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 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 RO.

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	16/07/2013	Esteban García-Cuesta	Conclusions added
0.5	17/07/2013	Esteban García-Cuesta	Improvements on introduction and conclusions, and overall review.
0.6			
0.7			
0.8			
0.9			
1.10			
1.11			
1.12			

Executive Summary

This is the last of two deliverables regarding the design, implementation, and deployment of Workflow Integrity and Authenticity and it includes the **updated prototypes of the different integrity and authenticity (I&A) components** of the project and provides a summary of the updated implementation of the I&A evaluation tools developed during the phase II. Due to this is the last deliverable of two and it is based on the previous work (following an incremental project development), it also contains some parts or references which were already discussed previously but we decide to include it for better understanding and to make this document self-contained.

Regarding the quality evaluation of a RO the updates on the two specific quality dimensions which were used to drive the design and implementation of the I&A evaluation tool, namely **completeness** and **stability** are described. Furthermore, the definition and implementation of a new dimension so called **reliability** which makes use of provenance of completeness information for providing a more user oriented and meaningful information regarding the quality of an RO is also included as part of the authenticity work.

Another important content of this document is the **provenance work** where the strongest effort has been put on the **standardization of provenance in the World Wide Web Consortium (W3C)** and on the creation of a big corpus of provenance, so called **ProvBench**, which makes use of wfprov ontology and updated provenance exporting plugins initially implemented during the Phase I.

Last, 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 on top of them to use its functionalities within Wf4Ever portal.

This deliverable should be read together with D1.4v2, D2.2v2, D3.2v2, and D4.2v1 in order to obtain a complete overview of the current state of the components implemented during phase II of Wf4Ever.

Table of contents

List of Figures

1 Introduction

This document provides a precise description of the software components produced during phase II of Wf4Ever in the context of WP4 (workflow integrity and authenticity maintenance). These components use the Research Object resources allocated in RODL (Research Object Digital Library) and the different models used for the definition of a Research Object for evaluating the RO overall quality and provide some meaningful information gaining insight of its current status (e.g. by showing explanations).

According to the DOW this deliverable: “will include the following functionalities: an updated Research Object provenance model that is the basis of the standardization process in existing international initiatives, and extended methods for computing integrity and authenticity, taking into account different granularities, and visualization tools for them”, and in the next we introduce how this has been accomplished extending the explanations throughout this document.

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]) we have been able to create a PROV-Corpus during the phase II of this work. This corpus, so called ProvBench, 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 also important to highlight that several contributors from Wf4Ever team have been involved in the World Wide Web Consortium (W3C) effort to create a standard for provenance which was completed in May of 2013.

Regarding the implementation of the I&A work, the main improvements can be summarized on the following: i) construction of a new Minim model based on new specified requirements, ii) design of new checklists types, iii), updated version of the evaluation completeness component to use SPARQL1.1 standard, iv) access to RODL repository for retrieving the ROs and its aggregated resources to be evaluated, v) implementation of a new dimension so called reliability, vi) extension of the Minim purposes definition to provide finer quality granularity, vii) providing and new visualization

tools for the completeness, stability, and reliability dimensions, viii) storing and providing accessibility to the provenance of the quality results as an aggregated resource (using ORE vocabulary) of the RO, and ix) new visualization tools for providing quality information of a RO to end-users focused on availability and reuse of a Research Object. Among other useful information provided by the implemented tools is worth to highlight the importance of collecting the provenance of the different quality dimension scores to provide a historical view of the RO quality which turned up into better reusability user experience.

Furthermore, we have 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-O[1] standardization. Section 3 describes the general framework for evaluating the quality of a RO describing the interaction between the models (qualitative information) and the visualization 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 and 5 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 prototype we have made some decisions about the technical environment within which Wf4Ever is being deployed. These are:

- 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

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

concepts, object types, etc. Concepts and entities manipulated by Wf4Ever are preferably identified using URIs

- Interfaces of the developed components have used HTTP/RESTful
- Research Objects (RO) are the main piece of information used which are the digitalization of a scientific experiment
- An RO contains metadata about provenance of its lifecycle, and also about its execution
- The provenance information has been modelled by the evolution ontology (roevo) and the provenance of workflow results ontology (wfprov) by using OWL[2].

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 relation with the overall integration of the project and user interfacing aspects like RO visualization, being 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], D3.2v2 [D3.2v2], and D4.2v1 [D4.2v1].

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

2 Provenance

Provenance collects information about entities, activities, and people involved in producing a piece of data (in our project a research object), which among others can be used to form assessments about its overall quality, reliability or trustworthiness. An overview of a family of provenance information focusing in making them inter-operable can be found at PROV-Overview[3].

2.1 Provenance in Wf4Ever

In Wf4Ever there are two main types of provenance which have been modeled 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 experienced in the different stages of their lifecycle.

The provenance of artifacts 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 following the Annotation Ontology. This provenance is expressed using the wfprov ontology[4], which is part of the RO Model[5] which also is in turn defined as a refinement of the W3C PROV-O ontology[6].

Regarding the provenance of the Research Object evolution, along with its possible origins in previous work, is captured through the Research Object Digital Library RODL[7], and keeps track of the lifecycle of an RO. This provenance is represented

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

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

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

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

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

using the roevo ontology[8] which also is defined as a refinement of the W3C PROV-O ontology. We want to point out that the description of the wfprov ontology and roevo ontology were introduced and described in D4.2v1 Design, implementation and deployment of Workflow Integrity and Authenticity Maintenance components – Phase I [D4.2v1] in the WP4 context and can be consulted there.

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]. 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 resource may describe provenance of multiple target resources, and an application that consults it does not need to know about the ro:annotatesAggregatedResource link in order to properly interpret the provenance information.

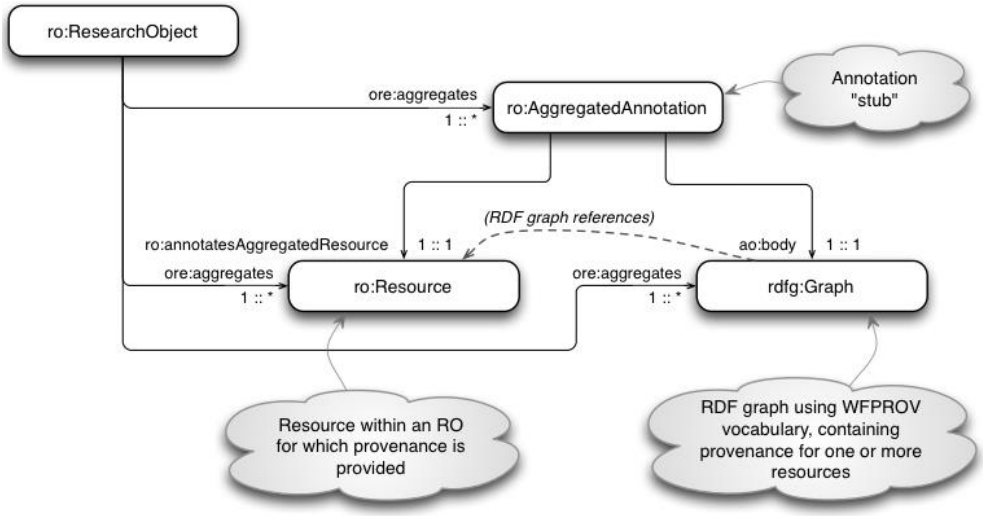


Figure 1 Provenance of workflow results

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

The provenance resource itself (the rdfg:Graph value) need not be part of the RO aggregation (i.e. it may be an external resource), but for practical purposes in our work an annotation body is generally treated as part of the RO aggregation.

