Wf4Ever: Advanced Workflow Preservation Technologies for Enhanced Science

STREP FP7-ICT-2007-6 270192

Objective ICT-2009.4.1 b) – "Advanced preservation scenarios"

D4.2: Design, implementation and deployment of Workflow Integrity and Authenticity Maintenance components – Phase II

Deliverable Co-ordinator: Esteban García-Cuesta

Deliverable Co-ordinating Institution: iSOCO

Other Authors: Graham Klyne (OXF), Esteban García-Cuesta (iSOCO), Aleix Garrido (iSOCO), Jose Manuel Gómez-Pérez (iSOCO), Jun Zhao (OXF).

This document describes the second phase of delivery of Integrity and Authenticity components implementation. It includes the latest updates on provenance models, their standardization, and the community building, and also describes the updates of completeness, stability, and the description of the new implemented dimension reliability for overall evaluation of the quality of a research object.

Document Identifier:	Wf4Ever/2013/D4.2v2/v1.0	Date due:	31/07/2013
Class Deliverable:	Wf4Ever 270192	Submission date:	31/07/2013
Project start date:	December 1, 2010	Version:	v2.0
Project duration:	3 years	State:	Final
		Distribution:	Public

Wf4Ever Consortium

This document is a 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. Edificio Testa Avda. del Partenón 16-18, 1º, 7ª Campo de las Naciones, 28042 Madrid Spain Contact person: Dr. Jose Manuel Gómez-Pérez E-mail address: jmgomez@isoco.com	**University of Manchester** Department of Computer Science, University of Manchester, Oxford Road Manchester, M13 9PL United Kingdom Contact person: Professor Carole Goble E-mail address: carole.goble@manchester.ac.uk
Universidad Politécnica de Madrid Departamento de Inteligencia Artificial Facultad de Informática, UPM 28660 Boadilla del Monte, Madrid Spain Contact person: Dr. Oscar Corcho E-mail address: ocorcho@fi.upm.es	**University of Oxford** Department of Zoology University of Oxford South Parks Road, Oxford OX1 3PS United Kingdom Contact person: Dr. Jun Zhao / Professor David De Roure E-mail address: {jun.zhao@zoo.ox.ac.uk, david.deroure@oerc.ox.ac.uk}
Poznań Supercomputing and Networking Center Network Services Department Poznań Supercomputing and Networking Center Z. Noskowskiego 12/14, 61-704 Poznan Poland Contact person: Dr. Raúl Palma de León E-mail address: rpalma@man.poznan.pl	**Instituto de Astrófísica de Andalucía** Dpto. Astronomía Extragaláctica Instituto Astrofísica Andalucía Glorieta de la Astronomía s/n 18008 Granada, Spain Contact person: Dr. Lourdes Verdes-Montenegro E-mail address: lourdes@iaa.es
Leiden University Medical Centre Department of Human Genetics Leiden University Medical Centre Albinusdreef 2, 2333 ZA Leiden The Netherlands Contact person: Dr. Marco Roos E-mail address: M.Roos1@uva.nl	

Change Log

Version	Date	Amended by	Changes
0	22/05/2013	Esteban García-Cuesta	Outline included
0.1	04/07/2013	Esteban García-Cuesta	Initial draft included by adding information provided by the different authors
0.2	04/07/2013	Aleix Garrido	Section 5 improvements and format corrections
0.3	15/07/2013	Esteban García-Cuesta	Improvements on the contents of sections 2,3,4,5
0.4	17/07/2013	Esteban García-Cuesta	Conclusions added
0.5	18/07/2013	Graham Klyne	Review pass and editing of executive summary
0.6	18/07/2013	Esteban García-Cuesta	Review pass
0.7	29/07/2013	Esteban García-Cuesta	Review and updates including the QA comments
0.8	29/07/2013	Esteban García-Cuesta	Changing section 5 subsections names and some contents for better understanding
0.9	-	-	-
1.0	31/07/2013	Esteban García-Cuesta	Final version of the document

Executive Summary

This is the last of two deliverables regarding the design, implementation, and deployment of Workflow Integrity and Authenticity and it includes the **prototypes of the different integrity and authenticity (I&A) components** of the project and provides a summary of the updated implementation developed during the phase II. This is the last deliverable of two, and is based on the previous work (following incremental project development). As such, it contains material or references which were already presented previously but which we have decided to include here for ease of understanding and to make this document self-contained.

Updates to work on I&A evaluation are described, based on quality evaluation of a research object (RO), which has been driven by evaluation of **completeness** and **stability**. Further, we include a definition and implementation of a new dimension called **reliability**, which makes use of the history of completeness information for providing a more user oriented and meaningful information regarding the quality of an RO.

This document also describes our **provenance work** in which we have made substantial contributions to the **standardization of provenance in the World Wide Web Consortium (W3C)**, and have created a significant corpus of provenance data, called **ProvBench**, which makes use of standard provenance ontologies, project specific refinements of these, and updated provenance exporting plugins initially implemented during the Phase I.

Finally, some updates on the current design and implementation of the dimensions completeness, stability, and reliability are presented highlighting their improvements, their APIs, and the tools which have been built using these ideas to enhance functionalities within Wf4Ever portal.

This deliverable should be read together with D1.4v2, D2.2v2, and D3.2v2 in order to obtain a complete overview of the current state of the components implemented during phase II of the Wf4Ever project.

Table of contents

List of Figures

1 Introduction

This document provides a description of the software components produced during phase II of Wf4Ever in the context of WP4 (workflow integrity and authenticity maintenance) which are the result of the incremental work done throughout the project. These components makes use of the research object resources in RODL (research object Digital Library) and the models used for the definition of a research object to evaluate the overall quality of a research object (RO), and provide insight into its current status (e.g. by showing explanations).

According to the Description of Work this deliverable: "will include the following functionalities: an updated research object provenance model that is the basis of the standardisation process in existing international initiatives, and extended methods for computing integrity and authenticity, taking into account different granularities, and visualisation tools for them".

Due to the advance state of the provenance vocabularies, wfprov and roevo, and the fact that they were early available to the project (a detailed description of these models can be found at [D2.2v1] and [D3.2v1]), several members of the Wf4Ever team were able to joint and made significant contributions to the World Wide Web Consortium (W3C) effort to create a standard for provenance, including as lead editors for several documents. This has also allowed the creation of a PROV-Corpus during the phase II of this work, so called ProvBench, which is based on Taverna and Wings workflow repositories and uses the wfprov ontology and the updated exporting Taverna plugin to export the provenance of workflow results from Taverna format to Wf4Ever wfprov ontology.

The resulting corpus has also been made accessible to the community for the main purpose of providing a suitable number of provenance of workflow results samples for benchmarking (e.g. extraction of macros, or identification of similar workflows based on their provenance of workflow results). It is worth highlighting that the above mentioned standardization effort was completed in May of 2013[1].

The main improvements about the I&A work are: i) construction of a new Minim model based on further exposed requirements, ii) design of new checklists, iii), updated version

[1] http://www.w3.org/2011/prov/wiki/Main_Page

of the evaluation completeness component to use SPARQL1.1 standard, iv) access to RODL repository for the ROs to be evaluated, v) implementation of a new “reliability” dimension , vi) new presentation tools for the completeness, stability, and reliability dimensions, viii) storing and providing accessibility to the history of the quality results as an aggregated resource (using ORE vocabulary) of the RO, and ix) new presentation tools for providing quality information of a RO to end-users focused on availability and reuse of a research object.

Among other information, we would highlight the importance of collecting the history of the different quality scores to provide a longer term view of the RO stability and reliability.

Furthermore, we started the evaluation process which will be also finished before M36 and fully included in the deliverable D4.3. “Final evaluation report of the workflow integrity and authenticity maintenance”.

The remainder of this document is structured as follows. Section 2 presents the provenance models, the applications which have been implemented using provenance, and the community building effort around provenance which includes the W3C PROV standardization[2]. Section 3 describes the general framework for evaluating the quality of a RO describing the interaction between the models (qualitative information) and the presentation tools (quantitative information) and how the last are based on the first to define the scores for the three dimensions completeness, stability, and reliability. Sections 4 (completeness dimension) and 5 (stability and reliability dimensions) present our current design and implementation of the I&A evaluation components and how they have been integrated with other components of the project in the context of the Wf4Ever architecture. Finally Section 6 presents our conclusions including a summary of this work and our plan for the next phase of the project (M36).

1.1 Technical Context

During the implementation of the integrity and authenticity components we have made some decisions about the technical environment within which Wf4Ever is being deployed. These are:

[2] http://www.w3.org/TR/prov-overview/

- The system operates in the environment of the World Wide Web, supporting normal Web capabilities of retrieval, linking, etc. As such, URIs are used to denote arbitrary concepts, object types, etc. Concepts and entities manipulated by Wf4Ever are preferably identified using URIs
- Interfaces of the developed components have used HTTP/RESTful interfaces.
- research objects (RO) are the central digital information structure used to represent a scientific experiment and its context.
- Among other things, an RO contains metadata about provenance of its lifecycle, and also about the execution of any workflows that it uses.
- The provenance information has been modelled by the evolution ontology (roevo) and the provenance of workflow results ontology (wfprov), which are based on the W3C PROV ontology, and represented using OWL[3].

1.2 Relation with Other WPs

Our work in WP4 about integrity and authenticity evaluation relies on different aspects that are treated elsewhere in the project. The main information units under study are ROs, whose representation is treated as part of WP2 work. Likewise, aspects about provenance dealing with RO evolution and versioning are addressed in combination with WP3. On the other hand, the evaluation of RO integrity and authenticity provides end users in WP5 and WP6 with valuable criteria to get some insight on the quality of ROs for the main purposes of availability and reuse. There is also a strong interaction with the overall integration of Wf4Ever project elements, along with user interfacing aspects and RO presentation, addressed in WP1. Therefore, for a better understanding of the document we recommend it be read together with deliverables produced by other technical WPs, including D1.4v2 [D1.4v2], D2.2v2 [D2.2v2], and D3.2v2 [D3.2v2].

[3] http://www.w3.org/TR/owl-ref/

2 Provenance

Provenance collects information about entities, activities, and people involved in producing a piece of data, which can be used to form assessments about its overall quality, reliability or trustworthiness. An overview of the W3C PROV family of provenance information specifications can be found at PROV-Overview[4].

2.1 Provenance in Wf4Ever

In Wf4Ever there are two main types of provenance which have been modelled and used:

- **Provenance of workflow results**: providing a trace of the workflow processes, data resources and associated metadata that were used to produce the result of a workflow execution, and
- **RO Evolution**: as an underpinning for the representation of research object evolution (roevo), describing the evolution of research objects over time, providing a record of the changes applied at the different stages of their lifecycle.

The provenance of artefacts created by a workflow execution is captured during execution of a workflow by the workflow execution engine, and is published as annotations in a workflow RO using the Annotation Ontology (AO). This provenance is expressed using the wfprov ontology[5], which is part of the RO Model[6] which also is defined as a refinement of the W3C PROV-O ontology[7]. A detailed description about the wfprov ontology can be found at the RO model specification page[8].

The provenance of the research object evolution, along with its possible origins in previous work, is captured through the research object Digital Library RODL[9], and

[4] http://www.w3.org/TR/2012/WD-prov-overview-20121211/

[5] https://github.com/wf4ever/ro/blob/master/wfprov.owl

[6] http://wf4ever.github.io/ro/

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

[8] http://wf4ever.github.io/ro/

[9] http://www.wf4ever-project.org/wiki/display/docs/Research+Objects+Digital+Library+%28including+the+ROSRS%29

keeps track of the lifecycle of an RO. This provenance is represented using the roevo ontology[10] which also is defined as a refinement of the W3C PROV-O ontology.

2.2 Provenance information in research objects

Representing provenance in ROs

To record provenance information in ROs we have used semantic annotations following the Annotation Ontology standard [Cicca'11], which is a central part of the RO model. That is, the RO includes RDF metadata resources, containing provenance information, and these are identified as annotations of corresponding target resources by statements in the RO manifest. The Figure 1 shows the provenance of workflow results where the arrow labelled "RDF graph references" indicates that the provenance data contains direct references to the resource whose provenance is described. One such provenance resource may describe provenance of multiple target resources, and is self-describing concerning the resources whose provenance is provided.

