Using ontologies of software : example of R functions management

Emilie Gennari, Pascal Neveu, Caroline Domerg, Juliette Fabre, Vincent Negre, Anne Tireau, Olivier Corby, Catherine Faron Zucker, Isabelle Mirbel

HAL Id: hal-02754347

https://hal.inrae.fr/hal-02754347v2

Submitted on 6 Aug 2020

Using Ontologies of Software, Example of R Functions Management (Abstract)

Pascal Neveu
INRA, UMR 729 MISTEA
F-34060 Montpellier, France
Pascal.Neveu@supagro.inra.fr

Caroline Domerg
INRA, UMR 759 LEPSE
F-34060 Montpellier, France

Juliette Fabre
INRA, UMR 759 LEPSE
F-34060 Montpellier, France

Vincent Nègre
INRA, UMR 759 LEPSE
F-34060 Montpellier, France

Anne Tireau
INRA, UMR 729 MISTEA
F-34060 Montpellier, France

Olivier Corby
INRIA Sophia Antipolis
F-06902 Sophia Antipolis, France

Catherine Faron Zucker
Université Nice-Sophia Antipolis, I3S, UMR 6070
F-06903 Sophia Antipolis, France

Isabelle Mirbel
Université Nice-Sophia Antipolis, I3S, UMR 6070
F-06903 Sophia Antipolis, France

Promote, sustain and make available scientific resources, as computer programs, for multidisciplinary research teams is often a real difficulty. We propose an ontology-based approach to manage, share and promote software programs in a research community. Our proposition takes into account relations between programs: how they are linked, how they could collaborate, follow on and be retrieved. We have developed a new kind of software repository for a team of biologists, statisticians, agronomists and geneticians.

In the research laboratory LEPSE specialized on studying plant responses to environmental stresses, dozens of R[1] functions are produced every year. As a result, there is an important turn-over of function authors and users which calls for understanding, sharing and re-using these functions. In this context, we have initiated a development to organize and promote these functions through the development of a knowledge-based repository.

Given the great diversity of R functions, we have decided to index them with some formalized knowledge describing them, in order to retrieve them by formal reasoning. For this purpose, we developed an ontology providing a controlled and structured vocabulary that captures the concepts and properties necessary to describe R functions. This ontology comprises concepts and properties to describe functions like "Intention", "Argument" and the relations between functions like "hasCall" or "couldBeUsedAfter".

As a result, functions can be retrieved according to a wide range of criteria: author, graphics type, intentions, function calls –more generally, it is relevant to generate the call graph of one function to understand it–, functions from which they are adapted –this makes easier the maintenance of the repository–, functions used after or before –this helps to construct chaining of treatments–, etc..

To formalize both the ontology and the annotations of R functions, we adopt the Semantic Web models: the annotations are represented into the Resource Description Framework (RDF) and the ontology in the Ontology Web Language (OWL). As a result we are able to semantically retrieve R functions by expressing queries in the SPARQL language. We have developed a Semantic Web application for the repository, annotation and search of R functions. It relies upon the Corese engine dedicated to ontological query answering on the Semantic Web.

The architecture of our application is based on a Web Service and allows to: (i) Upload, download and update R functions or RDF annotations, (ii) Retrieve functions by processing SPARQL queries over RDF annotations. The Web Service allows to be used by heterogeneous clients for different purposes (download to R session, versioning, etc.).

We have also developed a user-friendly Web interface with dynamic pre-filled forms. Our application provides an environment for (1) *create and edit annotation*: a Web user interface allows authors to upload R functions and to describe them in a few minutes; and (2) *powerful search*: based on a SPARQL queries generator, users can find and get R functions with a global and accurate understanding and receive suggestions to support their search.

To conclude, we have built a semantic repository of annotated R functions to centralize and share R functions for biologists. It capitalizes expert know-hows that would otherwise often be lost or become non-usable because of a lack of documentation and description. We are convinced that this kind of repository developed for the LEPSE could benefit a much wider community of R function authors and users and be adapted to handle other programming languages.

---

[1] R is a software language for statistics and graphics