The second type of provenance associated to Research Objects is captured in the description of Research Object Evolution (roevo). This type of provenance 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 looking relations are color coded in blue, and the historical provenance relationships are colored in red as can be seen in Figure 2. In the next we described how to access to this provenance.

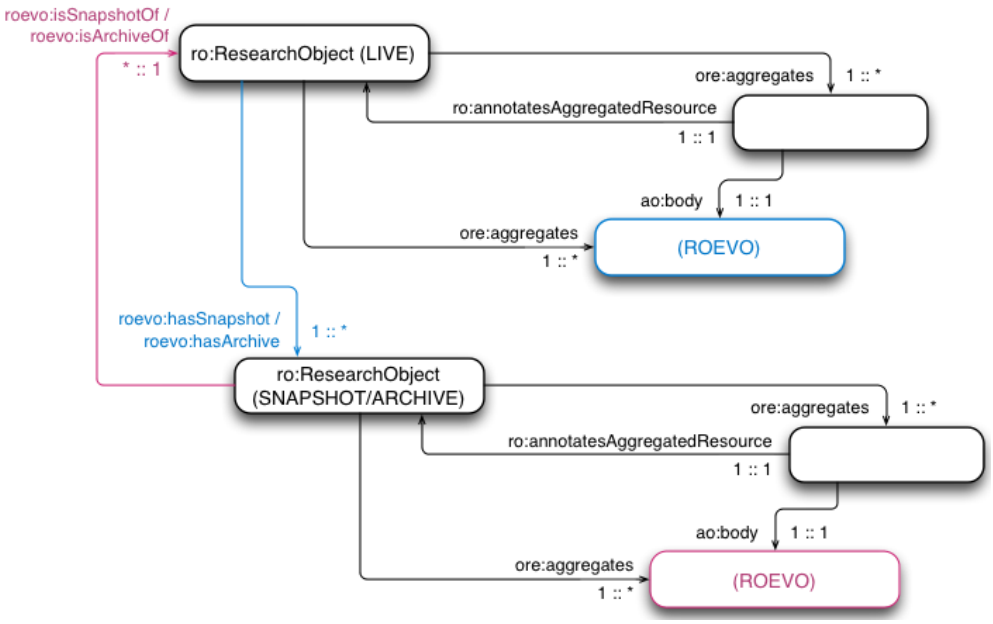


Figure 2 roevo provenance diagram

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

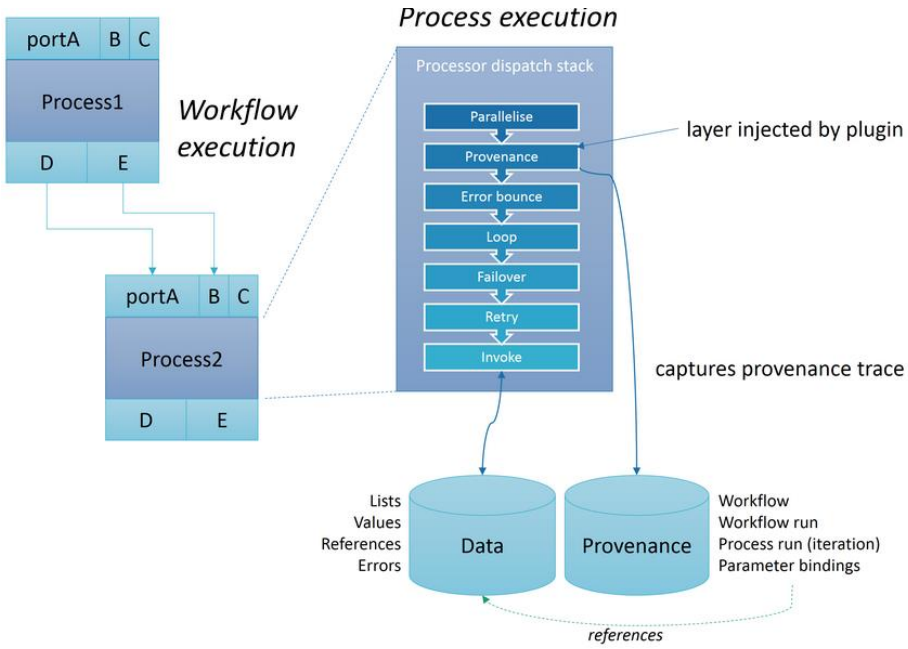
the RO manifest, and creates a single RDF merged annotation graph of the entire provenance and other information thus obtained. Provenance information can then be tested by suitably constructed SPARQL queries that are evaluated against the merged annotation graph.

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.

So far it has been explained how to model provenance and how to access to that data once it is stored but we have not introduced how to obtain that data which has been mostly provided by Taverna[9].

2.4 Taverna provenance export tools

Taverna executes workflows and therefore 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 workbench to populate previous runs and intermediate results in the results view. The Figure 3 shows the current Taverna provenance architecture.



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

Figure 3 Taverna provenance architecture.

During execution of a Taverna workflow, the dispatch stack[10] is responsible for the execution logic of an individual process invocation, with layers like *parallelize* 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.

The provenance trace has been used by the implemented Taverna-PROV plugin[11] to export the workflow run, including the output and intermediate values, and the provenance trace as a PROV-O RDF graph[12] 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[13]. The Taverna-PROV ontology[14] 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[15]. 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.