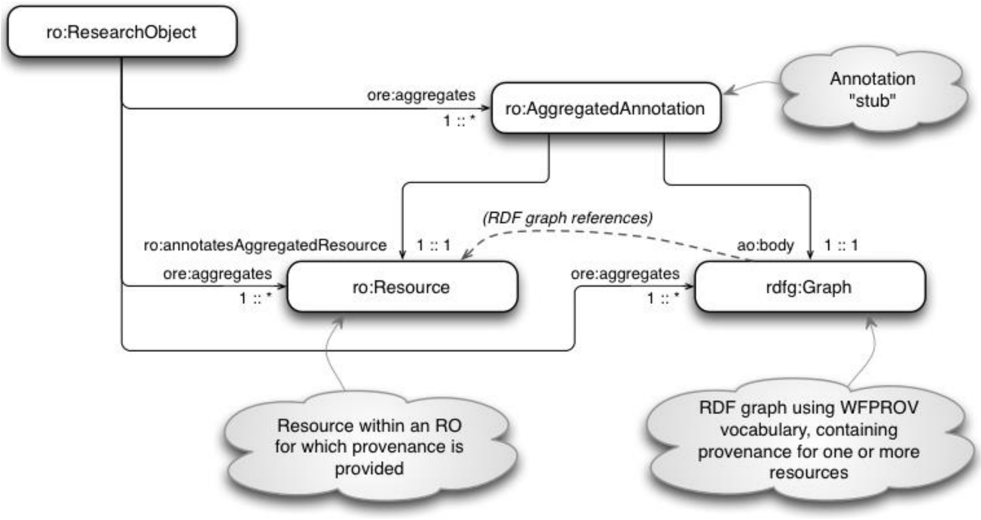


Figure 1 Provenance of workflow results

The second type of provenance associated with research objects is research object Evolution (roevo). This is expressed using a similar approach to that shown above, but with provenance relationships described between ROs, rather than between resources aggregated by an RO. Here, the roevo provenance resources capture the evolutionary relationships between a *Live* RO and its *Snapshots* or *Archives* states, and the forward

[10] https://github.com/wf4ever/ro/blob/master/roevo.owl

looking relations are colour coded in blue, and the historical provenance relationships are coloured in red as can be seen in Figure 2. In the next section we describe how this provenance is accessed.

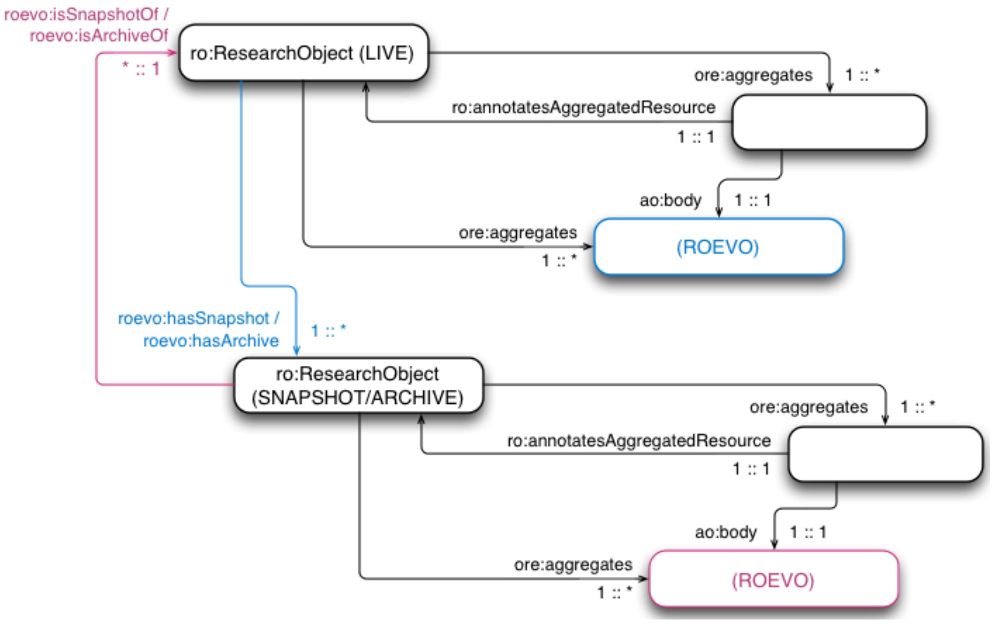


Figure 2 roevo provenance diagram

2.3 Accessing provenance in ROs

Accessing provenance in an RO generally involves first reading the RO manifest, which contains the information described in the Figure 1 and Figure 2. The RO manifest information is then used to locate descriptions of the RO and its resources, which may include provenance and other information. The relevant information is read as one or several RDF graphs (annotations), from which the desired provenance information can be extracted. For example, the checklist service reads all the annotations mentioned in the RO manifest, and creates a single RDF merged annotation graph of the entire provenance and other information thus obtained. Provenance information can then be tested by suitably constructed SPARQL queries that are evaluated against the merged annotation graph.

Other applications may choose to be selective about the annotations they read, selecting those that are indicated in the RO manifest as having relevance to a particular target resource of interest.

2.4 Taverna provenance export tools

Taverna[11] executes workflows and can capture provenance of workflow results, including individual processor iterations and their inputs and outputs. This provenance is kept in an internal database, which is used within the Taverna workbench to present information about previous runs and intermediate results. Figure 3 shows the current Taverna provenance architecture.

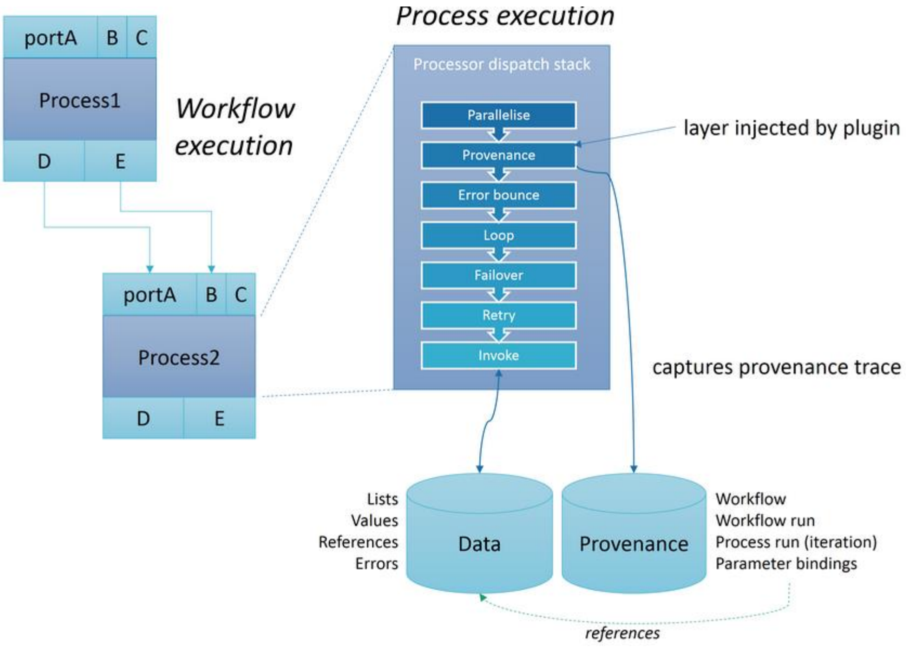


Figure 3 Taverna provenance architecture.

During execution of a Taverna workflow, the dispatch stack[12] is responsible for the execution logic of an individual process invocaton, with layers like *parallelise* and *retry*. By injecting a provenance layer towards the top of the stack, a trace of each execution can be captured and stored in an internal provenance database. This includes a copy of the workflow definition, start/stop times for the workflow run and for each process execution. In addition the input and output parameters for every workflow and process execution is captured as references to Taverna's internal data store.

[11] http://www.taverna.org.uk/

[12] http://www.taverna.org.uk/api-2.3/net/sf/taverna/t2/workflowmodel/processor/dispatch/DispatchStack.html

The provenance trace has been used by the implemented Taverna-PROV plugin[13] to export the workflow run according to the PROV-O standard. This includes the output and intermediate values, and the provenance trace as a PROV-O RDF graph[14] and a directory structure of the contents as individual files. The graph contents can be queried using SPARQL and processed with other PROV tools, such as the PROV Toolbox[15].

The used Taverna-PROV ontology[16] for exporting this provenance traces extends the Wf4Ever wfprov ontology, which is based on PROV-O. Therefore no transformation (beyond OWL reasoning) has been required within Wf4Ever to understand the created Taverna-PROV traces and for using them.

A complete description of the interaction between the different implemented parts for exporting the provenance of workflow results of a Taverna workflow is shown in Figure 4,and some examples of provenance traces, in addition to installation and usage instructions for the Taverna PROV export plugin are available at the taverna-prov project at GitHub[17]. We also want to point out that the Taverna provenance support was key for generating the PROV-corpus as explained in the ProvBench Challenge part of Section 2.6.

[13] https://github.com/wf4ever/taverna-prov

[14] http://www.w3.org/TR/2013/REC-prov-o-20130430/

[15] https://github.com/lucmoreau/ProvToolbox/

[16] https://raw.github.com/wf4ever/taverna-prov/master/prov-taverna-owl-bindings/src/main/resources/org/purl/wf4ever/provtaverna/taverna-prov.ttl

[17] https://github.com/wf4ever/taverna-prov

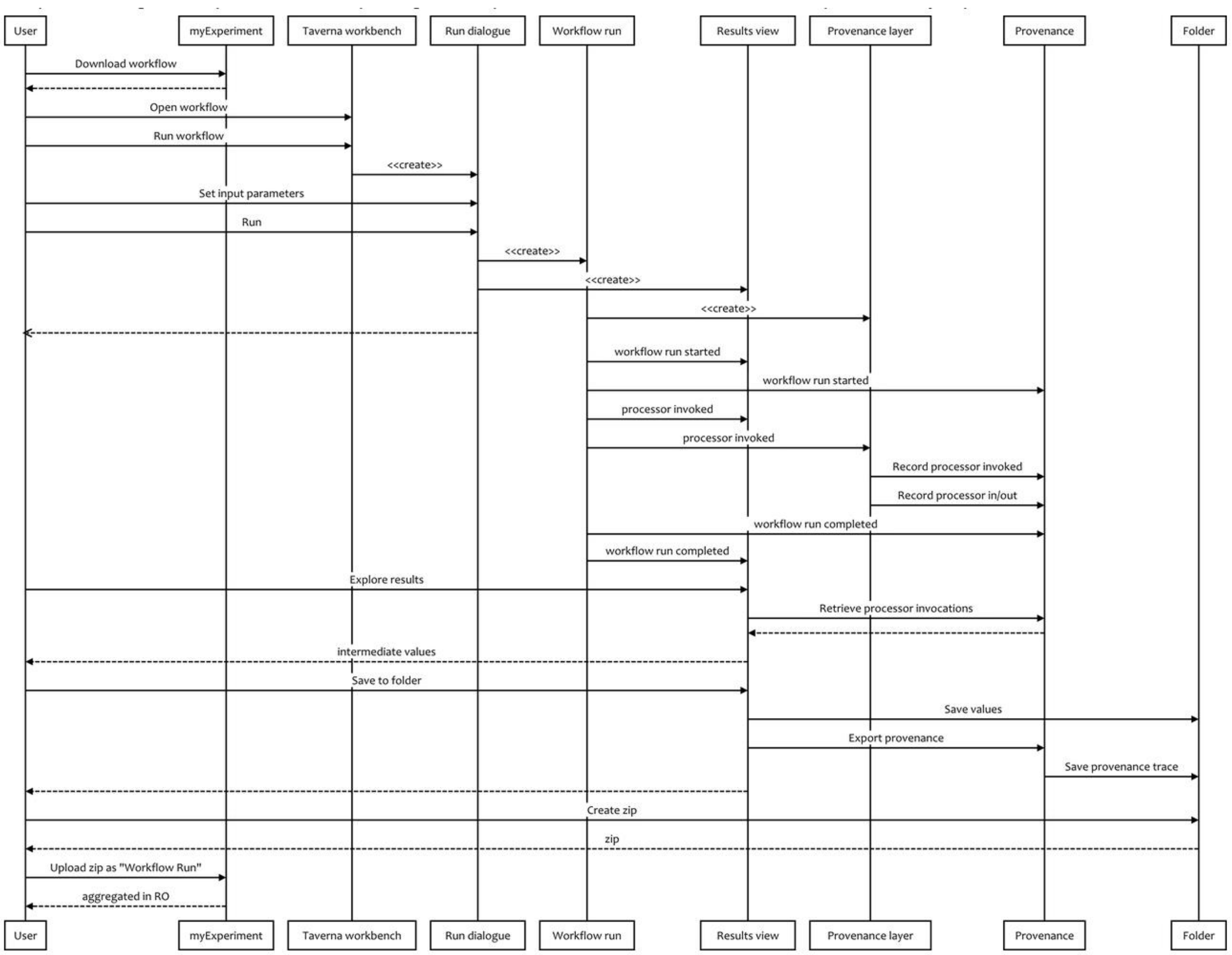


Figure 4 Taverna provenance export plugin sequence diagram

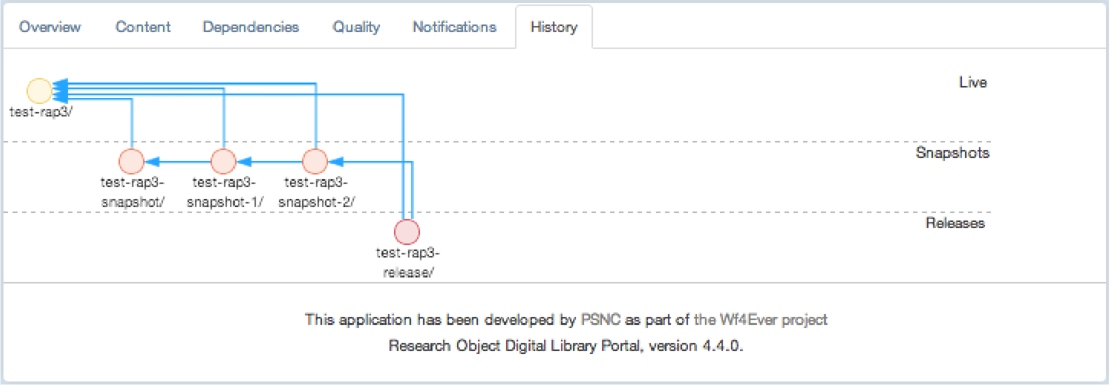
2.5 Provenance applications

Within the Wf4Ever project, provenance information has been used for diverse purposes, as described in the following.

