ROHub — A Digital Library of Research Objects Supporting Scientists Towards Reproducible Science

Raúl Palma[1], Oscar Corcho[2], José Manuel Gómez-Pérez[3], and Cezary Mazurek[1]

[1] Poznan Supercomputing and Networking Center, Poznań, Poland,
rpalma@man.poznan.pl,mazurek@man.poznan.pl
[2] Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain,
ocorcho@fi.upm.es
[3] iSOCO, Madrid, Spain,
jmgomez@isoco.com

Abstract. Research Objects (ROs) are semantic aggregations of related scientific resources, their annotations and research context. They are meant to help scientists to incorporate and refer to all the research materials that they are working with in the course of an investigation. ROHub is a digital library system for ROs that supports their storage, lifecycle management and preservation. It provides a Web interface and a set of RESTful APIs. ROHub enables the sharing of scientific findings via ROs and includes features that help scientists throughout the research lifecycle to create and maintain high-quality ROs that can be interpreted and reproduced in the future. For instance, during the RO creation, scientists can assess and visualise the conformance of the RO to a set of predefined requirements. Scientists can also create at any point in time RO Snapshots. Snapshots may be useful to release the current version of research outcomes, submit it to be peer reviewed or published, share it with supervisors or collaborators, or for acknowledgement and citation purposes. ROHub can also generate nested ROs for workflow runs, exposing their full content and annotations, and includes monitoring features, such as fixity checking and RO quality, which generate notifications when changes are detected.

1 Introduction

Scientific publications have played a key role in the expansion of the vast body of human knowledge. From the traditional paper-based to the more recent digital publication, they have proved to be an effective dissemination channel of scientific findings. However, as stated in [8], a key idea that underpins science is *trust, but verify*. Verification is a step necessary to ensure the quality of the published results, and in order to do so, scientists should be able to reproduce the experiments.

As research is increasingly being conducted in digital environments new types of content and artefacts are emerging [11], including computational methods like scientific workflows that encapsulate the research processes used to manipulate data and produce results in experimental science. Consequently, publication of results may not be enough to reproduce them. We need also the data used and produced, the methods employed, and the research context in which these artefacts were conceived. Similarly, in order

to enable the reusability of scientific results, they should be published along with the set of all the resources associated to the enclosing investigation, including the research context and descriptions about the usage and provenance of these resources.

Research objects provide such container. They are aggregating objects that bundle together experimental resources that are essential to a computational scientific study or investigation, along with annotations on the bundle or the resources needed for the understanding and interpretation of the scientific outcomes. The resources aggregated can include the data used or results produced in an experiment study, the (computational) methods employed to produce and analyse that data, and the people involved in the investigation. The annotations can include provenance and evolution information, descriptions of the computational methods, dependency information and settings about the experiment executions. The RO model, introduced in [9], provides the means for capturing and describing such objects, their provenance and lifecycle, facilitating the reusability, reproducibility and better understanding of scientific experiments. The model consists of the core ro ontology, which provides the basic structure for the description of aggregated resources and annotations on those resources, and extensions for describing evolution aspects and experiments involving scientific workflows.

Hence, research objects can help scientists in sharing research outcomes that are more reusable and reproducible. However, scientists will need the appropriate technological support that assists them in creating research objects that satisfy certain requirements so that they can be easily understood, validated, reused and extended, thereby enhancing the quality of scientific production.

In this paper we present ROHub digital library system. ROHub enables the sharing of research outcomes via research objects and includes features that help scientists throughout the research lifecycle to create and maintain research objects satisfying predefined requirements so that they can be interpreted and reproduced in the future, to collaborate along this process, to publish and search these objects, and to monitor and preserve them to ensure that they will remain accessible and reproducible.

2 ROHub

ROHub is a digital library system for research objects that supports their storage, lifecycle management and preservation. It provides a Web interface and a set of RESTful APIs defining resources and representations according to the RO model, which expose a number of functionalities and possibilities for extension.

2.1 The interfaces

ROHub provides a set of REST APIs, being the two primary ones the RO API [5] and the RO Evolution API [6]. The RO API defines the formats and links used to create and maintain ROs in the digital library. It is aligned with the RO model, hence recognising concepts such as aggregations, annotations and folders. The RO model ontology [10] is used to specify relations between different resources. ROHub supports content negotiation for metadata, including formats like RDF/XML, Turtle and TriG. The RO Evolution API defines the formats and links used to change the lifecycle stage of a RO, to create an immutable snapshot or archive from a mutable Live RO, as well as to

retrieve their evolution provenance. The API follows the RO evolution model [9]. Additionally, ROHub provides a SPARQL endpoint, a Notification API [4], a Solr REST API, and a User Management API [7]. ROHub also provides a Web interface, which exposes all functionalities to the users. This is the main interface for scientists and researchers to interact with ROHub. A running instance of ROHub is accesible from `http://www.rohub.org/portal/`.