---

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

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

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

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

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

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

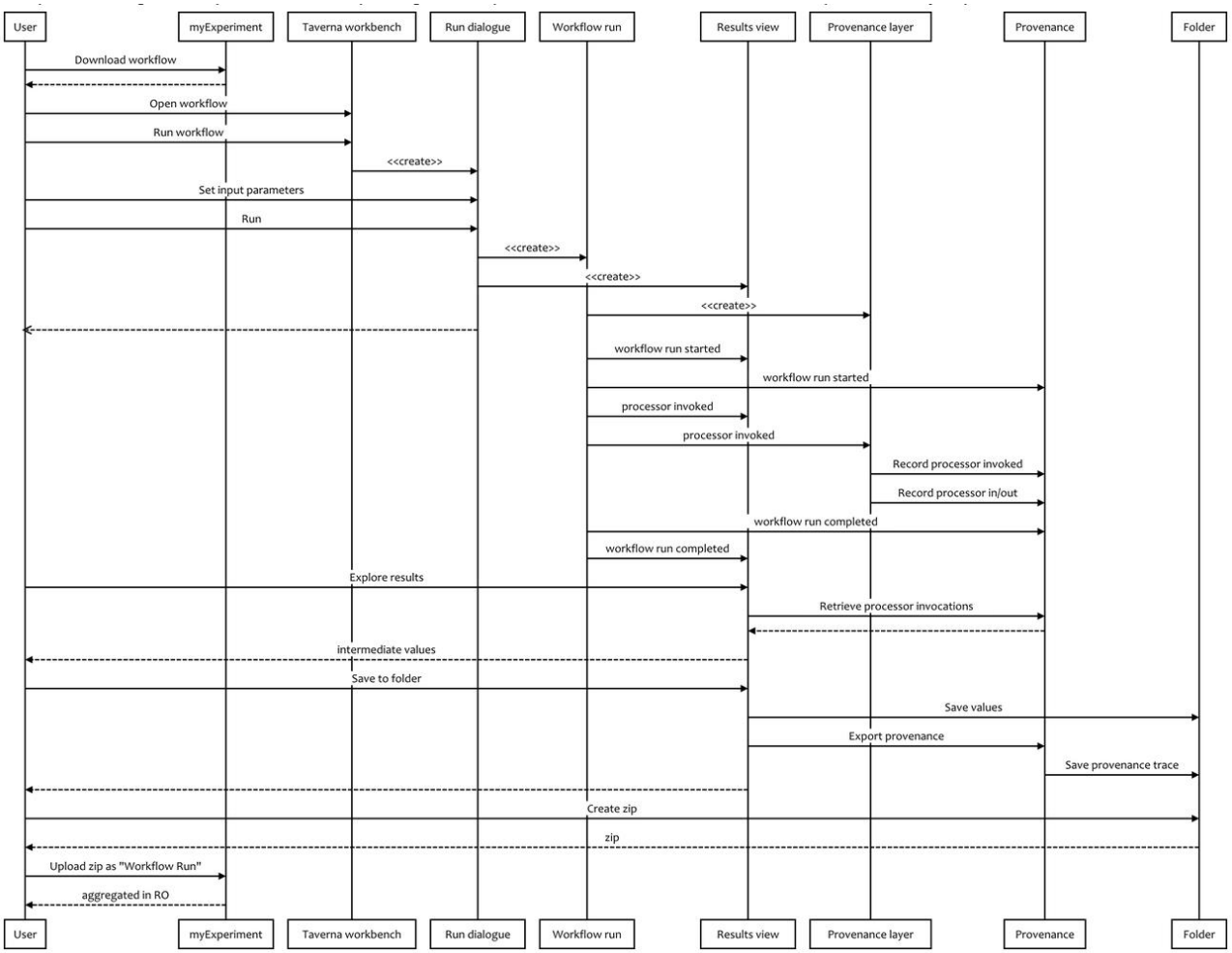


Figure 4 Taverna provenance export plugin sequence diagram

2.5 Provenance applications

Within the Wf4Ever project, provenance information has been used for different purposes as it is described in the next:

RO Portal Visualization

The RO Portal[16] 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.

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

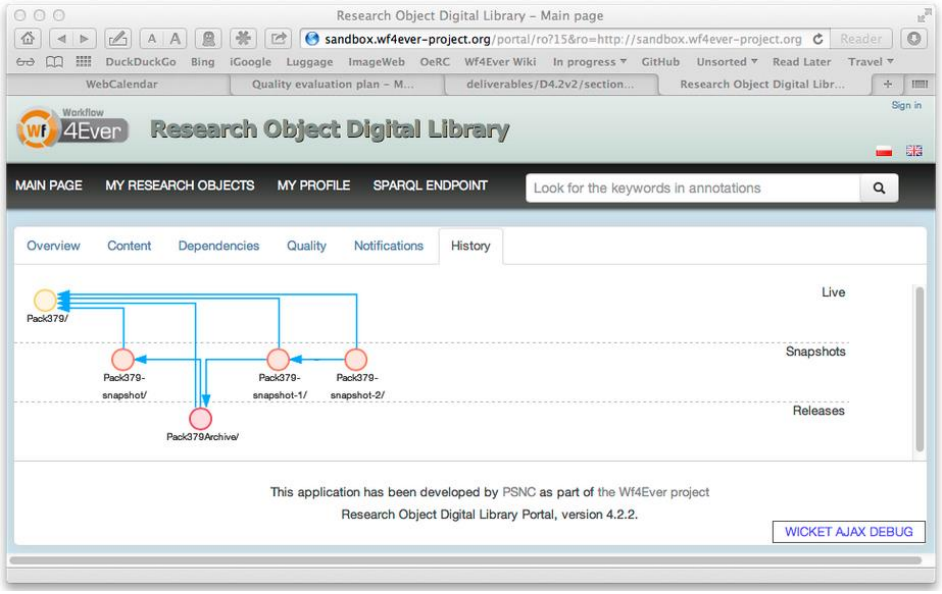


Figure 5 roevo visualization at RO Portal

myExperiment

The provenance information has also been included as a mockup of workflow run view in alpha-myExperiment[17] and it will be upgraded to provide a high-level overview of wfprov on each RO resource page. The mockup reveals if there are workflow runs in the research object and shows the text based inputs and outputs for each run, and the execution information as shown in Figure 6.

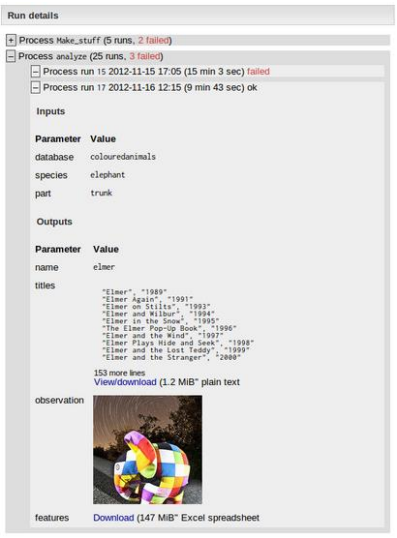


Figure 6 Provenance of workflow results mock-up included in alpha-myExperiment

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

Assessment of Keeg workflows

Provenance information has been also used in the assessment of decay in KEGG workflows (KEGG: Kyoto Encyclopedia of Genes and Genomes[18]), specifically to locate the input data used to create additional RO annotations tested by the checklist evaluation. For this purpose, provenance information was extracted from a Taverna-generated provenance trace using a command line SPARQL query tool[19]. En example of a script of how to incorporate the provenance traces and convert KEGG workflows to ROs in preparation to using the checklist service to perform decay detection can be also found at[20].

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 relating them to templates. This approach has been tested with a dataset of workflows published by Wings[21]. The obtained results showed that by using these kinds of abstractions we can highlight the most common abstract methods used in the executions of a repository, relating different runs and workflow templates with each other [Daniel'13].

Provenance summarization

The use of provenance of workflow results is suitable for several applications as veracity analysis though it adds complexity and large volume of data to be computed. These problems can be alleviated by applying a reduction approach of this large volume of data to obtain a summary of the execution. For this purpose we obtained a

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

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

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

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

set of primitives which identifies uniquely the different parts of the executed workflows and allows summarizing them by a set of those primitives almost without losing its effectiveness for final applications use [Pinar'13].

Quality assessment

The evaluation of the completeness quality dimension of a 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 RO annotation. The Figure 7 shows the visualization of this checklist evaluation including the verification of provenance existence ("Workflow run found").

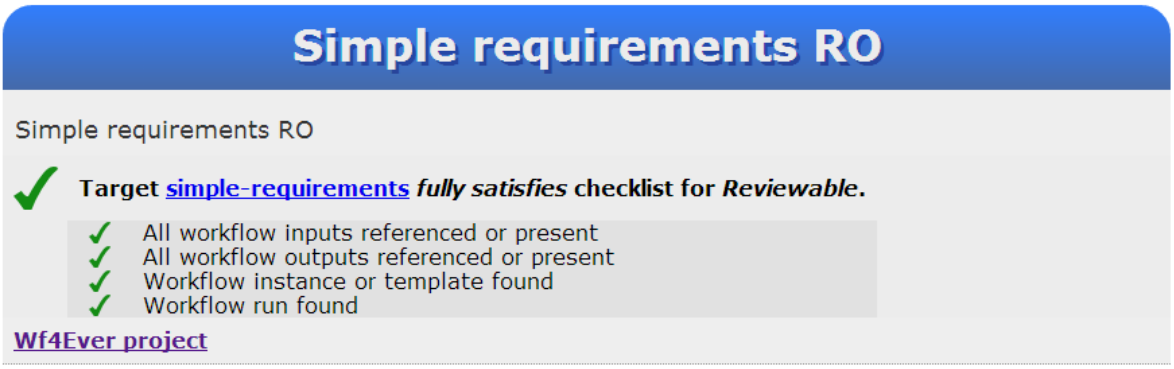


Figure 7 Provenance verification for quality assessment

2.6 Provenance community engagement

As part of the provenance work there were some activities which promoted its use at community level as provenance standardization and the creation of a PROV-corpus.