RO Portal presentation of RO evolution

The RO Portal[18] displays RO evolution traces under the history tab of a research object page. This visualization can be seen in the Figure 5 and provides browsing capabilities throughout the different versions of a RO which are stored using the roevo ontology.

[18] http://sandbox.wf4ever-project.org/portal/home



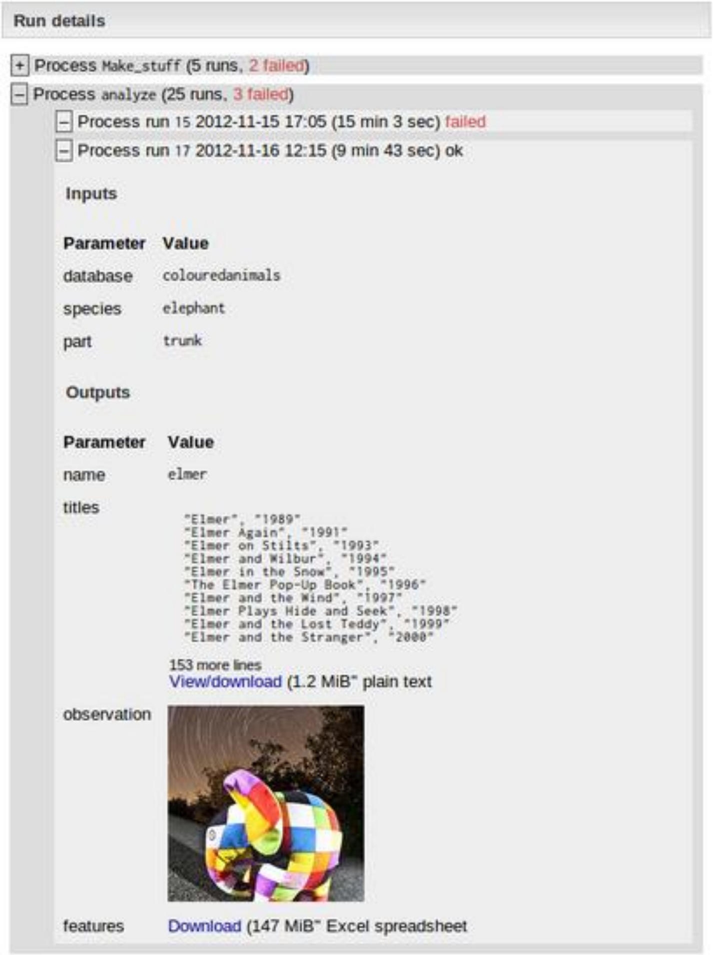
This application has been developed by PSNC as part of the Wf4Ever project
Research Object Digital Library Portal, version 4.4.0.

Figure 5 roevo visualization at RO Portal

myExperiment

The provenance information was originally included as a mockup of workflow run view in alpha-myExperiment[19] and is being upgraded to provide a high-level overview of wfprov on each RO resource page. The display shows any workflow runs in the research object, the inputs and outputs for each run, and the execution information as shown in Figure 6.

[19] http://alpha.myexperiment.org/



Run details

+ Process Make_stuff (5 runs, 2 failed)

– Process analyze (25 runs, 3 failed)

– Process run 15 2012-11-15 17:05 (15 min 3 sec) failed

– Process run 17 2012-11-16 12:15 (9 min 43 sec) ok

Inputs

Parameter	Value
database	colouredanimals
species	elephant
part	trunk

Outputs

Parameter	Value
name	elmer
titles	"Elmer", "1989" "Elmer Again", "1991" "Elmer on Stilts", "1993" "Elmer and Wilbur", "1994" "Elmer in the Snow", "1995" "The Elmer Pop-Up Book", "1996" "Elmer and the Wind", "1997" "Elmer Plays Hide and Seek", "1998" "Elmer and the Lost Teddy", "1999" "Elmer and the Stranger", "2000" 153 more lines View/download (1.2 MiB" plain text
observation	
features	Download (147 MiB" Excel spreadsheet

Figure 6 Provenance of workflow results mock-up included in alpha-myExperiment

Assessment of KEGG workflows

Provenance information has been used in the assessment of decay in KEGG workflows (KEGG: Kyoto Encyclopedia of Genes and Genomes[20]), specifically to locate the input data used to create additional RO annotations tested by the checklist evaluation for workflow decay. For this purpose, provenance information was extracted from a Taverna-generated provenance trace using a command line SPARQL query tool[21]. En example of a script of how to incorporate the provenance traces and convert

[20] http://www.genome.jp/kegg/

[21] https://github.com/gklyne/asqc

KEGG workflows to ROs in preparation to using the checklist service to perform decay detection can be also found at[22].

Discovering common workflow fragments on provenance

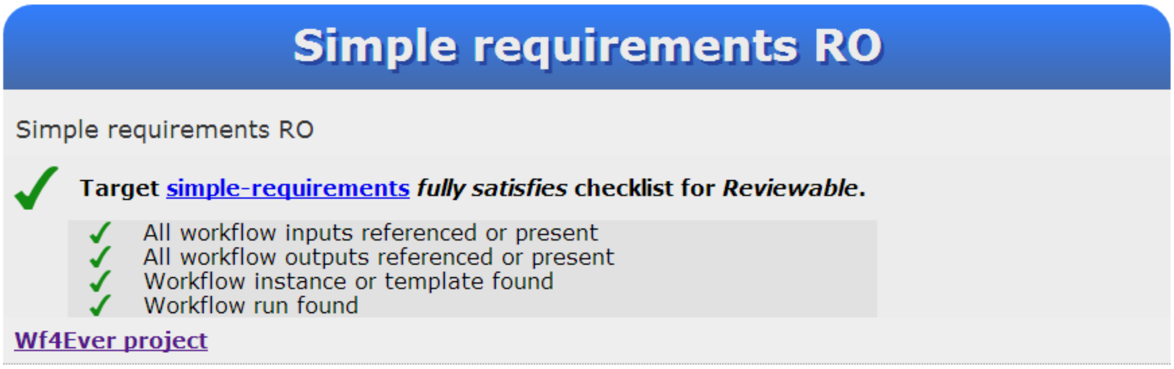
The provenance is used to automatically obtain abstractions from low-level provenance data by finding common workflow fragments on provenance of workflow execution and matching them to templates. This approach has been tested with a dataset of workflows published by Wings[23]. The obtained results showed that by using these kinds of abstractions we can highlight the most common patterns of methods used in the execution of a set of workflows (as ProvBench) relating different runs and workflow templates with each other [Gar'13]. The discovery of these common patterns also allows extending this application to provenance summarization locating the specific provenance information that is relevant to our final applications (e.g. trustworthiness) without losing its effectiveness [Alp'13].

Quality assessment

The evaluation of the completeness quality dimension of an RO uses the checklist evaluation service to query and test provenance values and resources with the main purpose of testing its availability and reusability. In such cases, the provenance is queried like any other RO annotation. The Figure 7 shows the presentation of this checklist evaluation including the confirmation that provenance exists in the RO ("Workflow run found").

[22] https://github.com/wf4ever/ro-catalogue/blob/master/v0.1/Kegg-workflow-evaluation/wf_conversion.sh#L142

[23] http://wings-workflows.org



Simple requirements RO

Simple requirements RO

✓ Target simple-requirements *fully satisfies* checklist for *Reviewable*.

✓ All workflow inputs referenced or present
✓ All workflow outputs referenced or present
✓ Workflow instance or template found
✓ Workflow run found

Wf4Ever project

Figure 7 Provenance verification for quality assessment

2.6 Provenance community engagement

As part of the provenance work we have participated in community activities for standardization, to promote its use, and the creation of a PROV-corpus.

Provenance standardization in W3C

The World Wide Web Consortium (W3C)[24] effort to create a standard for provenance was started at about the same time as the Wf4Ever project, and completed its work in May of 2013. A full list of the working group documents produced is summarized in [PROV-Overview][25]. During this period, several members of the Wf4Ever project have been active participants in the working group, including as contributors to the key standards documents published:

- PROV-O - the PROV ontology, an OWL2 ontology allowing the mapping of the PROV data model to RDF PROV-O[26].
- PROV-DM - the PROV data model for provenance PROV-DM[27].
- PROV-N - a notation for provenance aimed at human consumption PROV-N[28].

[24] http://www.w3.org/

[25] http://www.w3.org/TR/prov-overview/

[26] http://www.w3.org/TR/2013/REC-prov-o-20130430/

[27] http://www.w3.org/TR/2013/REC-prov-dm-20130430/

[28] http://www.w3.org/TR/2013/REC-prov-n-20130430/

Wf4Ever members have been also co-editing or contributing to the next supporting working group documents: PROV-PRIMER[29], PROV-AQ[30], PROV-DICTIONARY[31] and PROV-DC[32]. Furthermore, at the time of their publication, there were over 60 documented implementations ([PROV-implementations][33]) related to some aspects of PROV, most of which were producing or consuming elements of the provenance ontology (PROV-O), and some of which are already in deployed commercial products. We highlight that the Wf4Ever project made significant contribution to this early adoption of the new provenance standards.

ProvBench Challenge

The ProvBench[34] initiative objective was to bootstrap the publication of provenance information in an open and accessible fashion. The first ProvBench event was held at the 6th International Conference on Extending Database Technology (EDBT)[35], as part of the First International Workshop on Managing and Querying Provenance Data at Scale (BIGProv'13)[36]. This inaugural event received 8 submissions[37] from diverse interested research groups, including one from Wf4Ever which is explained in the following.

[29] http://www.w3.org/TR/2013/NOTE-prov-primer-20130430/

[30] http://www.w3.org/TR/2013/NOTE-prov-aq-20130430/

[31] http://www.w3.org/TR/2013/NOTE-prov-dictionary-20130430/

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

[33] http://www.w3.org/TR/prov-implementations/

[34] https://sites.google.com/site/provbench/

[35] http://edbticdt2013.disi.unige.it/

[36] https://sites.google.com/site/bigprov13/

[37] https://sites.google.com/site/provbench/provbench-at-bigprov-13/acceptedsubmissions

Wf4Ever provenance corpus

We have generated a provenance corpus[38], submitted to the first ProvBench event, whose dataset can be found at[39]. For this, we collected 120 provenance traces of workflows results from two well-known scientific community platforms, Taverna and Wings, are associated with 12 different applications domains. The provenance traces have been represented using the PROV-O ontology, with terms from other vocabularies such as RO model and OPMW[40] also used to associate the provenance traces with their corresponding workflow descriptions. Recall that OPMW is an ontology for describing workflows based on the Open Provenance Model which is used by many workflows allocated at WINGS repository and it has also been extended to be compliant with PROV-O standard.

The workflows associated with the Taverna platform have been generated by automatic capture of provenance by using the developed provenance plug-in[41] (see section 2.4 for further details), which provides wfprov output format. This plug-in was already implemented in its early stage at M20 and has been improved and tested for the generation of the ProvBench corpus. The whole Wf4Ever provenance corpus was assembled as a submission[42] to the first ProvBench event.

