Use of semantically annotated resources in the Mobyle2 Web Framework

Hervé Ménager[1,#], Bertrand Néron[1], Olivia Doppelt-Azeroual[1], Olivier Sallou[2]
[1] Centre d'Informatique pour la Biologie, Institut Pasteur, Paris, France
[2] IRISA, Rennes, France
[#] Presenting author email : hmenager@pasteur.fr

Mobyle website : https://github.com/mobyle2
Mobyle license : BSD

Mobyle 2 is a project currently under development at the Institut Pasteur and the GenOuest platform. The Mobyle framework is a web-based workbench for bioinformatics analyses. Its interface allows scientists, without installing anything locally, to use command line-based bioinformatics tools to perform analyses on remote computing resources. The high level of integration between the different tools provided enables and guides users in the construction of potentially complex protocols, chaining interactively successive tasks in an exploratory mode, or automating their execution with workflows. Mobyle 2 is a major rewrite which adds new features such as collaborative work, secure data sharing, a REST API and the use of an ontology-based annotation mechanism. This presentation will focus on this last feature.

The current integration mechanisms of Mobyle are based on a custom vocabulary which is used to annotate services, biological data banks available in Mobyle, as well as the user data and workflows. These annotations, linking the actual resources to the concepts they represent, provide an integration layer used to guide the user by selecting and connecting semantically and/or syntactically compatible resources.

The upcoming version of Mobyle replaces this vocabulary with an ontology. The richness and focus of the EDAM[1] ontology allows for a more precise and consistent description of the resources that are integrated in Mobyle (command line programs, workflows, web services, interactive widgets), linked with external applications or shared through MobyleNet. Additionally, the use of this ontology as an abstraction layer for the classification of equivalent resources enables the development of a number of user-targeted features such as:

- the automatic suggestion of equivalent services, where a user can select alternative tools that perform the same task using a different method,
- the definition of abstract or semi-abstract workflows, where users can choose at runtime which specific tools will perform some or all of the tasks of a workflow,
- the automatic handling of an increased number of implicit tasks, improving the existing format detection and conversion mechanisms and adding new mechanisms such as implicit iterations.

[1] http://edamontology.org/