Provenance standardization in W3C

The World Wide Web Consortium (W3C)[22] 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

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

[PROV-Overview][23]. 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[24].
- PROV-DM - the PROV data model for provenance PROV-DM[25].
- PROV-N - a notation for provenance aimed at human consumption PROV-N[26].

Wf4Ever members have been also co-editing or contributing to the next supporting working group documents: PROV-PRIMER[27], PROV-AQ[28], PROV-DICTIONARY[29] and PROV-DC[30]. Furthermore, at the time of their publication, there were over 60 documented implementations ([PROV-implementations][31]) 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. Therefore, it is worth to highlight that the Wf4Ever project made significant contribution to this early adoption of the new provenance standards.

ProvBench Challenge

The ProvBench[32] initiative objective was to bootstrap the publication of provenance information in an open and accessible fashion. The first ProvBench event was held at

---

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

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

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

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

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

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

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

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

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

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

the 6th International Conference on Extending Database Technology (EDBT)[33], as part of the First International Workshop on Managing and Querying Provenance Data at Scale (BIGProv'13)[34]. This inaugural event received 8 submissions[35] from diverse interested research groups, including one from Wf4Ever which is explained in the following.

Wf4Ever provenance corpus

We have generated a provenance corpus, so called ProvBench[36] whose dataset can be found at[37], and collected 120 real provenance of workflows results from two well-known scientific community platforms, Taverna and Wings, and are associated to 12 different applications domains. The provenance traces have been specified by using the PROV-O ontology and terms from other vocabularies as RO model and OPMW have been also used for the association between the provenance with their corresponding workflow description.

The workflows associated to Taverna platform have been generated by automatic capture of provenance by using the developed provenance plug-in[38] (see section 2.4 for further details) which provides PROV-O 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[39] to the first ProvBench event.

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

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

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

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

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

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

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

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

- 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],
- discovery of execution pattern similarities and linking of similar scientific experiments,
- identification of patterns of use for obtaining dependencies recognition and provide recommendation,
- verification of replicability of previously certified results,

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?

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).

3 Quality Evaluation in Wf4Ever

This section introduces general framework designed and implemented in Wf4Ever which have provided the needed information for the establishment of a quantitative measure of the different dimensions (completeness, stability, and reliability) identified as very important for the definition of an overall quality RO criteria [D4.1, D4.2v1].

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.

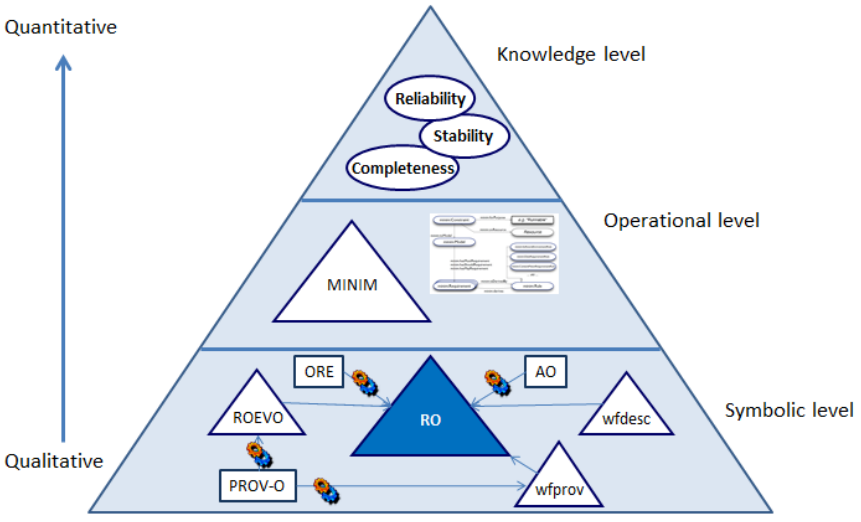


Figure 8 Quality ontology pyramid

The bottom layer of the pyramid spans across the main resources included in a research object and can be classified mainly as aggregations of several information resources, built on top of the ORE vocabulary and annotations which follows the Annotation Ontology. This layer corresponds to the RO model, described in the RO model specification RO model[40]. This layer is also the placeholder of information related to the workflow included in the research object, in terms of the wfdesc ontology, and of the provenance of execution, following the wfprov ontology defined as an extension of the PROV-O standard. The roevo ontology is built upon wfprov the roevo ontology and enables the representation of the different stages of the RO life-cycle, their dependencies, changes and versions.

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.

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

Regarding the advances accomplished since M20 we want to highlight the implementation of the above introduced quality framework that unifies the two previously work on completeness and stability, and also includes the new dimension so called reliability. Also, the individual dimensions have been improved by incorporating new functionalities as it is described in the next sections (e.g. new rules and tests), and a new set of presentations for visualizing the quality of a research object have been developed such as the new RO-Monitoring tool or the checklist verification service.

4 Completeness Evaluation

4.1 Introduction

In Wf4Ever the completeness evaluation has been accomplished by implementing checklists in order 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 apply 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[41].

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 implemented or updated the following parts:

[41] 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 also described by a Minim model. Afterwards the results of an assessment are presented using the Minim results model. In the next we described these models.

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 my 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 supported in the same single implementation. The Minim ontology[42], its specification[43], and its OWLDoc documentation[44] are maintained in a GitHub project[45].

The main elements of the Minim model are:

- **Checklists Model**: 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 it is intended to serve as shown in the Figure 9.

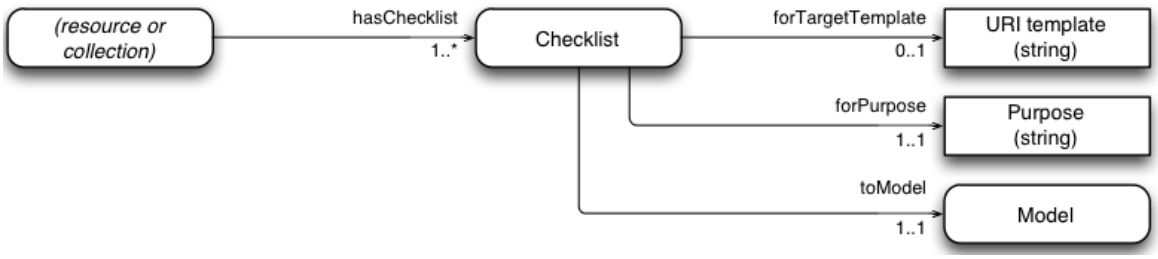


Figure 9 Checklist Model Diagram

- **Minim Models**: a Minim Model defines a list of requirements to be satisfied, which can be of three different types: mandatory (hasMustRequirement), desirable (hasShouldRequirement), or optional (hasMayRequirement) (see Figure 10).
    - **Requirements**: denotes some specific requirement to be satisfied by a Research Object, such as the presence of certain information about an experiment. For example, we may wish to test not only that a suitable reference

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

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

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

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

to input data is provided by an RO, but also that the data is live (accessible), or that its contents match a given value (integrity).