Among others, this dataset has been created for supporting the following scientific community interests and applications:

- discovery of common “motifs” and annotation of workflows subgraphs by identifying the most frequent in-use patterns. This work can be consulted in the [D2.2v2] and [Gar’13],
- search for discovered similar scientific workflows based on pattern similarities[43][44],

[38] http://www.wf4ever-project.org/wiki/display/docs/Provenance+corpus

[39] https://github.com/wf4ever/provenance-corpus

[40] http://www.opmw.org/

[41] http://wf4ever.github.com/taverna-prov/

[42] http://dx.doi.org/10.1145/2457317.2457376

[43] http://sandbox.wf4ever-project.org/wfabstraction/rest/search

[44] http://www.wf4ever-project.org/wiki/display/docs/44c.+Discover+workflow+pattern+similarities+and+linking

- identification of patterns of use for obtaining dependencies recognition and provide recommendation[45,46],

and allows answering questions such as:

- what are the workflow runs available, and what is their start and end time?,
- what are the workflow runs associated with a given workflow template, and how many of them failed?,
- what are the workflow runs of a given workflow template, and what are the inputs they used and the outputs they generated?,
- how many process runs are associated with a given workflow run, what is the start and end time of each one, and what are the inputs they used and the outputs they generated?,
- who executed a given workflow run?, and
- what are the services invoked as a result of a given workflow run?[47,48]

which have been assembled as a set of queries. Also, part of this corpus has been used subsequently in our analysis of KEGG workflow decays (see section 4 of this document).

[45] http://sandbox.wf4ever-project.org/wfabstraction/rest/recommend

[46] http://www.wf4ever-project.org/wiki/display/docs/Workflow+Indexing+API

[47] http://www.wf4ever-project.org/wiki/display/docs/Taverna+provenance+query+examples

[48] http://www.wf4ever-project.org/wiki/display/docs/Wings+provenance+query+examples

3 Quality Evaluation in Wf4Ever

This section introduces general framework designed and implemented in Wf4Ever which has provided the needed information for the establishment of a quantitative measure of the different dimensions (completeness, stability, and reliability) identified as important for the definition of an overall quality RO criteria.

Evaluating the health of the workflow contained in a specific research object requires transforming the additional information encapsulated by the research object, provided by the different implemented/used models within the Wf4Ever project, into a quantifiable value and providing the scientists with the necessary means to interpret such values.

We have established a clear separation between the different types of knowledge involved in order to evaluate the quality of a scientific workflow, as illustrated in Figure 8 which depicts a pyramid structured in three main layers, where the completeness, stability and reliability dimensions which helps to define the overall quality score of a research object is obtained through the evaluation of the information contained in the underlying levels. It is important to clarify that the overall quality score includes both, integrity and authenticity terms, defining authenticity as the evaluation of whether a RO is exactly what it is claimed to be, and by integrity referring to the verification that the transformations to which the RO has been subjected have not introduced any undisclosed distortion or loss in the resulting RO.

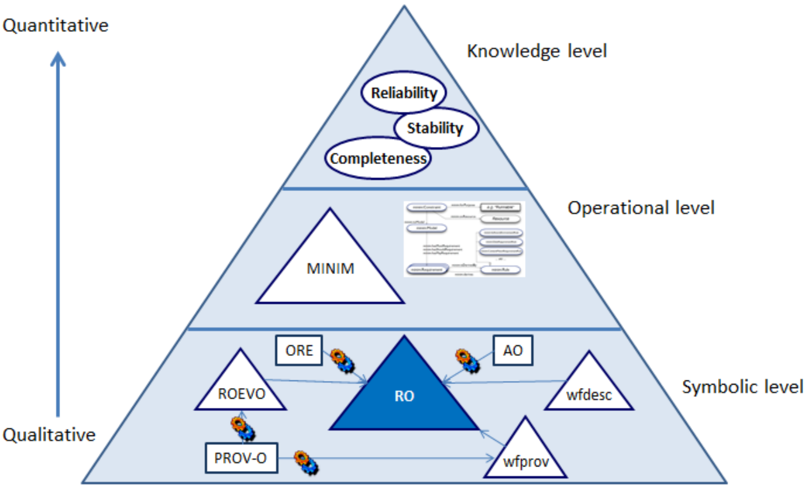


Figure 8 Quality ontology pyramid

The bottom layer of the pyramid corresponds to the RO model, described in the RO model specification RO model[49]. Based on the metadata about the research object, its constituent parts and annotations, a new layer is included that contains knowledge about the minimum requirements that must be observed by the research object in order to remain fit for a particular goal and about the predicates in charge of evaluating such requirements. This layer, which we call operational in the sense of the methods through which the requirements are evaluated, is modelled as checklists (see [zhao'12]) following the Minim OWL ontology. The evaluation of the checklists results into a number of Boolean values indicating whether the specified requirements are fulfilled or not.

Finally, the top of the pyramid for assessing the reliability of scientific workflows contains quantitative values about reliability, stability, and completeness based on information derived from the outcomes of the checklist evaluation in the previous layer. These metrics are calculated following the algorithms and methods described in sections 4 and 5 and their values are stored as additional metadata in the research object, providing a compact type of quantitative information about the reliability of specific workflows. Based on these metrics plus the tooling necessary to interpret them scientists are enabled to make an informed decision about workflow reuse at the knowledge level, i.e. focusing on their domain expertise and not requiring a deep inspection of the information in the research object.

Advances accomplished since M20 include the implementation of the above introduced quality framework that unifies previous work on completeness and stability, and also includes the new dimension so called reliability. Also, the individual components have been improved by incorporating new functionalities described in the next sections (e.g. new rules and tests), and new ways to present quality information about a research object, such as the new RO-Monitoring tool and the checklist display.

[49] http://wf4ever.github.io/ro/

4 Completeness

4.1 Introduction

In Wf4Ever the completeness evaluation has been accomplished by evaluating checklists to verify the existence of specific resources within the RO. Checklists are a widely used tool for controlling and managing quality assurance processes [Hales'06], and they have appeared in data quality assurance initiatives such as MIBBI [MIBBI], which deals with coherent minimum reporting guidelines for scientific investigations. A checklist provides a measure of fitness for purpose rather than some overall measure of quality. We see this kind of fitness for purpose assessment as being of more practical use than a generic quality assessment, and indeed as the ultimate goal of any quality evaluation exercise. The suitability of a research object for different purposes may be evaluated using different checklists: there is no single set of criteria that meaningfully applies in all situations, which leads to a need to describe different quality requirements for different purposes. For this purpose, we have defined a Minim model using OWL[50].

Some of the ideas for minimum information models developed at [MIBBI] initiative have been adopted and generalized in our Minim model, which is an adaptation of the MIM model [MIM], to deal with a range of research object (RO) related quality concerns. Conforming to a minimum information model gives rise to a notion of completeness, i.e. that all information required for some purpose is present and available. In our work, a checklist is a set of requirements on a research object that can be used to determine whether or not all information required for some purpose is present, and also that the provided information meets some additional criteria.

The Minim model was introduced in D4.2v1 [D4.2v1], reflecting its development as of August 2012, but its design and application has substantially progressed. In applying the checklist evaluation capability to myExperiment RO quality display, and other quality evaluations, we have since implemented or updated the following parts:

[50] http://purl.org/minim/

- refactored the Minim model, and extended its range of capabilities to meet additional requirements,
- updated the checklist evaluation code to use a SPARQL 1.1 library in place of SPARQL 1.0, significantly enhancing the expressive capability of the Minim model,
- developed a "traffic light" display of checklist results (for myExperiment integration and other uses),
- developed a REST web service for RO checklist evaluation, and deployed this in the Wf4Ever sandbox,
- created new checklist designs using the Minim model for myExperiment RO quality display, based on scenarios articulated by Wf4Ever project user partners, and incorporated checklist evaluation into work on RO stability and reliability evaluation (described below).
- We have also started work to evaluate the capabilities of the Minim model applied to a range of quality evaluation scenarios.

In the next subsections we describe the Minim data model used to define checklists, the Minim results data model used to express the result of a checklist evaluation, additional services created to support presentation of evaluation results to users of research objects, the checklist evaluation software structure and its integration with other Wf4Ever project elements, and some applications that have been created using the checklist evaluation capabilities.

4.2 Ontological models

The evaluation of completeness is based on a set of requirements defined as a checklist, which is described by a Minim model. The results of a checklist evaluation are presented using the Minim results model. We describe these models in the following sections.

4.3 Minim model for defining checklists

This model has been significantly refactored and enhanced since M20. The enhancements provide a cleaner structure to the overall model, greater expressive capability (including value cardinality tests similar to those supported by MIM), and clear identification of extension points at which new capabilities can be added to the model. The refactoring is done so that old-style Minim definitions do not conflict with new style definitions, and both may be (and are) supported in a single implementation. The Minim ontology[51], its specification[52], and its OWLDoc documentation[53] are maintained in a GitHub project[54].

The main elements of the Minim model are:

- **Checklists**: different models may be provided for different purposes; e.g. the requirements for the purpose of reviewing an experiment are different from those for a purpose of workflow runnability. A minim Checklist associates a minim Model with a description of the quality evaluation purpose it is intended to serve, as shown in the Figure 9.

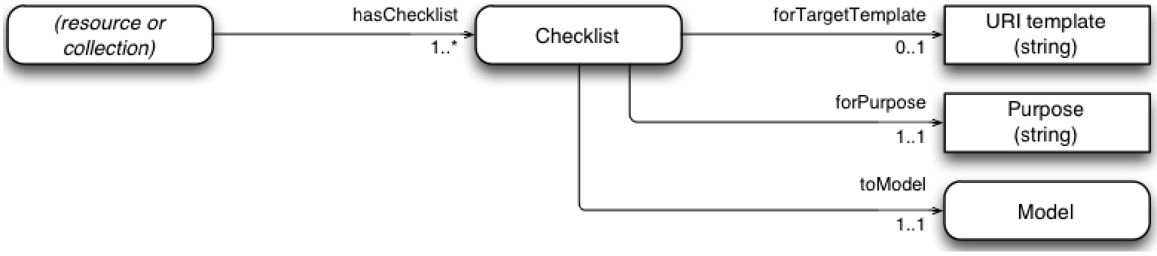


Figure 9 Checklist Model Diagram

- **Model**: a Minim Model defines a list of requirements to be satisfied, which can be of three different types: mandatory (minim:hasMustRequirement), desirable (minim:hasShouldRequirement), or optional (minim:hasMayRequirement) (see Figure 10 where the grey coloured boxes represent abstract classes).

- **Requirement**: denotes some specific requirement to be satisfied by a research object, such as the presence of certain information about an experiment. For

[51] http://purl.org/minim/minim

[52] https://github.com/wf4ever/ro-manager/blob/develop/Minim/minim-revised.md

[53] http://purl.org/minim/owldoc

[54] https://github.com/wf4ever/ro-manager/tree/master/Minim

example, we may wish to test that a suitable reference to input data is provided by an RO, and also that the data is live (accessible), or that its contents match a given value (integrity).

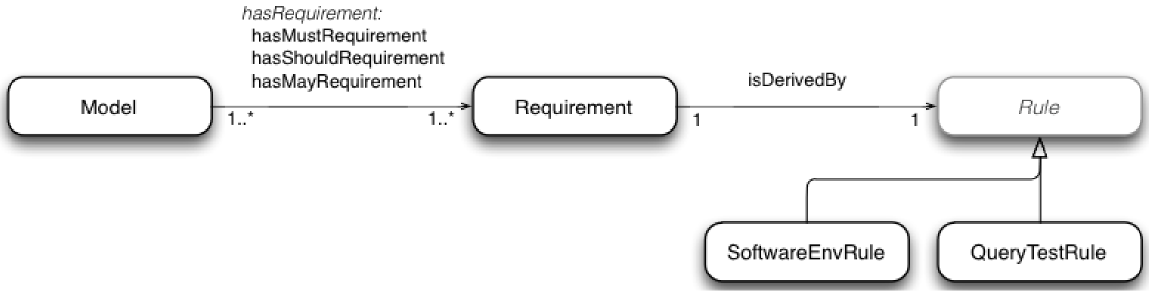


Figure 10 Minim Model Diagram

- **Rule**: a rule is associated with each requirement, and describes how the requirement has to be tested. A small number of different rule types are currently supported by the checklist service, including tests of the local computing environment for presence of particular software, and tests that query a research object and perform tests on the results obtained. A rule determines whether a research object satisfies some technical requirement (e.g. that some specific resources are available, or accessible), which is interpreted as an indicator of some end-user goal.

