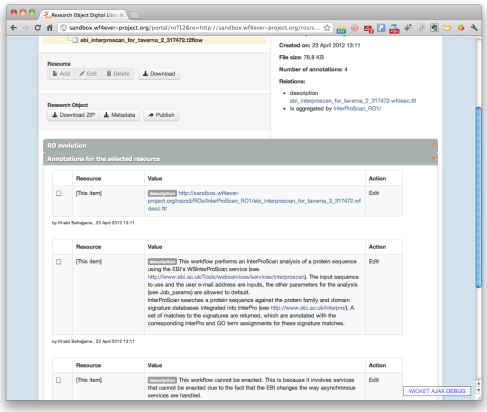


Figure 3: InterProScan Example RO

description using the **wfdesc** ontology). To the right, the RO also includes details of a workflow execution or run, using the **wfprov** ontology.

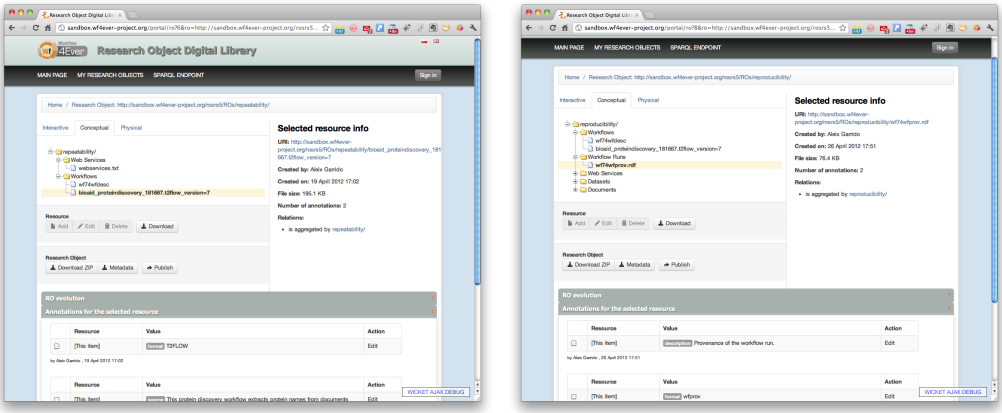


Figure 4: RO with Workflow description and provenance

The ROs are available in the RODL portal[10]

5 Research Object Storage and Retrieval Service

The Wf4Ever Research Objects Digital Library (RODL) will provide a number of services that support the creation, management and manipulation of Research Objects (ROs). Among these services is Storage and Retrieval – referred to here as ROSRS.

The ROSRS is provided as a RESTful interface, with the following functionality:

- Storing and retrieving research objects;
- Storing and retrieving resources aggregated within the research objects;
- Annotating the aggregated resources, including the research object itself.

The ROSRS uses the RO Model (as discussion in Section 2 to structure and describe the objects it creates. Further information describing the details of the ROSRS API are contained in D1.2v1 [PH12].

[10] `http://purl.org/net/wf4ever/ro/repeatability` and `http://purl.org/net/wf4ever/ro/reproducibility`

6 Research Object Manager

The Research Object Manager provides a command line tool for creating, displaying and manipulating Research Objects. The RO Manager functionality is complementary to that provided by the ROSRS described in Section 5. In particular, the RO Manager is primarily designed to support a user working with ROs in the host computer's local file system, with the intention being that the ROSRS and RO Manager can exchange ROs between them – in part facilitated by the use of the shared RO vocabulary and model.

Past experience has suggested that lightweight, command line tools give users early access to functionality and provide an opportunity to gather additional feedback and requirements on that functionality. Command line tools can also be used with built in operating system functionalty as pipes and input/output redirection in order to quickly build prototype tool chains.

The RO Manager allows users to

- Create local ROs;
- Add resources to an RO;
- Add annotations to an RO;
- Read and write ROs to the RODL.

As with the ROSRS, the RO Manager uses the RO Model to structure and describes the objects it creates.

Further information describing the details of the RO Manager are contained in D1.2v1 [PH12].

7 Workflow Abstraction

The main goal of making an abstraction of a workflow is to be able to reuse it as a whole, or some of its parts, independently of the domain to which they belong to. In this context the concept of classification is crucial because being able to classify a workflow or its parts allows also to index them for easier accesibility.