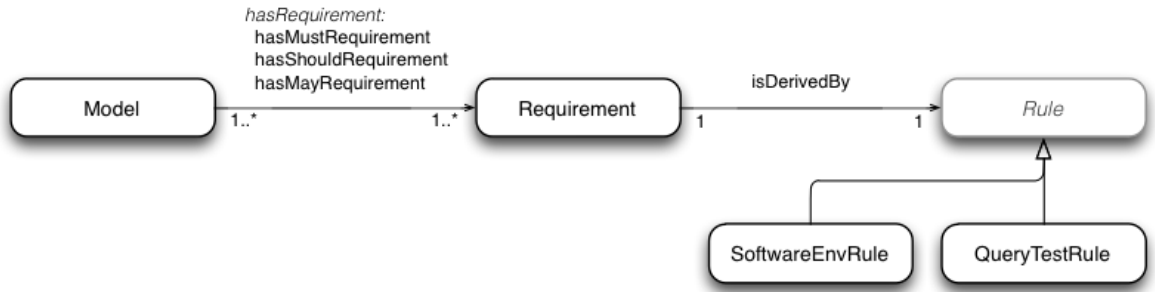


Figure 10 Minim Model Diagram

- **Rules**: 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 Ressearch 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.

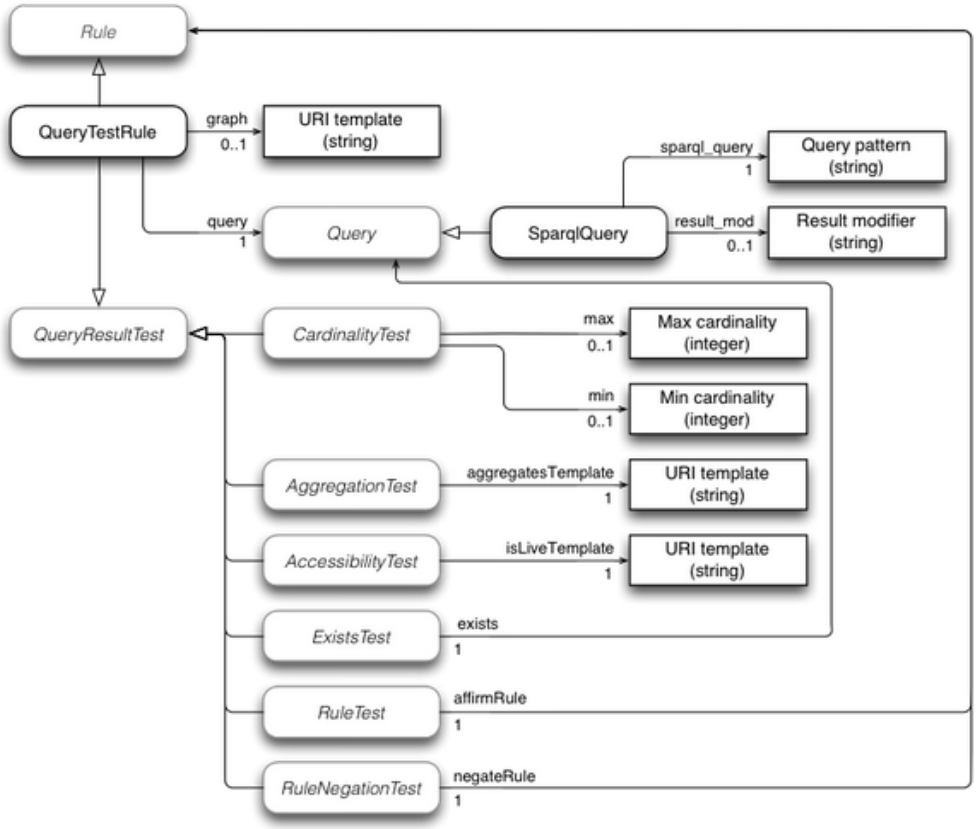


Figure 11 Rule Model Diagram

The Minim model provides de capability of being further subclassed in one the next three classes to add new testing capabilities:

- **Rule**: new rule types can be introduced to perform tests for new kinds of requirement that cannot be handled within existing 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).
- **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.

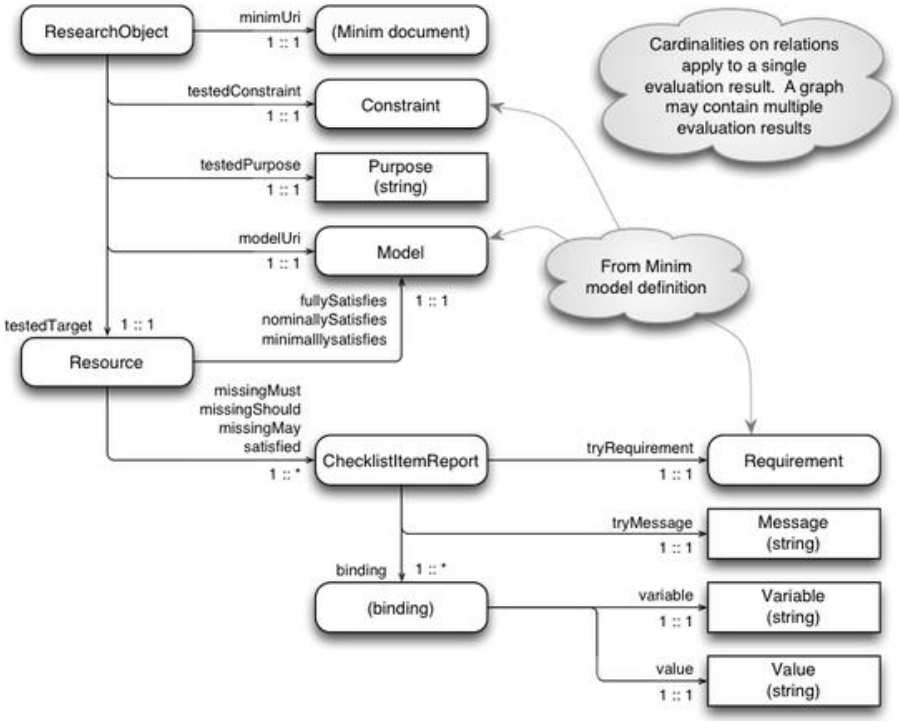


Figure 12 Minim Results Model

The main result of a checklist evaluation is an indication of whether a target resource **fullySatisfies**, **nominallySatisfies**, or **minimallySatifies** the associated checklist, evaluated in the context of a particular research object. By fullySatisfies we mean that all MUST, SHOULD and MAY requirements are satisfied indicating that the completeness is maximum; by nominallySatisfies we mean that all MUST and SHOULD requirements are satisfied indicating that the RO is complete for the main purpose that is

defined and also have some desirable characteristics, and minimallySatifies means that all MUST requirements are satisfied indicating that it is complete only for the specified purpose.