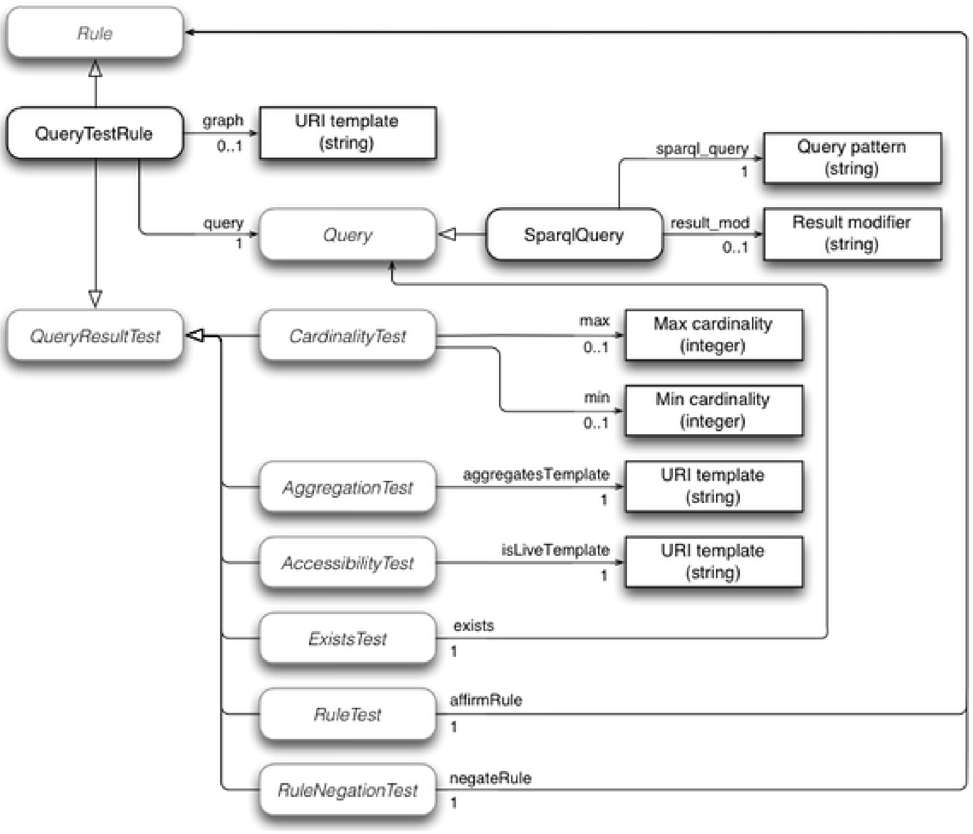


Figure 11 Rule Model Diagram

The Minim model uses the “abstract” classes that may be further subclassed to introduce new evaluation capabilities:

- **Rule**: new rule types can be introduced to perform tests for new kinds of requirement that cannot be handled within existing rule structures. For example, if a workflow has a dependency on a particular kind of computing hardware environment, such as a particular model of quantum computing coprocessor, then new rule types might be introduced to cover tests for such things.
- **Query**: this is an extension point within QueryTestRule, which allows query types other than SPARQL to be introduced. For example, a SPIN query processor, or an OWL expression used to find matching instances in the RO metadata might be introduced as different query types. The model assumes that query results are returned as lists of variable-binding sets (e.g. lists of dictionaries or hashes).
- **QueryTestRule**: is a particular Rule which is used for different tests. This type of is based initially on a query that is performed over the RO metadata the results of which may be tested in different ways.

- **QueryResultTest**: this is another extension point within QueryTestRule, which allows different kinds of test to be applied to the result of a query against the RO metadata. For example, checking that a particular URI in the metadata is the access point for an implementation of a specific web service might be added as a new query result test.

The outcome of a checklist evaluation is returned as an RDF graph, using terms defined by the Minim results model as described in the Figure 12. The results returned graph also includes a copy of the Minim description used to define the assessment allowing the creation of a fully meaningful rendering of the result. The design is intended to allow multiple checklist results to be merged into a common RDF graph without losing information about which result applies to which combination of checklist, purpose and target resource.

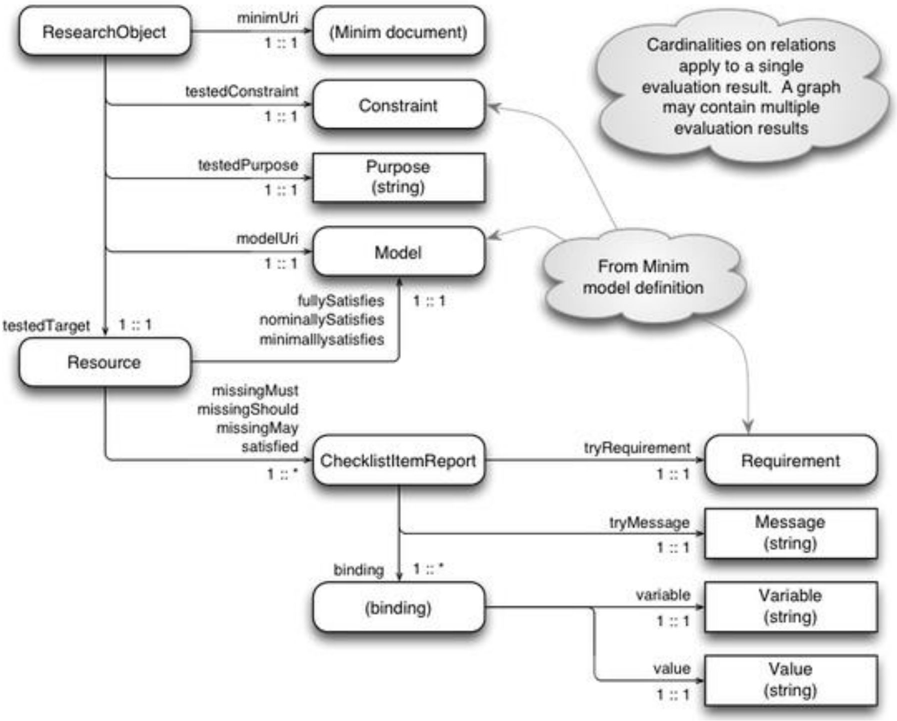


Figure 12 Minim Results Model

The main result of a checklist evaluation is an indication of whether a target resource **fullySatisfies**, **nominallySatisfies**, or **minimallySatifies** the associated checklist model, evaluated in the context of a particular RO. A fullySatisfies relation means that all requirements of the model are satisfied by the target resource; nominallySatisfies means that all MUST and SHOULD requirements are satisfied, indicating that the target resource is complete with desirable features for the specified purpose; and minimallySatifies means that all MUST requirements are satisfied indicating that the target resource is complete for the specified purpose, but some desirable features or characteristics may be missing.

The model also describes a breakdown of the checklist evaluation result by using **missingMust**, **missingShould**, **missingMay** and/or satisfied properties, which indicate the evaluation result for each individual checklist item as a relationship between the target resource and the corresponding checklist requirement. Also the explanations of this outcome are stored at the Message class providing more detailed information about the reason for success or failure of the test. Figure 13 shows an example of a Minim requirement that tests for presence of a synonym in chembox data,

```
:Synonym a minim:Requirement ;
  minim:isDerivedBy
    [ a minim:QueryTestRule ;
      minim:query
        [ a minim:SparqlQuery ;
          minim:sparql_query "?targetres chembox:OtherNames ?value" ;
        ] ;
      minim:min 1 ;
      minim:showpass "Synonym is present" ;
      minim:showfail "No synomym is present" ;
    ] .
```

Figure 13 Minim requirement for presence testing

and this may returns a result as shown in Figure 14 for the target resource N-Methylformamide[55], for which no synonym exists in the data provided by the RO. This result describes that the RO satisfies minimally and nominally the requirements of the Minim Model, and that there are some MAY requirements (defined by the hasMayRequirement property in the Minim model) which are not satisfied, as explained by the missingMay statement.

[55] http://purl.org/net/chembox/N-Methylformamide

```
<http://purl.org/net/chembox/N-Methylformamide>
    minim:minimallySatisfies :minim_model ;
    minim:nominallySatisfies :minim_model ;
    minim:missingMay
        [ minim:tryMessage "No synomym is present" ;
            minim:tryRequirement :Synonym ;
            result:binding
                [ result:variable "targetres" ;
                  result:value "http://purl.org/net/chembox/N-Methylformamide" ],
                [ result:variable "query" ;
                  result:value "?targetres chembox:OtherNames ?value" ],
                [ result:variable "min" ;      result:value 1 ],
                [ result:variable "_count";     result:value 0  ]
        ] .
```

Figure 14 Results represented with the Minim Results Model

4.4 Implementation and integration

The implementation and integration of completeness evaluation in the context of Wf4ever has the main goal of interacting with the data available through RODL, providing APIs that offer users and client applications access to the checklist evaluation service. The checklist evaluation service is implemented as part of the codebase for RO Manager [D2.2v2]. It is implemented in Python, is available as an installable package at[56], and its source code can be found at[57].

The checklist evaluation has been implemented as a command line tool (which can be called by the command “ro evaluate checklist”), and as a web service[58,59]. The command line version of checklist evaluation has been used mainly for development purposes, and in the next section we focus our discussion on the web service implementation.

[56] https://pypi.python.org/pypi/ro-manager

[57] https://github.com/wf4ever/ro-manager

[58] http://sandbox.wf4ever-project.org/roevaluate/

[59] http://purl.org/minim/checklist-service

4.5 Service interface and interactions

Overall, the Wf4Ever architecture [D1.3v2] is designed around use of linked data and REST web services, with interaction between components being handled by HTTP requests. A checklist evaluation is invoked by a simple HTTP GET operation, in which the RO, Minim resource URI, target resource URI and purpose are encoded within the request URI. The evaluation result is the result of the GET operation. A complete description of the API can be found at the Wf4Ever project wiki page[60].

The checklist service in turn interacts with the RO through RODL, mainly to retrieve the RO annotations. Some checklist items, such as those that check for liveness of workflow dependencies, may cause further requests to arbitrary web resources named in the RO metadata. The Figure 15 shows the interaction between RODL, external services, and the checklist service during a typical checklist call for obtaining the evaluation results for the completeness of a RO.

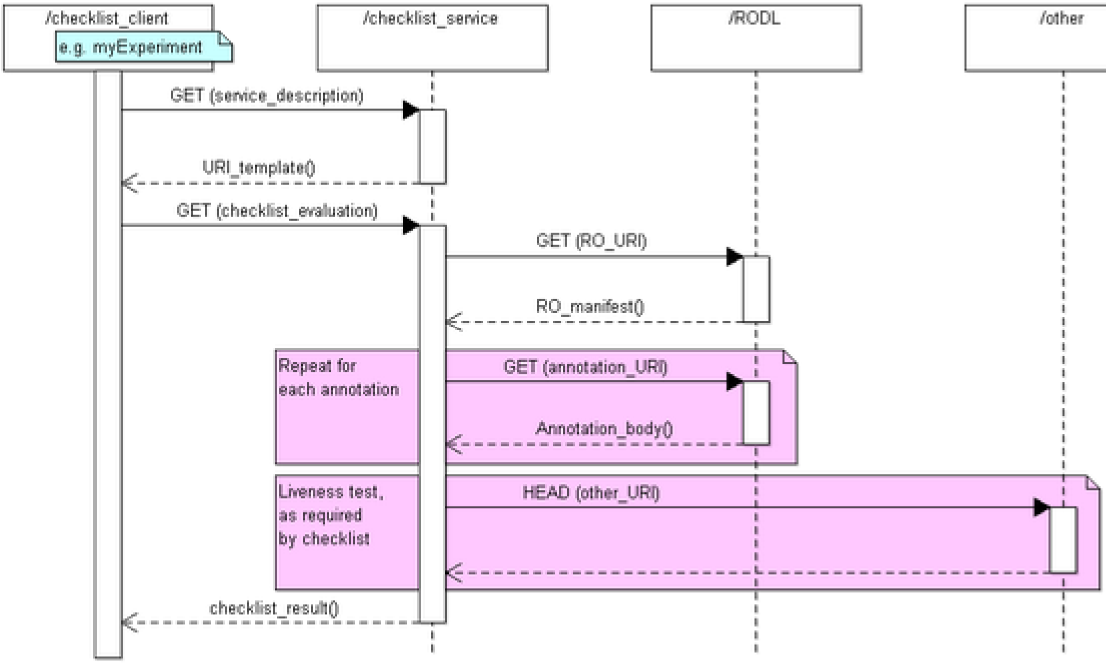


Figure 15 Sequence diagram of the checklist service.