2.2 The implementation

Internally, ROHub[4] has a modular structure that comprises access components, longterm preservation components and the controller that manages the flow of data (see Fig. 1). ROs are stored in the access repository once created, and periodically the new and/or modified ROs are pushed to the longterm preservation repository.

The access components are the storage backend and the semantic metadata triplestore. The storage backend can be based on dLibra [3], which provides file storage and retrieval functionalities, including file versioning and consistency checking. It has a built-in text search engine and allows organising stored objects into hierarchical structures and associating metadata at the aggregation level. Alternatively, storage backend can use a built-in module for storing ROs directly in the filesystem.

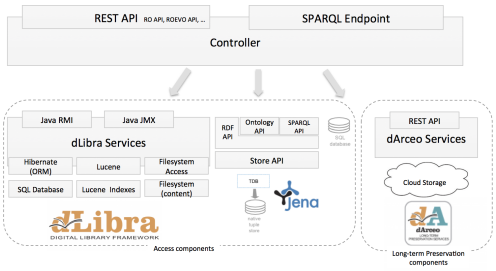

Fig. 1. ROHub internal component diagram

The semantic metadata are additionally parsed and stored in a triplestore backed by Jena TDB [1]. Jena TDB provides good support for transactions, querying, cacheing and using named graphs. The use of a triplestore offers a standard query mechanism for clients. It also provides a flexible mechanism for storing metadata about any component of a RO that is identifiable via a URI.

The longterm preservation component is built on dArceo [2]. dArceo stores ROs, including resources and annotations, and monitors their quality, alerting administrators if necessary. Standard monitoring activities include file format decay alerts and fixity checking but ROHub also monitors the RO quality through time against a predefined set of requirements. If a change is detected, notifications are generated as Atom feeds according to the Notification API.

2.3 Functionalities supporting scientists towards reproducible science

Create, manage and share ROs There are different methods for creating ROs in ROHub. They can be created from (i) scratch, adding resources progressively; (ii) by importing pack of resources from other systems (currently myExperiment); and (iii) from a ZIP file aggregating files and folders. Resources can be added and annotated from

[4] Source code available at: `https://github.com/wf4ever/rodl` and `https://github.com/wf4ever/portal`

the content panel, which also shows the folder structure. Also, as in daily practice it is sometimes common to work with local resources (scripts, data, etc.), users can use the RO-Manager tool (`https://github.com/wf4ever/ro-manager`) to create local ROs and push/pull them to/from ROHub. ROHub also provides different access modes to share the ROs: open, public or private. In open mode, anyone with an account can visualise and edit the RO. In public mode, everyone can visualise the RO, but only users with correct permissions can edit it. In private mode, only users with correct permissions can visualise and/or edit the RO. ROHub provides a faceted search interface, in addition to a simple keyword search mode, to find stored ROs. Additionally, scientists and other applications can use the provided SPARQL endpoint to query the RO metadata.

Assessing RO quality During the RO creation, users can visualise a progress bar on the RO overview panel (see Fig. 2), which shows its quality based on set of predefined basic RO requirements. When clicked, a preview of the requirements and their compliance is displayed. Users can also get more information about the quality of the RO from the Quality panel, where they can choose the template used to evaluate the RO. There are a number of templates available, each specifying particular set of RO requirements for certain purpose or task. Users can create their own templates for their particular domain or standards. Internally, this feature calls a Restful service for the evaluation.

Managing RO evolution At any point in time, users may want to create a snapshot (or release) of the current state of their RO for sharing the current outcomes with colleagues, get feedback, send it to review, or to cite them. Also, when the research has concluded, they would like to release and preserve the outcomes for future references. In ROHub this can be easily done from the RO overview panel. ROHub keeps the versioning history of these snapshots, and even calculates the changes between one snapshot and the subsequent one. Users can visualise the evolution of the RO from the History panel, where they can see a diagram with nodes the live RO and its associated snapshots/archives, and arrows showing the versioning relations between the latter. Users can click on the nodes to navigate the RO history.