The model also allows a breakdown of the checklist evaluation result by using missingMust, missingShould, missing May 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. The Figure 13 shows an example of a Minim requirement that test 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 it returns the result shown in the Figure 14 for the target resource N-Methylformamide[46] for which no synonym exists. That results describes that the RO satisfies minimally and nominally the requirements of the Minim Model, and that there are some may requirements which are not being accomplish as can be seen explained by the property missingMay.

```
<http://purl.org/net/chembox/N-Methylformamide>
    minim:minimallySatisfies :minim_model ;
```

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

```
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 metric 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 users and client applications accessibility to the checklist evaluations. The checklist evaluation service is implemented as part of the codebase for RO Manager [D2.2v2], which is implemented in Python, and is available as an installable package at[47], and its source code can be found at[48].

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[49,50]. We want to point out that the command line version of checklist evaluation has been used mainly for development purposes and in the next we are going to describe the web service deployment.

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

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

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

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

4.5 Service interface and interactions

Overall, the Wf4Ever architecture [D1.4v1][D1.4v2] 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[51].

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.

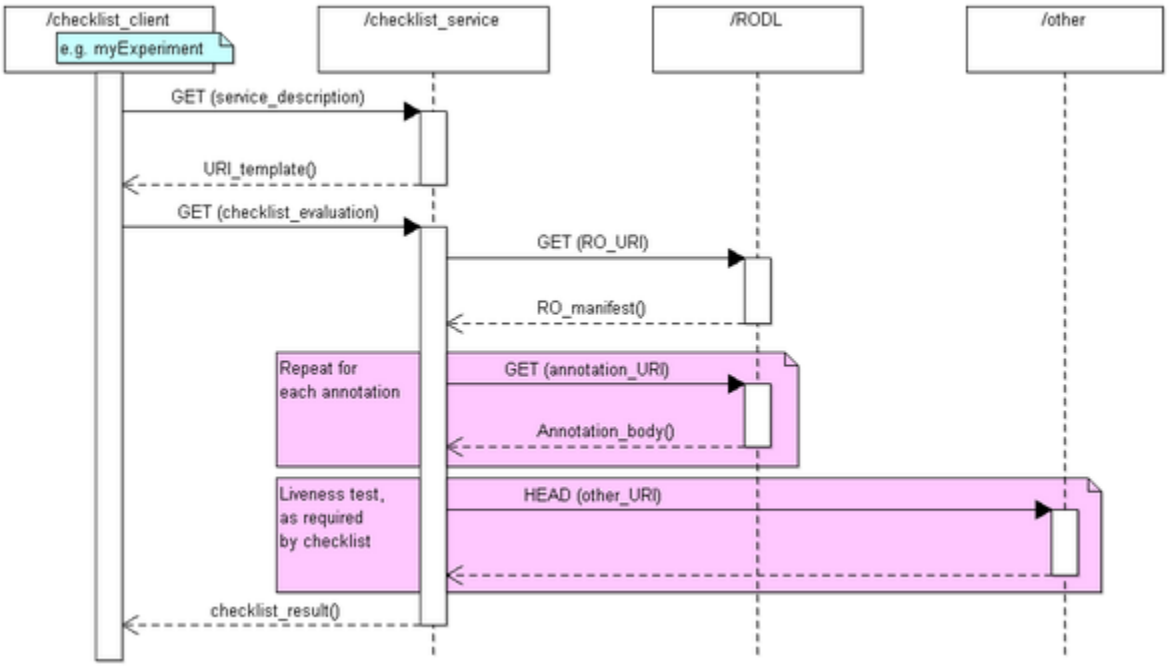


Figure 15 Sequence diagram of the checklist service.

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

4.6 Completeness Applications

In this section we briefly described 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[52] announced (2012) that they were introducing a REST interface for their discovery service, and discontinuing the older web Services based interface. Due to there are a number of workflows in myExperiment that use the older KEGG services we decided to use this movement 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. As results of this study we obtained a set of results indicating that decay (or failures e.g. KEGG web service has been withdrawn) and its visualization for an specific sample can be seen in the Figure 16.

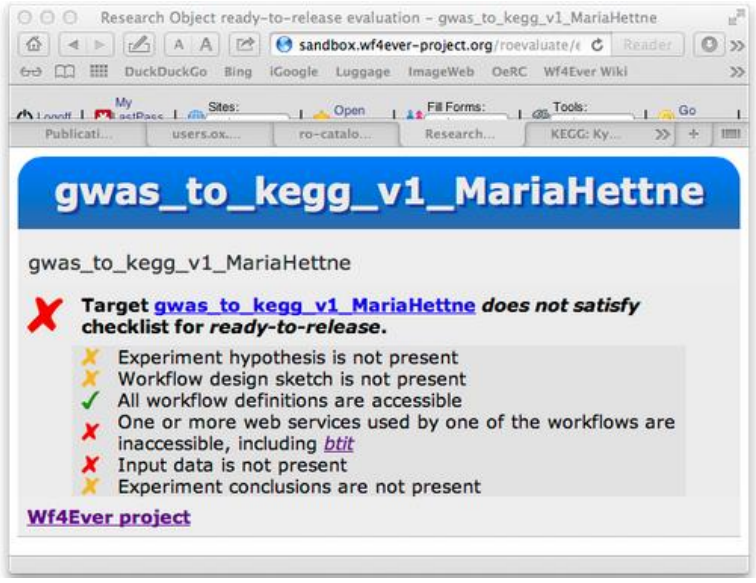


Figure 16 Checklist service visualization of KEGG service.

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

Completeness assessment for workflow decay prevention

This application focus on the creation of a checklist that can be used for testing the presence of information to support workflow re-use and repair with the main goal that such a checklist can be incorporated into the practices of workflow creation and use 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 (and all the other models described in this section) and in earlier work where we analyzed 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 checklist such as checklist-runnable.rdf[53] and workflow-experiment-checklist.rdf[54], which provides similar assessments to that shown in Figure 16.

Completeness assessment of resource descriptions: chembox

This application evaluates the completeness of resource descriptions for external sources. 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[55] was created and used, jointly with a script[56] for automatically perform the evaluations. The results of this study are available at[57].

Basis for stability assessment

As shown in the previous section "Quality Evaluation in Wf4Ever" the checklist service has been considered for the static analysis of Research Objects, but furthermore has

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

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

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

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

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

also been used as the basis for the stability and reliability assessments for considering the dynamic analysis of Research Object. How the completeness score is used and its interpretation in that context is described in the next section.

5 Stability/Reliability Evaluation

5.1 Introduction

In Wf4Ever the stability and the reliability assessments have been accomplished by implementing two REST services which use the information provided by the completeness assessment (explained in section 5.2) during a concrete previous period of time of the RO lifetime. This dynamic analysis has been adopted due to workflows (which are the executable resources of a RO) can break throughout the time unexpectedly and therefore taking into account only the static perspective, i.e. the current RO state, would provide a bias view of the RO. In [zhao'12] we saw that most of the times this decay is due to the volatility of some of third party resources which furthermore means that it cannot be controlled or predicted locally and are not easy to recover.

Because of the above mentioned problem, the stability and reliability metrics aim to keep track and measure the changes of the completeness assessment of a RO throughout the 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 run it). This is the reason why stability and reliability uses the completeness as baseline because it provides the definition of what exactly means correct behaviour of a RO which is defined by the Minim Model and the set of requirements that it incorporates.

While the completeness evaluation (introduced and explained in the previous section 4) allows identifying the different reason of decay by running a checklist service it against a RO, the stability and reliability add a new parameter to be considered, the time. The inclusion of this new parameter allows developing a new model which reflects how much the user should trust a Research Object for reusing purposes. Therefore the stability measures the ability of a workflow to preserve its completeness state during the RO lifetime, and extending this approach to include also the completeness assessment allows the 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.

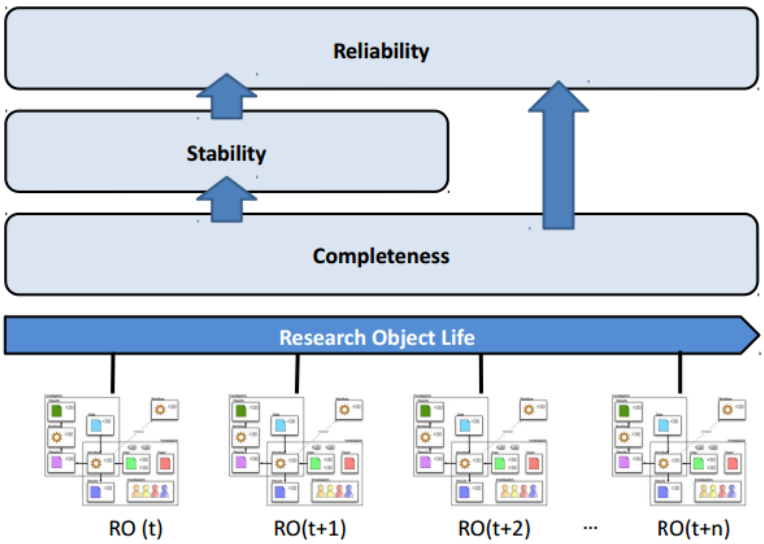


Figure 17 Layered Components of Reliability Measurement

In the next 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 Assessment

The completeness dimension evaluates the extent to which a workflow satisfies a number of requirements specified in the form of a checklist following the Minim OWL ontology. Such requirements can be of two main types: compulsory (must) or recommendable (should). In order to be runnable and reproducible all the must

requirements associated to a workflow need to be satisfied while should requirements propose a more relaxed type of constraint. 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 can be parameterized and 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]. Once identified the band, we define a normalized value of the completeness score 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.

5.3 Stability Assessment

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:

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

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 Assessment

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:

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

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 subsumed in the reliability score the provided service is unique. Also the checklist evaluation service is subsumed into the reliability and internally it is called for accessing to the ROs stored in RODL as shown in the Figure 21.

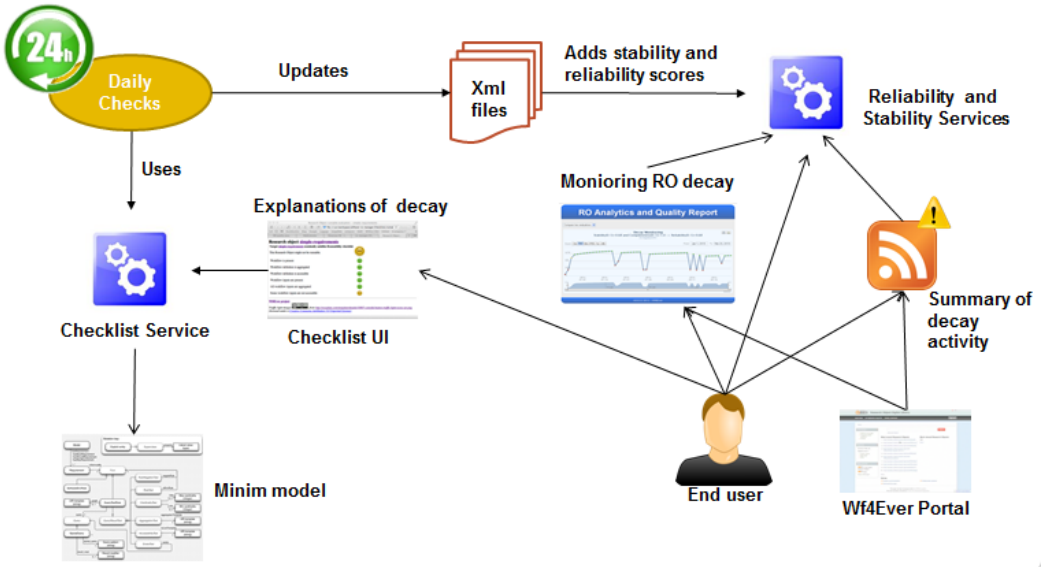


Figure 18 Wf4Ever quality assessment components interactions

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 on a file) and the purpose definition are also other needed parameters for obtaining the reliability score which have to be defined previously. 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 is the input for the calculation of the stability and reliability scores (one value per day as shown in the Figure 22).

We furthermore have implemented an alternative for accessing to the obtained results via notifications. These notifications provide a warning flag including a short summary of completeness, stability and reliability scores for those days where their quality scores change significantly due to some type of decay. The used notification format is ATOM standard and users can subscribe to the service[58] in order to get those notifications (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).

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 passing the RO URI as parameter within the request. A complete description of the API can be found at the Wf4Ever project wiki page[59].

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

[59] 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[60].