[60] http://www.wf4ever-project.org/wiki/display/docs/RO+checklist+evaluation+API

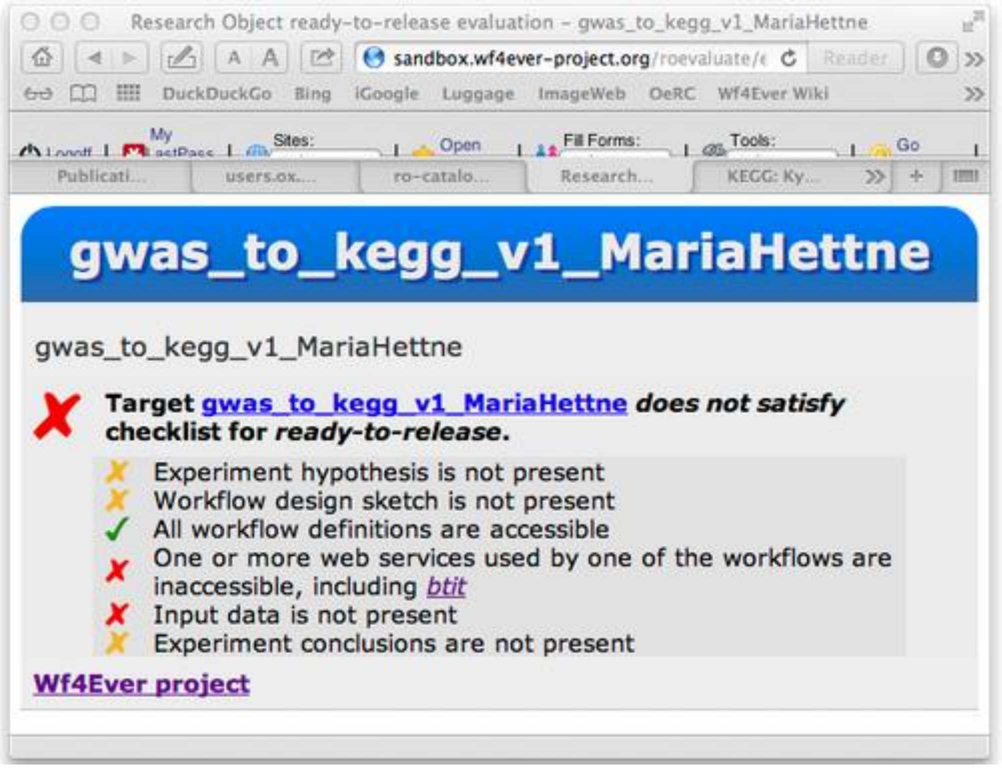
4.6 Completeness applications

In this section we briefly describe some applications where the checklist service has been used within the Wf4Ever project.

Detection of workflow decay

The main purpose of this application is to anticipate and detect the potential causes of workflow decay. During the execution of the project, the Kyoto Encyclopedia of Genes and Genomes[61] announced (2012) that they were introducing a REST interface for their discovery service, and discontinuing the older web Services based interface. Due to there being a number of workflows in myExperiment that use the older KEGG services we decided to use this service update to test our decay detection capabilities. Before the old service was shut down, the KEGG-using workflows were surveyed and a considerable number were found to still be executable. Our hypothesis was that after the KEGG web services were shut down at the end of 2012, our checklist service should successfully detect and report the workflow decay. The outcome of this study was a set of results indicating showing decay of workflows due to withdrawal of the KEGG web. The presentation of this for a specific workflow can be seen in Figure 16, as the 4th checklist item.

[61] http://www.genome.jp/kegg/


Research Object ready-to-release evaluation – gwas_to_kegg_v1_MariaHettne
sandbox.wf4ever-project.org/roevaluate/
Reader
DuckDuckGo Bing iGoogle Luggage ImageWeb OeRC Wf4Ever Wiki
Sites: Open Fill Forms: Tools: Go
Publicati... users.ox... ro-catalo... Research... KEGG: Ky...

gwas_to_kegg_v1_MariaHettne

gwas_to_kegg_v1_MariaHettne

✗ **Target gwas_to_kegg_v1_MariaHettne *does not satisfy* checklist for *ready-to-release*.**

- ✗ Experiment hypothesis is not present
- ✗ Workflow design sketch is not present
- ✓ All workflow definitions are accessible
- ✗ One or more web services used by one of the workflows are inaccessible, including *btit*
- ✗ Input data is not present
- ✗ Experiment conclusions are not present

Wf4Ever project

Figure 16 Checklist service visualization of KEGG service.

Completeness assessment for workflow decay prevention

This application focuses on the creation of ROs containing workflow results with additional information to support workflow re-use and repair. Our goal was to create a checklist that can be incorporated into the conduct of workflow-based experiments, to encourage experimenters to provide useful information, and to automate some mechanical aspects of the review process that otherwise have to be done manually. This has been based on the current implementation of the completeness dimension (i.e. using models described in this section) and on earlier work where we analysed the main causes of workflow decay for a set of representative workflows selected from myExperiment [zhao'12]. This work has led to the definition of a set of checklists such as checklist-runnable.rdf[62] and workflow-experiment-checklist.rdf[63], which provide similar assessments to that shown in Figure 16.

[62] https://github.com/wf4ever/ro-catalogue/blob/master/v0.1/golden-exemplar-gk/checklist-runnable.rdf

[63] https://github.com/wf4ever/ro-catalogue/blob/master/v0.1/Y2Demo-test/workflow-experiment-checklist.rdf

Completeness assessment of resource descriptions: chembox

This application evaluates the completeness of resource descriptions in external linked data. Specifically, we have used the checklist evaluation service to assess the completeness of chemical descriptions in DBPedia, which in turn were extracted from Wikipedia "Chembox" templates. For this purpose a new checklist[64] was created and used, jointly with a script[65] for automatically perform the evaluations. The results of this study are available at[66].

Basis for stability assessment

The preceding discussions have considered use of the checklist service for static analysis of research objects. It has also been used for assessment of dynamic properties, articulated as stability and reliability. How the completeness evaluation is used, and its interpretation in that context is described in the next section.

[64] https://github.com/wf4ever/ro-catalogue/blob/master/v0.1/minim-evaluation/chembox-minim-samples.ttl

[65] https://github.com/wf4ever/ro-catalogue/blob/master/v0.1/minim-evaluation/chembox_evaluate.sh

[66] https://github.com/wf4ever/ro-catalogue/blob/master/v0.1/minim-evaluation/chembox.ttl

5 Stability/Reliability and Quality Scores

5.1 Introduction

In Wf4Ever the stability and the reliability assessments have been accomplished by implementing a REST service which uses the information provided by the completeness assessment (explained in section 5.2) during some previous period of the RO lifetime. This dynamic analysis has been adopted due to workflows (which are the executable resources of a RO) might break unexpectedly at any time, and therefore taking into account only the static perspective alone (i.e. the current RO state), which would provide an incomplete view of the usability of an RO over time. In [Zhao'12] we saw that a common cause of decay is the volatility of some of third party resources, which cannot be controlled or predicted locally and are not easy to repair.

The stability and reliability metrics aim to keep track and measure the changes of the completeness assessment of a RO throughout time. Both try to establish a criteria for allowing the verification that the transformations to which the RO has been subjected have not introduced any undisclosed distortion or loss which could damage the correct behaviour of the RO (e.g. for the purpose of running it). Thus, stability and reliability use the completeness assessment as a basis, as it provides the definition of current fitness-for-purpose of an RO, defined by a Minim Model and the set of requirements that it incorporates.

While the completeness evaluation (introduced and explained in the previous section 4) provides detection of specific factors causing decay in an RO, the stability and reliability evaluations consider an additional dimension, namely time. The inclusion of this new factor leads to a new model which reflects how much the user should trust a research object for purposes of re-use. The stability measures the ability of a RO to maintain its status during its lifetime; extending this to incorporate the completeness assessment allows computation of a RO's reliability (including the workflow that contains).

By reliability we measure the confidence that the scientist can have on a particular workflow for preserving its capabilities to be executed and produce the expected results. A reliable workflow is expected not only to be free of decay at the moment of being inspected but also in general throughout its life span. Consequently, in order to establish the reliability of a workflow it becomes necessary to assess to what extent it is complete with respect to a number of requirements and how stable it has been with respect to such requirements historically. Figure 17 zooms in the top of the pyramid at Figure 8, schematically depicting the reliability concept as a compound on top of completeness and stability along time.

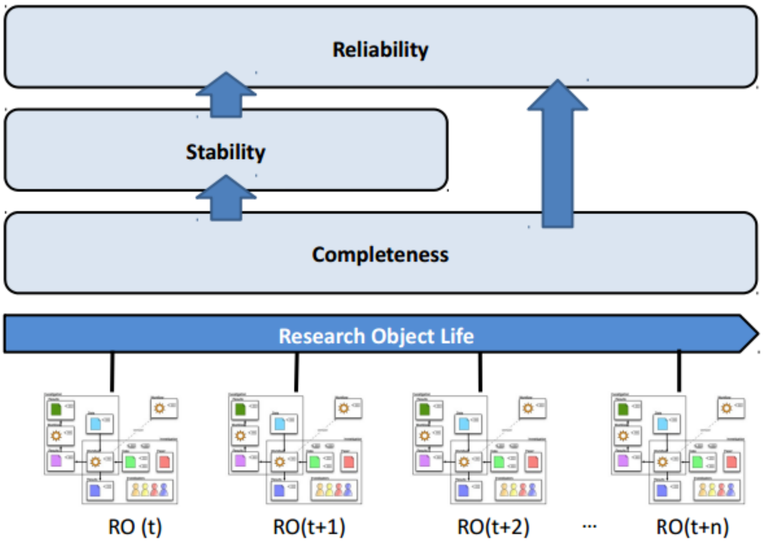


Figure 17 Layered Components of Reliability Measurement

In the next section we explain how the scores for these three dimensions are calculated and how to interpret them. Later on in this section we also show its implementation and the visualization developed in Wf4Ever for showing the obtained results.

5.2 Completeness Score

The completeness dimension evaluates the extent to which a RO satisfies a number of requirements specified in the form of a checklist following the Minim OWL ontology. As defined in the Section 4.3 such requirements can use any of three types (must, should, and may) but for easier end users' interpretability we have defined the completeness score using the two strongest constrained types: must (compulsory) and should (recommended)[67]. In order to be runnable and reproducible all the must requirements associated to a workflow need to be satisfied while should requirements are ones whose non-satisfaction might suggest problems in some circumstances. An example of the former is that all the web services invoked by the workflow be available and accessible (two of the main causes of workflow decay), while the presence of user annotations describing the experiment would illustrate the second case. Since must requirements have a strong impact in the quality we have defined two thresholds: a) a lower bound β_l which establishes the maximum value that the completeness score can have in case it does not satisfy all must requirements, and b) an upper bound β_u which establishes the maximum value that the completeness score can have given that it satisfies all should and must requirements. Both β_l and β_u are parameterizable and can be configured on a case by case basis.

Therefore if at least a must requirement fails the completeness score is in the lower band [0-β_l] and otherwise in the upper band [β_l - β_u] (Note that the parameters β_l and β_u need to be predefined) . Once identified the band, we define a normalized value of the completeness score for those bands (lower band and upper band) as:

$$completeness\_score(RO,t) = f(RO_{(t)}, requirements, type) = \alpha\frac{nSReq(RO_{(t)}, must)}{nReq(must)} + (1-\alpha)\frac{nSReq(RO_{(t)}, should)}{nReq(should)} \in [0,1],$$

Formula 1 Completeness score

where t is the point in time considered, RO the research object that contains the workflow being evaluated, requirements the specific set of requirements defined within the RO for a specific purpose, type ∊ {must, should} the category of the requirement, α ∊ [0,1] is a control value to weight the different types of requirements, nSReq the number of satisfied requirements, and nReq the total number of requirements for the specified type.

[67] Note: At this point there is not any study which reveals in a quantitative way how to use the different types so we have decided to limit the use to two for making easier the validation process.

5.3 Stability Score

The stability measures the ability of a workflow to preserve its properties through time. The evaluation of this dimension provides the needed information to scientists and end users in order to know how stable the workflow has been in the past in terms of completeness fluctuation and therefore to gain some insight as to how predictable its behaviour can be in the near future. We define the stability score as follows:

$$\boxed{stability\_score(RO,t) = 1 - std(completeness\_score(RO,\Delta t)) \in [0.5,1],}$$

Formula 2 Stability score

Where the completeness score is the measurement of completeness in time t and Δt is the period of time before t used for evaluation of the standard deviation.

The stability score has the following properties:

- It reaches its minimum value when there are severe changes over the resources of a workflow for the period of time Δt, meaning that the completeness score is continuously switching from its minimum value of zero (bad completeness) to its maximum of one (good completeness). This minimum value is therefore associated to unstable workflows.
- It has its maximum value when there are not any changes over a period of time Δt, meaning that the completeness score does not change over that time period. This maximum value is therefore associated to stable workflows.
- Its convergence means that the future behaviour of the workflow can be predictable and therefore potentially reusable by interested scientists.

5.4 Reliability Score

The reliability of a workflow measures its ability for converging towards a scenario free of decay, i.e. complete and stable through time. Therefore, we combine both measures completeness and stability in order to provide some insight into the behaviour of the workflow and its expected reliability in the future. We define the reliability score as:

$$\boxed{reliability\_score(RO,t) = completeness\_score(RO,t) * stability\_score(RO,t) \in [0,1],}$$

Formula 3 Reliability score

where RO is the research object, and t the current time under study. The reliability score has the following properties:

- It has a minimum value of 0 when the completeness score is also minimum.
- It has a maximum value of 1 when the completeness score is maximum and the RO has been stable during the period of time Δt.
- A high value of the measure is desirable, meaning that the completeness is high and also that it is stable and hence predictable.