In myExperiment or WINGS the processes of the workflows are not annotated and therefore the abstraction of the different parts of it is not straightforward. Therefore, we have choosen a bottom-up approach in order to study the actual provenance of workflow results (which represents the dataflow) from a set of available workflows at WINGS (@@todo add reference to dataset examples) trying to get some common structures (so called macros) for being able to categorize them and create their taxonomy. The use of the provenance of the workflow results seems to be more appealing that using the workflow templates mainly due to the following two reasons:

- It actually gathers information from the workflows which really are being used and executed and therefore rewarding the discovery of common patterns of those workflows which really work.
- It provides the sequence of processes (including their input parameters and their outputs) instead of having the workflow templates which are represented by acyclic graphs. This allows the dissambiguation of the different possible ways representing a workflow by its actual execution trace (p.e. different possibilities due to "if" control structures).

Because of the above mentioned reasons, the provenance of workflow results data is being used for feeding a *trie* structure (@@todo add reference to the code in Github) created to store the different executions of the workflows. This *trie* structure provides the following funcionality:

- It stores the provenance of the workflow results in an ordered way and the appearance frequency of their processes

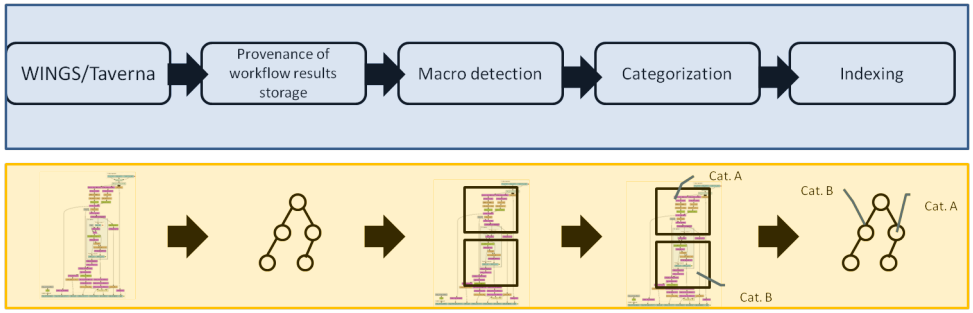


Figure 5: Workflow Abstraction Discovery Process

- It calculates relative frecuencies at different levels of the tree
- It provides different modes to traverse the structure (p.e. preordered)
- It provides an output XML strcuture with relative frecuencies per level and per process (an output example is available at @@todo include reference).

The next picture 7 shows the overall discovery process for workflow abstraction. At present we are currently in the second stage which includes the creation of a provenance of workflow results and the *trie* structure explained above.

8 Characterising Workflow Decay

One of the main impediment to workflow preservation is workflow decay. Indeed, our experience with scientific workflows suggests that a large proportion of workflows suffer from decay, which decreases their value over time. Broadly speaking, we can distinguish two forms of decay. i)- **Inability to re-execute the workflow**: due to many factors, including the unavailability of third party resources that are responsible for executing the tasks that compose the workflows, we may not be able to re-execute a given workflow. ii)- **Inability to reproduce workflow results**: a less-sever, yet relevant, form of decay, is the inability to obtain the same (or similar) results when re-executing the workflow. Reproducing previous results can be primordial in reinforcing trust in scientific results and can be used in peer-reviewing as a mechanism to validate the results claimed by given scientists [FBS11, GRB12].

The above discussion raises the following question. *What are the causes of workflow decay?* To identify and characterise the causes of workflow decay, we adopted a bottom-up approach, whereby we manually analyzed 92 workflows from the myExperiment repository. We chose Taverna workflows because this is the largest available workflow collection (more than half of the workflows in myExperiment are Taverna workflows at the time of writing) and Taverna workflows have been published on myExperiment since its launch in 2007, therefore providing a good insight into decay over those years. To base our analysis on a sample of workflows that is representative of the set of workflows in myExperiment, we tried to select workflows by three criteria: 1) the year they were created, 2) the creator of the workflows; and 3) the domain studied by the workflows. We believe that the decay of workflow could be directly impacted by the year they were created, hence we tried to make an even coverage of T1/T2 workflows between the years 2007 and 2012. In order to reduce possible bias introduced by the specific workflow creators, we avoided choosing workflows created by the same person in the same year. Our workflow selection also had a good coverage of domains, covering 18 different scientific (such as life sciences, astronomy, or cheminformatics) and non-scientific domains (such as testing of Grid services). Figures 6 illustrates the domains of the workflows that we analyzed.