```
<trace>
<rouri> ro=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 Evaluation results for a research object presented in XML format.

Also, all the data resulting for the application of any of the quality criteria explained in this section is stored in XML and JSON formats and are provided upon the indicated accept header being both compliant with the Wf4Ever interoperability rules.

The Figure 21 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 uses these results whenever an end

[60] http://sandbox.wf4ever-project.org/decayMonitoring/rest/notifications?ro=http://sandbox.wf4ever-project.org/rodl/ROs/myExpRO_1167/&from=2013-1-3T14:30:00&to=2013-31-6T14:30:00

user demands it and it calculates the completeness score for each checklist results and afterwards also the stability and reliability scores being then returned to the client.

The data format of these results are shown in XML in the Figure 20 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 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>ro=http://sandbox.wf4ever-project.org/rodl/ROs/myExpRO_1167/</rouri>
<completeness>0.733</completeness>
<stability>0.90</stability>
<reliability>0.66</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 20 Stability and Reliability evaluation results presented in XML format.

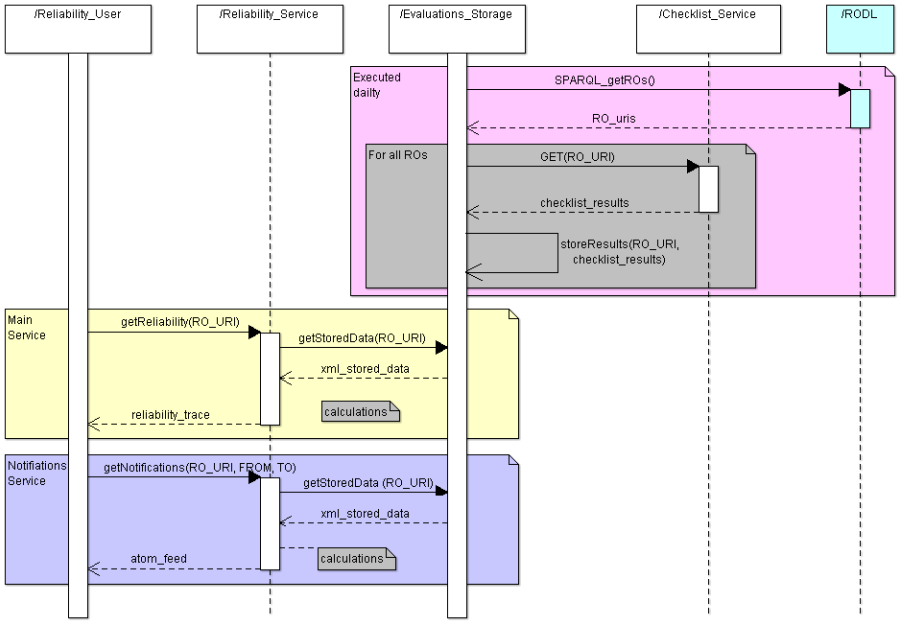


Figure 21 Sequence diagram for reliability evaluation, access, and notification services.