5.5 Implementation and integration

The implementation and integration of reliability and stability metrics in the context of Wf4ever has the main goal of interacting with the data available in the platform through RODL and providing useful APIs that offer to end users and client applications accessibility to these services and the quality evaluations for a period of time. Due to stability is used by the reliability score the provided service is unique (see section 5.6 and Figure 20 for more details).

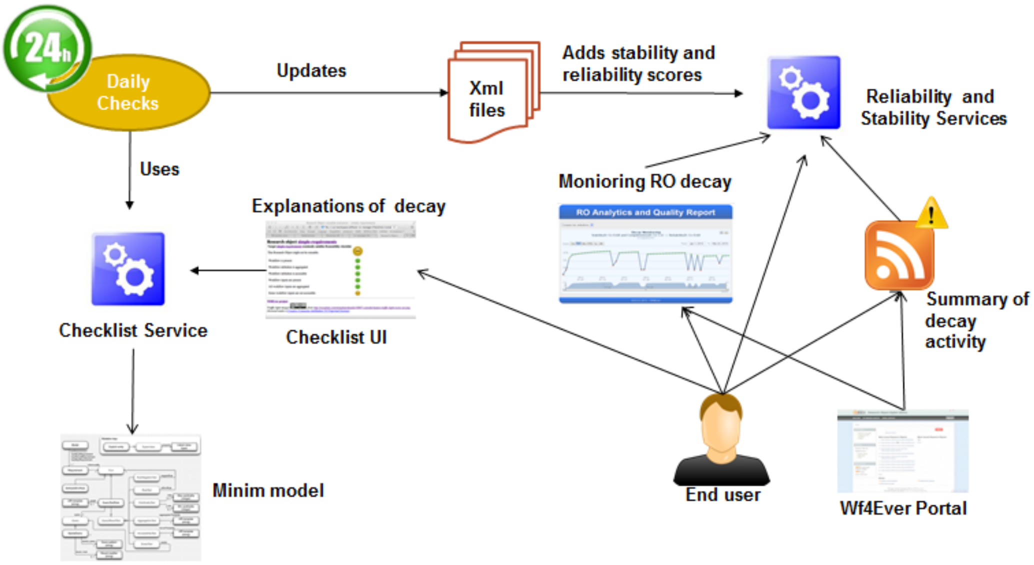


Figure 18 Wf4Ever quality assessment components interactions

The overall components interactions related to the I&A work are shown in Figure 18. Firstly the checklist service is called periodically (e.g. daily) for obtaining historical completeness scores (see Formula 1) which are then used for calculating the stability trace of the research object as indicated by Formula 2. This process identifies the existing RO's URIs by performing a SPARQL query on the RODL endpoint and then calls the checklist evaluation service returning the results in JSON format.

Afterwards these results are processed in order to calculate the completeness score and to summarize the satisfaction of the requirements, "pass" or "not pass", specified by the Minim model (see Section 4.3). The Minim description (which is available as a web resource) and the purpose label are also needed parameters, which have to be predefined, for obtaining the reliability score. All this information is stored on an XML file for each RO as shown in the Figure 19 which includes the information for different days and it is the input for the calculation of the stability and reliability scores (one value per day as shown in the Figure 22).

We have also implemented another alternative way for getting the quality results via notifications. These notifications provide a short summary of completeness, stability and reliability for all those days whenever the completeness had changed during a specific requested period of time. The notification service[68] follows the ATOM standard format and users can subscribe to it in order to get alerted when something happens with a research object of their interest (e.g. a user may want to obtain notifications for all his ROs so whether decay is detected on one of them it will be notified allowing him to fix it quickly).

[68] http://sandbox.wf4ever-project.org/decayMonitoring/rest/notifications

```
<trace>
<rouri> http://sandbox.wf4ever-project.org/rodl/ROs/myExpRO_1167/</rouri>
<evaluations>
<eval evalresultclass="must">
<date>2013,5,9,15,17</date>
<checklistitems>
       <checklistitem itemlevel="must" itemsatisfied="true">Third party resources accessible</checklistitem>
       <checklistitem itemlevel="must" itemsatisfied="true">Third party resources have not changed</checklistitem>
       <checklistitem itemlevel="should" itemsatisfied="true">Execution environment available</checklistitem>
       <checklistitem itemlevel="should" itemsatisfied="false">Workflow description not available</checklistitem>
</checklistitems>
</eval>
</evaluations>
</trace>
```

Figure 19 Checklist results for a research object presented in XML format.

5.6 Service interface and interactions

Following the Wf4Ever architecture [D1.4v1] [D1.4v2] the stability/reliability assessments and the notification service are designed using REST web services and linked data. The interaction between stability/reliability and the checklist evaluation is done by a simple HTTP GET operation. The stability and reliability service is invoked by a simple HTTP GET operation, in which the RO URI is encoded within the request URI[69]. The evaluation result is the output of the GET operation. A complete description of the API can be found at the Wf4Ever project wiki page[70].

[69] http://sandbox.wf4ever-project.org/decayMonitoring/rest/getReliability

[70] http://www.wf4ever-project.org/wiki/display/docs/Reliability+Evaluation+API

Regarding the notification service is also done via a HTTP GET operation passing the RO URI, and the starting (“from”) and ending (“to”) dates of the period under study in ISO 8601 time format[71].

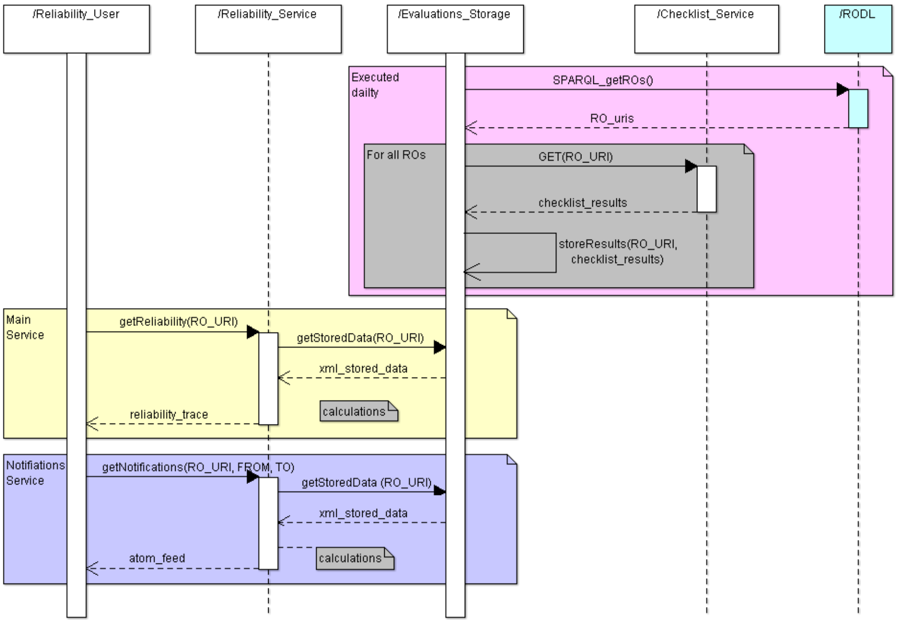


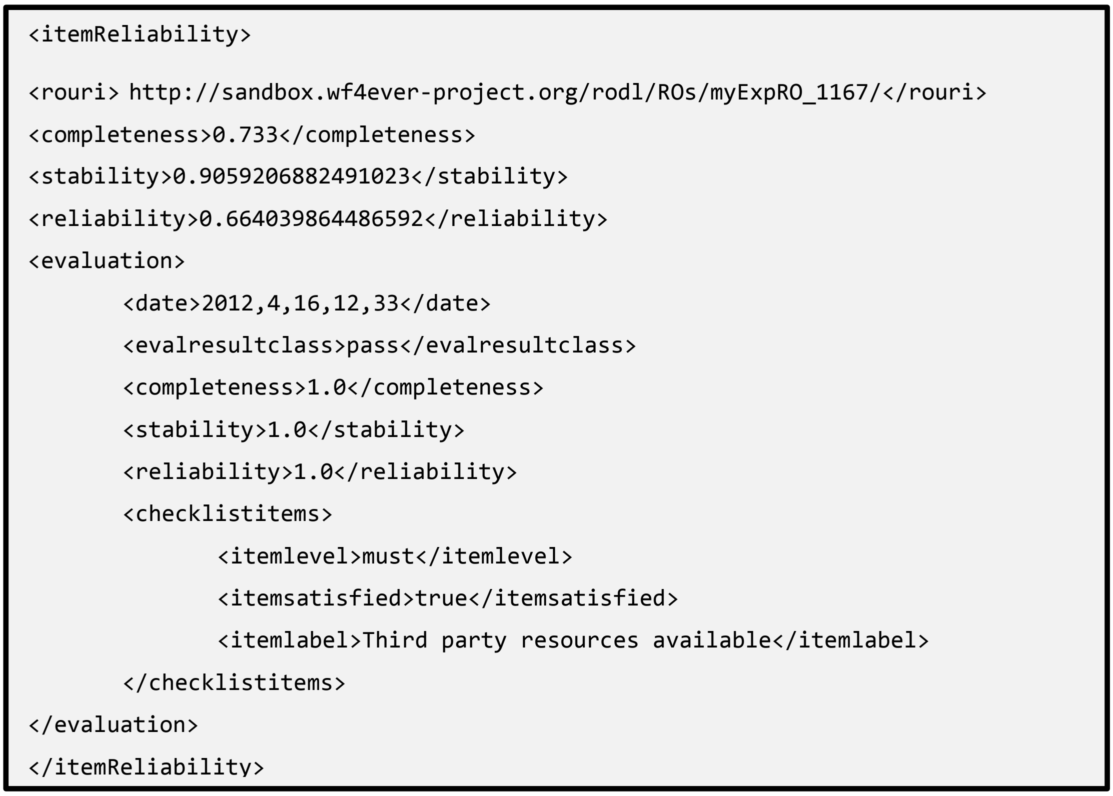
Figure 20 Sequence diagram for reliability evaluation, access, and notification services.

The Figure 20 shows the interaction between RODL, the checklist service, and the stability and reliability service during a typical call for obtaining the evaluation results of reliability and stability. This service in turn interacts with the RO through a daily executed evaluation storage component which uses the checklist service for retrieving the checklist results. Then the reliability service obtains these checklist results whenever an end user demands it and it calculates the completeness score for each checklist results. Afterwards the stability and reliability scores are calculated and returned to the client.

The data format of these results is shown in XML in the Figure 21 which includes not only the calculated scores for the three quality dimensions but also some explanations regarding the degree of satisfaction for the requirements specified by the completeness assessment. This kind of information provides some explanations of what happened at a

[71] http://sandbox.wf4ever-project.org/decayMonitoring/rest/notifications

specific time helping users to avoid decay (e.g. by fixing the decayed resource) or to improve the reusability of those ROs which have good quality.



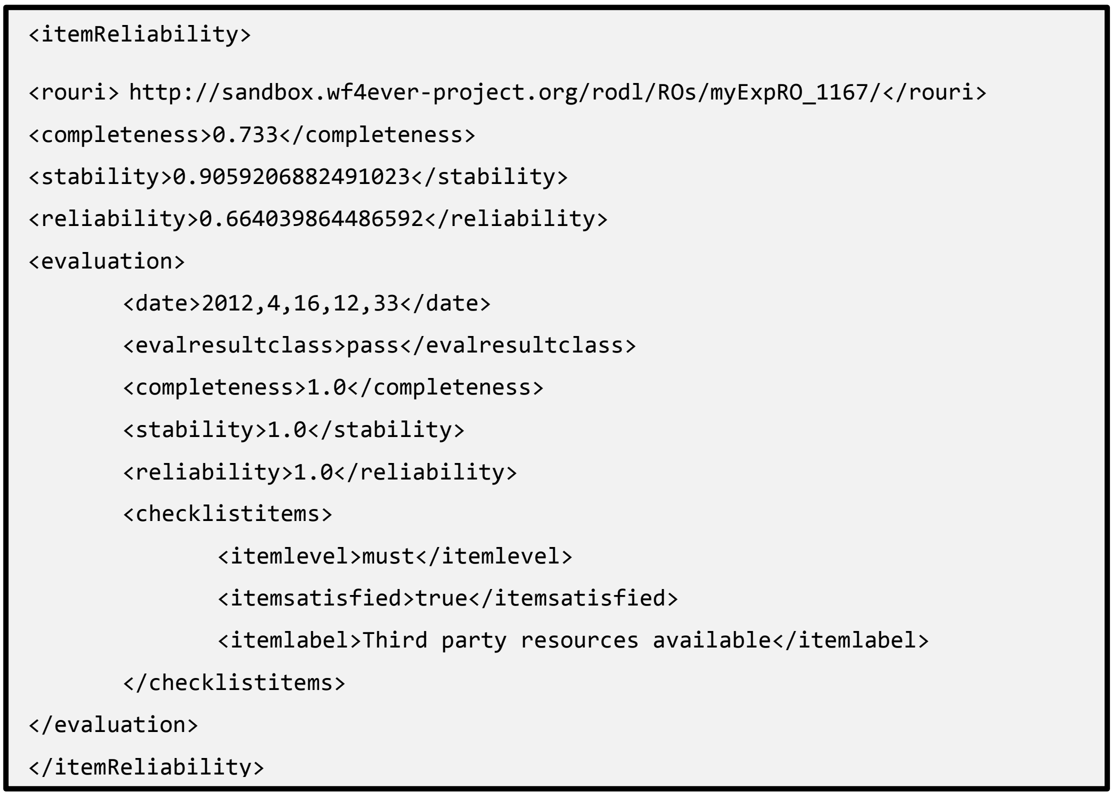
```
<itemReliability>
<rouri> http://sandbox.wf4ever-project.org/rodl/ROs/myExpRO_1167/</rouri>
<completeness>0.733</completeness>
<stability>0.9059206882491023</stability>
<reliability>0.664039864486592</reliability>
<evaluation>
       <date>2012,4,16,12,33</date>
       <evalresultclass>pass</evalresultclass>
       <completeness>1.0</completeness>
       <stability>1.0</stability>
       <reliability>1.0</reliability>
       <checklistitems>
              <itemlevel>must</itemlevel>
              <itemsatisfied>true</itemsatisfied>
              <itemlabel>Third party resources available</itemlabel>
       </checklistitems>
</evaluation>
</itemReliability>
```

Figure 21 Stability and Reliability evaluation results presented in XML format.