Although in this paper we focus on a particular family of workflows, we expect our approach and analysis to be applicable to many others and our analysis to be repeatable on a different corpus of workflows. This

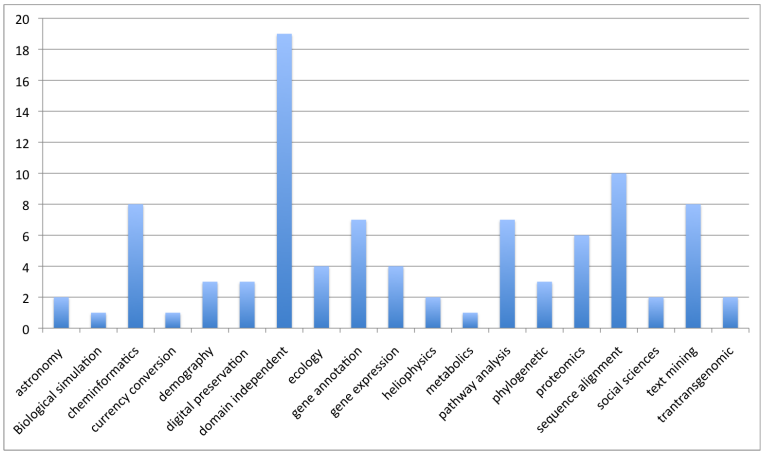
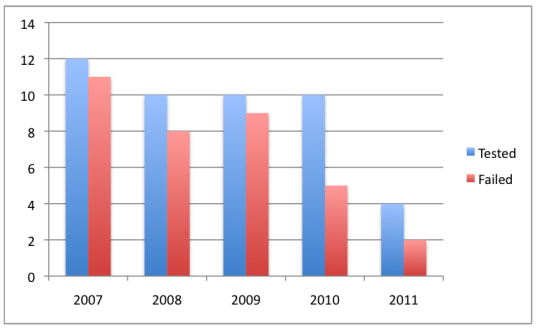


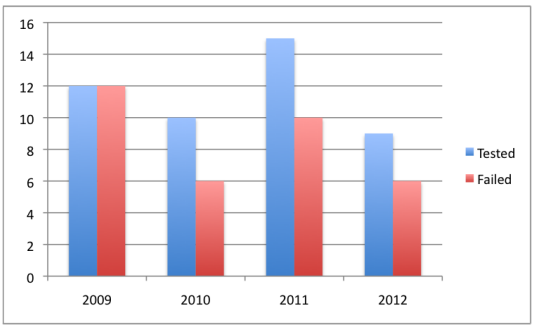
Figure 6: Distribution of the domain studied by our test workflows.

is to our knowledge the first study of its kind. The workflows that we selected for analysis were created by different users.

To identify the causes of decay that these workflows may suffer from, we attempted to execute them using the Taverna 2.3 workbench. We then manually examined their results, diagnosed broken links, etc. Our analysis showed that nearly 80% of the tested workflows failed to be either executed or produce the same results (if testable), and those from earlier years (2007-2009) had more than 80% failure rate (as shown in Figure 7. The causes of workflow decay can be classified into four categories, which we present in the rest of this section.



(a) Number of Taverna 1 workflows tested and failed between 2007-2009.



(b) Number of Taverna 2 workflows tested and failed between 2009-2012.

Figure 7: Number of Taverna workflows tested and failed

8.1 Volatile third-party Resources

Most of the workflows that we analysed make use of third-party resources such as web services and databases, e.g., the KEGG services used in our example workflow provided by the Data Bank of Japan. The provision of such resources may be interrupted or changed, causing failure of the workflow to execute. In certain cases, the workflow cannot be run, even when the third party resources that it relies on are available, e.g., when such resources require authentication. Another cause that may lead to workflow decay, is changes to third party resources. For example, if the web service provider decides to change the implementation of the web service, then the workflow execution may not deliver the same results, or worse, it may not be possible to execute. Table 1 summarises these causes of decay with concrete examples.