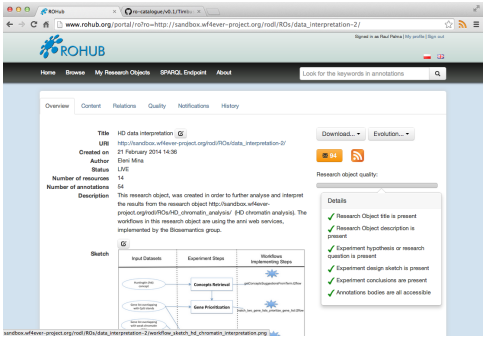

Fig. 2. ROHub - research object overview panel

Navigation of workflow run Scientists can aggregate any type of resource, including links to external resources and RO bundles, which are structured ZIP files representing self-contained ROs that facilitate their transfer and integration with 3rd party tools. Taverna, for example, can export provenance of workflow runs as RO Bundles. In ROHub, these bundles are unpacked into a nested RO, exposing its full content and annotations. Hence, scientists are able to navigate through the inputs, outputs and intermediate values of the run, something potentially useful for future reproducibility.

Monitoring ROs ROHub includes monitoring features, such as fixity checking and RO quality, which generate notifications when changes are detected. This can help to

detect and prevent, for instance, workflow decay, occurring when an external resource or service used by a workflow becomes unavailable or behaves differently. Users can visualise changes in the RO regarding content and quality monitoring in the notification panel and they can subscribe to the atom feed to get automatic notifications.

3 Conclusions

We have introduced in this paper ROHub, a digital library enhanced with semantic technologies that assists users in the generation and publication of research outcomes which are more reusable and reproducible, thus facilitating the assessment of the results quality. To this end, ROHub implements a set of APIs and a web interface for management of ROs, exposing functionalities according the RO model and related ontologies. Currently, the running instance of ROHub stores more than 1150 research objects. The future plans of ROHub include an enhanced interface of the evolution history that shows also the changes between different versions, customisation of snapshotting process, interface to facilitate the generation of quality templates, improved notification and access control panel, as well as improvements in the performance of RO loading.

4 Acknowledgements

This work was supported by Wf4Ever EU project (http://www.wf4ever-project.org, FP7-270129). We acknowledge Piotr Hołubowicz for his contribution to ROHub, Graham Klyne and Stian Soiland-Reyes for their inputs to APIs, and Wf4Ever users for their feedback, mainly Julián Garrido, Enrique Ruiz, Eleni Mina and Kristina Hettne.

References

1. Apache Jena. http://jena.apache.org/, [Online; accessed 20-Mar-2014]
2. dArceo. http://dlab.psnc.pl/darceo/, [Online; accessed 20-Mar-2014]
3. dLibra. http://dlab.psnc.pl/dlibra/, [Online; accessed 20-Mar-2014]
4. Notification API. http://www.wf4ever-project.org/wiki/display/docs/Notification+API, [Online; accessed 20-Mar-2014]
5. RO API. http://www.wf4ever-project.org/wiki/display/docs/RO+API+6, [Online; accessed 20-Mar-2014]
6. RO Evolution API. http://www.wf4ever-project.org/wiki/display/docs/RO+evolution+API, [Online; accessed 20-Mar-2014]
7. User Management API. http://www.wf4ever-project.org/wiki/display/docs/User+Management+2, [Online; accessed 20-Mar-2014]
8. How science goes wrong. http://econ.st/1hYoAaN (Oct 2013), the Economist Newspaper Limited [Online; accessed 20-Mar-2014]
9. Belhajjame, K., Corcho, O., Garijo, D., Zhao, J., Missier, P., Newman, D., Palma, R., Bechhofer, S., García-Cuesta, E., Gómez-Pérez, J.M., Klyne, G., Page, K., Roos, M., Ruiz, J.E., Soiland-Reyes, S., Verdes-Montenegro, L., De Roure, D., Goble, C.A.: Workflow-centric research objects: First class citizens in scholarly discourse. In: ESWC2012 Workshop on Semantic Publication (SePublica2012) (2012)
10. Belhajjame, K., Klyne, G., Garijo, D., Corcho, O., García Cuesta, E., Palma, R.: Wf4ever Research Object Model. http://wf4ever.github.io/ro/ (Nov 2013)
11. De Roure, D., Belhajjame, K., Missier, P., Gómez-Prez, J.M., Palma, R., Ruiz, J.E., Hettne, K., Roos, M., Klyne, G., Goble, C.: Towards the preservation of scientific workflows. In: 8th International Conference on Preservation of Digital Objects (iPRES 2011) (Nov 2011)