

Semantic Publishing - Research Life Cycle

Aleix Garrido¹, Jose Manuel Gomez-Perez¹, and Boris Villazon-Terrazas¹

iSOCO, Intelligent Software Components S.A., Av. del Partenon, 16-18, 1-7, 28042,
Madrid, Spain,
{agarrido,jmgomez,bvillazon}@isoco.com

Abstract.

1 Introduction

2 Reuse of scientific work

Reuse of scientific work, including search and exploration of relevant scientific experiments in scientific social networks (Collaboration Spheres). Wf4Ever

3 Reviewer finder

The Reviewer finder is illustrated by the APA prototype that we have developed. It consists on a web application that covers the process of finding a good reviewer for a specific article or publication. The whole process is driven by the actions performed by the user while creating different contexts towards his final objective. Each action allows the modification of the search parameters in an intuitive way and offers summaries of information that may help on understanding the available resources.

The user interface has been designed in order to keep a minimalist layout that makes user experience smooth and simple. This simplicity does not waive the content needed by the users in the aim of getting the desired results. The main part of the screen displays a set of concentric circles that serve both as a playground to customize the search and as a front of the most relevant results of the search. The center of the circles contains the article for which we are looking for reviewers. The immediate adjacent circle is the place where the user can define different contexts (each context created is equivalent to a search) by adding and removing authors and articles. The two external circles collect the most relevant reviewers that the system has found in order to tackle the main article together with the context that surrounds it. Other aspect that is covered by the Collaboration Spheres is the usage of a code of colors and warning icons when showing the results on the circles. The colors follow the traffic-light metaphor, where green represents the best reviewers, yellow is used for other good reviewers and red is used for the less (but still) recommended reviewers. As we evaluate possible conflicts between the reviewers and the article to be reviewed, we are able

to warn the users by adding a warning icon for those recommendations where conflict has been detected (author, previous co-author or same organization are examples of conflicts in our scenario). On the right of the screen there are two different columns that gather three distinct lists each. The first column focuses on authors at different levels: the first level shows the main authors of the article, the second level shows other authors that had collaborated with the main authors in previous publications, and the third level presents a set of relevant authors whose work share one or more topics with the main article. A parallel approach has been taken for the second column which focuses on articles at three different levels: the first level shows other articles authored by the main authors, the second level shows articles authored by previous co-authors, and the third level presents a set of relevant articles that have related topics with the main article. Every element from the lists is draggable and can be dropped at the circles. The drag-and-drop action allows the user to create their customized context in the Collaboration Spheres. Every time that a new element is added to the context, a cloud of tags that is placed under the circles gets updated. This cloud of tags represents the key topics for the created contexts and provides an informative abstraction of the search at a simple glance. Some of the elements at the cloud of tags are a link to their corresponding URIs at Wikipedia. Apart from that, it is worth to mention that every element is clickable in order to get a summary of its contents, together with the link to the resource itself in the VIVO platform enhancing the users exploration. A twitter search for the most relevant topics of every element is provided with the intention of adding a nice feature of pointing topic appearance in social networks.

The architecture and implementation of the system can be easily divided into three independent modules. Each module provides the needed information to the others creating a complete system. The first module is the data module. The base data used in our scenario is provided by the VIVO APA platform. It consists on an RDF that holds authors and publications that belong to the Psychology community, together with additional information for each element and the relations between them. One of our main tasks related to this dataset has been an information enrichment. Topics from the abstracts and titles have been extracted through NLP processing for each publication. One of the tools used to perform this task has been TextRazor through its API. TextRazor provides topic extraction and additional wikipedia uris for different categories. The second tool that has been used is KT (Knowledge Tagger) working together with a Thesaurus for the psychology domain provided by the APA. Calculations post-processing are used in order to add a weight value for each topic in every article. This topic weights together with the knowledge of the authors that provides the dataset allows the assessment of expertise values for each author. The new data created is semantically annotated and added to the initial dataset that is uploaded and stored in Virtuoso Triplestore. The Triplestore is accessible via HTTP queries that are posted by the next module: the Web Server. The Web Server is in charge of communicating the data from the Data Module to the

final user. It is a Java project that triggers different SPARQL queries by using JENA under user request. The results of the queries are processed and returned via REST-JSON web services that respond to the front end. Some filtering of the data and improvements are done in the Java project, but they are kept as minimum as possible. In this way we try to reduce the impact of this module and use it as a communication channel between the other modules. The power of the web server resides on the complex SPARQL queries and the ability to handle user requests. Finally, the Front End Module is a web application that is directly available for the users. It has been developed in HTML5, using also javascript, jQuery and CSS. The interface reacts to user actions and throws different Ajax queries to the Web Server REST Services. Once the JSON response is obtained it is parsed in order to show the results to the users and update the interface as expected. The final result offers a smooth experience that retrieves data through recommendations under user demand.

4 Evaluation

Validation and evaluation of the experiment results and materials (Completeness, Stability, Reliability).

5 Conclusions and Future Work

Acknowledgments

This work was supported by SmartContent project.