

Research shared: www.researchobject.org

Matthew Gamble¹, [Norman Morrison](#)¹, Stian Soiland-Reyes¹, Carole Goble¹,

¹ School of Computer Science, University of Manchester, Kilburn Building, Oxford Road, Manchester M13 9PL. Email: gamble@cs.manchester.ac.uk

Project Website: <http://www.researchobject.org/>

Source Code:

<https://github.com/apache/incubator-taverna-language/tree/master/taverna-robundle>

License: Apache v2

<https://github.com/myGrid/ruby-ro-bundle> **License:** BSD

<https://github.com/gambl/ro-python> **License:** MIT

researchobjects.org is a community project that has developed an approach to describe and package up all resources used as part of an investigation as Research Objects (RO's).

RO's - provide two main features; a manifest - a consistent way to provide a well-typed, structured description of the resources used in an investigation; and a 'bundle' - a mechanism for packaging up manifests with resources as a single, publishable unit.

RO's therefore carry the research context of an experiment - data, software, standard operating procedures (SOPs), models etc - and gather together the components of an experiment so that they are findable, accessible, interoperable and reproducible (FAIR). RO's combine software and data into an aggregative data structure consisting of well described reconstructable parts.

RO's have the potential to address a number of challenges pertinent to open research including: a) supporting interoperability between infrastructures by using ROs as a primary mechanism for exchange and publication b) supporting the evolution of research objects as a living collection, enabling provenance tracking c) providing the ability to pivot research object components (data, software, models) that are not restricted to the traditional publication.

Here we present work towards the development and adoption of ROs:

- (i) A series of specifications and conventions, using community standards, for the RO manifest and RO bundles.
- (ii) Implementations of Java, Python and Ruby APIs and tooling against those specifications;
- (iii) Examples of representations of the RO models in various languages (e.g. JSON-LD, RDF, HTML).