5.7 Presentation of data: RO-Monitoring Tool

The monitoring tool provides functionalities for time-based computation of the completeness, stability, and reliability scores of an RO and stores the results as additional metadata within the RO as shown in the Figure 23. Furthermore it also provides a visual friendly interface (as shown in the Figure 24) to explore daily evaluations of completeness alongside with its correspondent scores of stability and reliability providing a more comprehensive access to the quality information of a RO. The RO-Monitoring tool is available in the Wf4Ever sandbox[72].

[72] http://sandbox.wf4ever-project.org/decayMonitoring/visual.html



```
<itemReliability>
<rouri> http://sandbox.wf4ever-project.org/rodl/ROs/myExpRO_1167/</rouri>
<completeness>0.733</completeness>
<stability>0.9059206882491023</stability>
<reliability>0.664039864486592</reliability>
<evaluation>
       <date>2012,4,16,12,33</date>
       <evalresultclass>pass</evalresultclass>
       <completeness>1.0</completeness>
       <stability>1.0</stability>
       <reliability>1.0</reliability>
       <checklistitems>
              <itemlevel>must</itemlevel>
              <itemsatisfied>true</itemsatisfied>
              <itemlabel>Third party resources available</itemlabel>
       </checklistitems>
</evaluation>
</itemReliability>
```

Figure 22 Stability and Reliability evaluation results presented in XML format.

5.8 Evaluation of monitoring tool

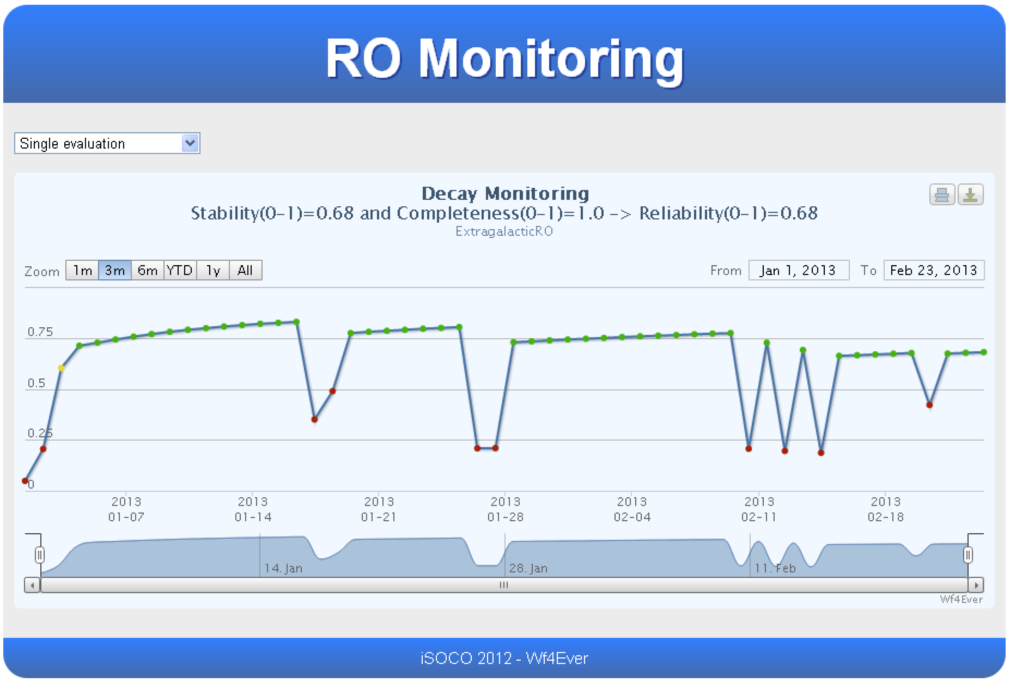
This section shows an initial evaluation work to validate how by using the quality criteria and the implemented presentation tools, which have the main goal of improving the user experience, the reusability of the scientific workflows can increases. For this purpose we have done a first evaluation of the monitoring tool to measure the benefit of using a historical perspective of the RO quality, represented by the reliability score, as opposed to instantaneous quality measures like the completeness score. We have simulated a year of changes over a hundred workflows based on the empirical data obtained on the study of “Why workflows break” [Zhao’12]. Those simulations were presented to a set of nine scientists that had access to the first 274 days of the history for each simulated RO and they had to decide if they would reuse that research objects or not by answering two questions: 1.Would you reuse this research object for your own experiments today?, and 2.Would you use it in three months from now?.

The same questions were asked for using only the completeness values and then we compared the obtained results with the actual behaviour of each workflow today and in three months.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://sandbox.wf4ever-project.org/rodl/ROs/Pack387/>

       <http://purl.org/wf4ever/rovalues#completeness> 1.0 ;

       <http://purl.org/wf4ever/rovalues#reliability> 0.869 ;

       <http://purl.org/wf4ever/rovalues#stability> 0.869 .
```

Figure 23 Metadata generated by the RO-Monitoring tool.

The results showed that that using the monitoring tool with the complete history and the reliability score we obtained a ratio improvement of 2.7 (51 better choices vs. 19 worsen) for in-the-day decisions and of 3.1 for in a few months question (69 better choices vs. 22 worsen) [Gomez'13].



RO Monitoring

Single evaluation

Decay Monitoring

Stability(0-1)=0.68 and Completeness(0-1)=1.0 -> Reliability(0-1)=0.68

ExtragalacticRO

Zoom 1m 3m 6m YTD 1y All

From Jan 1, 2013 To Feb 23, 2013

iSOCO 2012 - Wf4Ever

Figure 24 Wf4Ever RO-Monitoring Tool

These initial results confirm that the use of reliability information and the RO-monitoring tool enables scientists to make better decisions about the reuse of third party scientific research objects.

6 Conclusions

This document has presented the work done on the development of Wf4Ever integrity and authenticity (I&A) focusing on the Phase II period. During Phase II, the standardization effort of the World Wide Web Consortium (W3C)[73] for the creation of a provenance standard has been finished. The result of this effort has been the Family of W3C PROV standards, including the PROV-O ontology, in which Wf4Ever project members have been substantial contributors. We have also updated the provenance related vocabularies implemented in Wf4Ever to be aligned with this standard, allowing interoperability with other PROV implementations.

As part of the community building work, a so called provenance corpus has been generated by collecting provenance of workflow results from the two well-known Wings and Taverna workflow repositories, and submitted to the ProvBench initiative. The main goal of this corpus was to provide a set of provenance of workflow results samples for benchmarking purposes. Furthermore we have shown some applications which have used provenance such as the discovering of common workflow fragments on execution.

During Phase II we built upon our previous work on completeness and stability, and we implemented a new “reliability” evaluation. These three dimensions have been used in defining quality criteria for an RO. As result of the work done, both modelling and the creation of a quality framework including completeness, stability and reliability criteria, we have implemented and presented two main tools which provide end users with information regarding the quality of a research object, and establishing an indication of its reusability. These two tools are: i) the checklist service, which provides the current status of the different resources of a research object, and ii) the RO-Monitoring tool which provides not only the current quality status view of the research object, but also some historical information and the reliability score allowing to gain some insight into its possible future status. The implementation of the presented quality dimensions has been done by following the overall Wf4Ever methodologies and REST web services style.

Our ongoing work aims to validate the different implemented services and tools in real environments in order to verify the usefulness of the presented approach. We have

[73] http://www.w3.org/

already started this work for validating the reliability dimension and also some evaluation of the checklist service has been undertaken. Our intention is to continue this work (and extended it for instance including the may requirement type) in order to obtain feedback from end users and its application on real environments which will allow enhancing the current implemented quality criteria.

7 References

[Alp'13] P. Alper, K. Belhajjame, C. Goble, and P. Karagoz. "Small is beautiful: Summarizing scientific workflows using semantic annotations". In Proceedings of the IEEE 2nd International Congress on Big Data (BigData 2013), Santa Clara, CA, USA, June 2013.

[Cicca'11] P. Ciccarese, M. Ocana, L.J. Garcia Castro, S. Das, and T. Clark. "An open annotation ontology for science on web 3.0.", J Biomed Semantics, 2(Suppl 2):S4, 2011.

[D1.3v2] Raul Palma et. Al. "Wf4Ever Architecture – Phase II". Deliverable D1.3v2, Wf4Ever Project, 2013.

[D1.4v2] Raul Palma et. Al. "Reference Wf4Ever Implementation – Phase II". Deliverable D1.4v2, Wf4Ever Project, 2013.

[D2.2v1]: S. Bechhofer, Khalid Belhajjame, et. al., "Design, implementation and deployment of workflow lifecycle management components – Phase I". Deliverable D2.2v1, Wf4Ever Project, 2012. (Available at http://repo.wf4ever-project.org/Content/37/D2.2v1_Final.pdf).

[D2.2v2]: Khalid Belhajjame, et. al., "Design, implementation and deployment of workflow lifecycle management components - Phase II". Deliverable D2.2v2, Wf4Ever Project, 2013.

[D4.2v1]: Esteban García-Cuesta, et. Al. "Design, implementation and deployment of Workflow Integrity and Authenticity Maintenance components – Phase I". Deliverable D4.2v1, Wf4Ever Project, 2012," 2012. (Available at http://repo.wf4ever-project.org/Content/39/D4.2v1Final.pdf)

[Gar'13] Garijo, Daniel, Corcho Oscar, and Gil Yolanda , "Detecting Common Scientific Workflow Fragments Using Templates and Execution Provenance", Seventh ACM International Conference on Conference on Knowledge Capture, Banff, Canada, (2013).

[Gomez'13] José Manuel Gómez-Pérez, Esteban García-Cuesta, Aleix Garrido and José Enrique Ruiz, "When History Matters - Assessing Reliability for the Reuse of Scientific Workflows", in-use track held at ISWC2013 21-25 October, Sydney, Australia.

[Hales’06] B. Hales and P. Pronovost, “The checklist-a tool for error management and performance improvement”, Journal of critical care, vol. 3, no. 21, pp. 231-235, 2006.

[MIBBI]: C. Taylor, D. Field, S. Sansone, J. A. R. Aerts, A. M., B. P. Ball C.A., M. Bogue and T. Booth, “Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project”, Nature biotechnology, vol. 8, no. 26, pp. 889-896, 2008. (Available at http://www.nature.com/nbt/journal/v26/n8/pdf/nbt.1411.pdf)

[MIM]: Matthew Gamble, Jun Zhao, Graham Klyne, Carole Goble. "MIM: A Minimum Information Model Vocabulary and Framework for Scientific Linked Data", IEEE eScience 2012 Chicago, USA October, 2012.

[Zhao’12]: J. Zhao, J.M. Gómez-Pérez, K. Belhajjame, G. Klyne, E. García-Cuesta, Garrido A, Hettne K, Roos M, De Roure D, Goble CA. “Why Workflows Break - Understanding and Combating Decay in Taverna Workflows”. In the proceedings of the IEEE eScience Conference (eScience 2012), IEEE CS, Chicago, USA, 2012. (Available at http://users.ox.ac.uk/~oerc0033/preprints/why-decay.pdf)