5.7 RO-Monitoring Tool

The monitoring tool is a web-based visual and friendly interface for exploring daily evaluations of completeness alongside with its associated values of stability and reliability providing more comprehensive information to the end users. This visualization is shown in the Figure 22 and has been implemented using JavaScript and jQuery libraries. The x axis of the graph represents the time and the y axis the reliability score. This tool is also interactive allowing the users to inspect the values of the quality metrics by clicking on any of the drawn points and getting an overview of the set of rules that were evaluated and its impact in terms of reliability.

The monitoring tool is integrated in the Wf4Ever Sandbox[61] and is available at[62] being the detection of decay its main application.

[61] http://sandbox.wf4ever-project.org

[62] http://sandbox.wf4ever-project.org/decayMonitoring/monitor.html

Despite there is a deliverable at M36 “Final evaluation report of the workflow integrity and authenticity maintenance components” we already have started a first evaluation of the monitoring tool to measure the potential benefit of using it for improving the reusability satisfaction of end-users by using not only the instantaneous quality measurement but also the historical information.

The study hypothesis was that using this reliability criteria vs. an instantaneous completeness criteria the users could choose better what RO they should use upon if they suffer of decay or not. We simulated a scenario where the ROs suffer different types of decay throughout the time as was identified in [zhao’12] and then we tested by asking end users if they would reuse it or not for doing some experiments today? or in a few months?. The obtained results show that the ratio of improvement is 2.7 (51 better choices vs. 19 worsen) for in-the-day decisions and 3.1 for in a few months question (69 better choices vs. 22 worsen) [gomez’13].

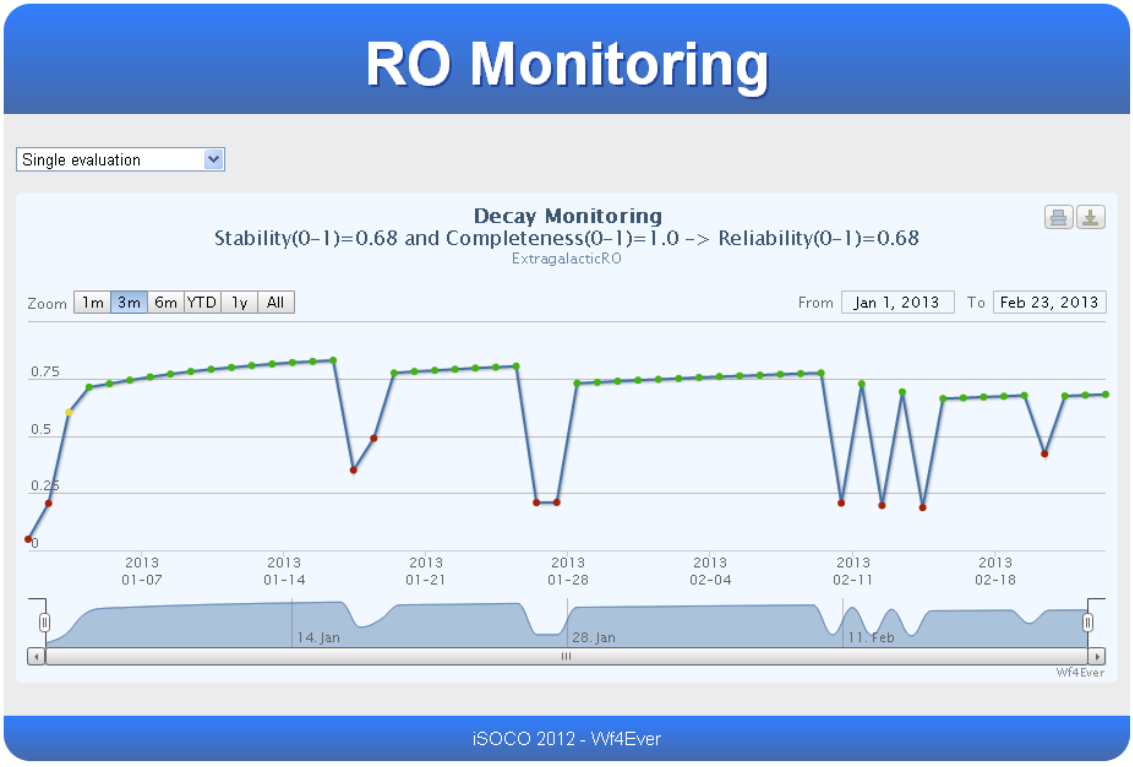


Figure 22 Wf4Ever RO-Monitoring Tool

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

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 of time. By Phase II we enhanced the two already previously defined dimensions, completeness and stability, and we implemented a new one so called reliability. These three dimensions have been used for defining the quality criteria of a 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 to the end users with some information regarding the quality status of a research object and establishing a more truthful 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 (following a thumbs up or thumbs down metaphor), 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 reliability scores allowing to gain some insight into the near future RO trustworthiness.

Regarding the implementation of the presented quality dimensions the overall Wf4Ever methodologies and REST web services approach have been adopted providing also the results in both XML and JSON format for improving the services interoperability.

By Phase II it also has been finished the standardization effort of the World Wide Web Consortium (W3C)[63] for the creation of a provenance standard. The result of this effort has been the W3C PROV-O standard ontology where several members of the Wf4Ever project have contributed. We also assured that the provenance related vocabularies implemented in Wf4Ever are compliant with this standard making a machine-processable alignment with it. Furthermore, as part of the community building work, a so called ProvBench provenance corpus has been generated by collecting provenance of workflow results from the two well-known Wings and Taverna workflow repositories. 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.

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

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 already started this work for validating the reliability dimension [Gómez'13] and also some preliminary steps have been taken for the checklist service. Our intention is to continue this work 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

[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.

[D2.2v1]: S. Bechhofer, Khalid Belhajjame, et. al., "Design, implementation and deployment of workflow lifecycle management components – Phase I. Deliverable D2.2v2, Wf4Ever Project, 2013," 2013.(Available at http://repo.wf4ever-project.org/Content/37/D2.2v1_Final.pdf).

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

[D4.2v1]: Esteban García-Cuesta, Jun Zhao, Graham Klyne, Aleix Garrido, Jose Manuel Gomez-Perez, "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)

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

[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.

[D2.2v2] S. Bechhofer ,et. Al. D2.2v2 "Design, implementation and deployment of workflow lifecycle management components– Phase II".

[D4.2v1] Esteban García-Cuesta et. Al. Design, implementation and deployment of Workflow Integrity and Authenticity Maintenance components – Phase I (Deliverable D2.2v2, Wf4Ever Project, 2012.)

[D4.1] Jun Zhao, et. Al. "Workflow Integrity and Authenticity Maintenance Initial Requirements" (Deliverable D4.1, Wf4Ever Project, 2011.)

[D1.4v2] Raul Palma et. Al. “Reference Wf4Ever Implementation – Phase II” (Deliverable D1.4v2, Wf4Ever Project, 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.

[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.ç

[Pinar’13] Pinar Alper, Khalid Belhajjame, Carole Goble, and Pinar Karagoz. “Small Is Beautiful: Summarizing Scientific Workflows Using Semantic Annotations” Submitted to IEEE Big Data'13 October 2013

[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)