Table 1: Categorisation of Decay Caused by Third-party Resources

Causes	Refined causes	Examples
Third party resources are not available	Underlying dataset, particularly those locally hosted in-house dataset, is no longer available	Researcher hosting the data changed institution, server is no longer available
	Services are deprecated	(DNA Data Bank of Japan) DDBJ web services are not longer provided despite the fact that they are used in many myExperiment workflows
Third party resources are available but not accessible	Data is available but identified using different IDs that the one known to the user	Due to scalability reasons the input data is superseded by new one making the workflow not executable or providing wrong results
	Data is available but permission, certificate, or network to access it is needed	Cannot get the input, which is a security token that can only be obtained by a registered user of ChemiSpider
	Services are available but need permission, certificate, or network to access and invoke them	The security policies of the execution framework are updated due to new hosting institution rules
Third party resources have changed	Services are still available by using the same identifiers but their functionality have changed	The web services are updated intentionally or unintentionally (e.g.malware) providing wrong results

8.2 Missing example data

It is not always obvious which data can be used as inputs to the workflow execution, and example inputs are often most helpful. Example outputs can also be useful to gain an insight of the outcome anticipated from the workflow. However, our analysis revealed that they are not always made available. Provenance traces of previous runs are also useful as indications of where example data may be found.

8.3 Missing execution environment

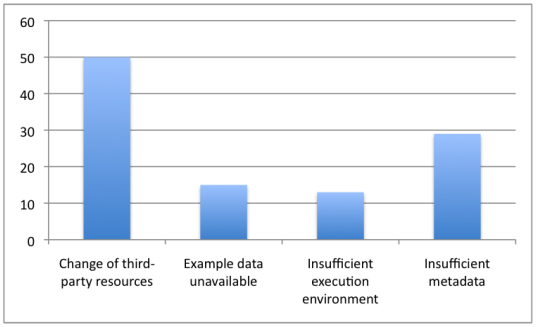
The execution of a workflow may rely on a particular local execution environment, for example, a local R server or a specific version of workflow execution software. Some of our test workflows exhibit this type of decay. Taverna often provides sufficient information about missing libraries, and sometimes workflow descriptions provide a warning about the requirement for a specific library. This type of decay appears to be fixable by installing the missing software, albeit requiring some effort.

8.4 Insufficient descriptions about workflows

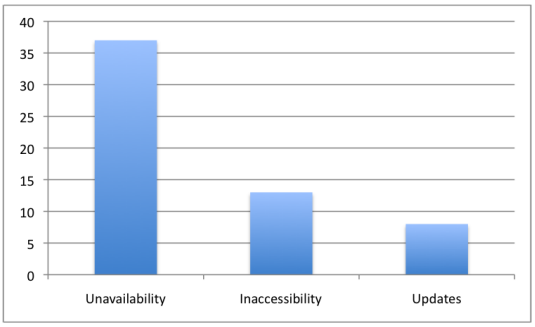
Sometimes a workflow workbench cannot provide sufficient information about what caused the failure of a workflow run. Additional descriptions in the workflow can play an important role in assisting re-users to understand the purpose of the workflow and its expected outcomes.

8.5 Summary

The results of our analysis are summarised in Figure 8-a in Appendix IV, which illustrates the number of workflows that suffer from each of the causes of decay presented above. It shows that 50% workflows suffer from decay due to third party resources. However, we might draw a different conclusion if a bigger corpus or a different collection of workflows are used.



(a) A summary of workflow decay causes.



(b) Workflow decay due to third party resources.

Figure 8: Results of workflow decay analysis.

To better understand the causes of decay due to third party resources. Figure 8-b shows the number of workflows that suffer from the causes presented in Table 1. It shows that the unavailability of third party resources is the leading cause of decay, followed by resources inaccessibility, and then resources changes.

The above results confirms our hypothesis that workflow decay is a serious problem. It also suggests that there is a need for additional information that can assist workflow designers detecting and repairing decayed workflows. We report in a separate deliverable [**?**], on a minimal model and associated checklists, that were designed and developed with the objective to prevent and repair workflow decay.

References

[FBS11] Juliana Freire, Philippe Bonnet, and Dennis Shasha. Exploring the coming repositories of reproducible experiments: Challenges and opportunities. *PVLDB*, 4(12):1494–1497, 2011.

[GC12a] Esteban García Cuesta. Design, implementation and deployment of Workflow Integrity and Authenticity Maintenance components - Phase I. Deliverable D4.2v1, Wf4Ever Project, 2012.

[GC12b] Rafael González-Cabero. Design, implementation and deployment of Workflow Evolution, Sharing and Collaboration components - Phase I. Deliverable D3.2v1, Wf4Ever Project, 2012.

[GRB12] Carole A. Goble, David De Roure, and Sean Bechhofer. Accelerating scientists' knowledge turns. In *Proceedings of The 3rd international IC3K joint conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management.*, 2012. in press.

[PH12] Raul Palma and Piotr Holubowicz. Reference Wf4Ever Implementation - Phase I. Deliverable D1.4v1, Wf4Ever Project, 2012.