Mobile Facetted Browsing LODD Applications for Supporting Clinicians

Daniel Sonntag, Jochen Setz and Maha Ahmed-Baker

German Research Center for AI (DFKI)
Stuhlsatzenhausweg 3, 66123 Saarbruecken, Germany

Abstract. An LODD application can be used to improve a clinician's daily work. Specific requirements of the radiology domain let us aggregate RDF results from several LODD sources such as DrugBank, Diseasome, and LinkedCT, as well as links to the web pages of ClinicalTrials and Pubmed, in a facetted browsing based mobile graphical user interface.

1   Introduction

Linked Data has the potential to provide easy access to related data sets in the healthcare domain. This should allow medical experts to explore the interrelated data more conveniently and efficiently without having to switch between many different applications, data sources, and user interfaces. In discussions and usability studies with clinicians we found that there are basically three different scenarios where the medical information contained in Linked Data can be useful: (1) the clinical reporting process; (2) the patient follow-up treatment (i.e., monitoring the patient's health condition and the development of the disease); and (3) the clinical disease staging and patient management. We will explain how an LODD (http://www.w3.org/wiki/HCLSIG/LODD) application based on diseases, drugs, and clinical trials can be used in all three scenarios. Such an application is based on medical disease knowledge as input and provides an incremental graphical HCI with a facetted browsing functionality to get relevant trial and drug information related to the disease information in the patient's record.

2   Facetted Browsing Application

In order to implement the facetted browsing LODD application, we connected RadLex, www.radlex.org/, our initial domain reference ontology for anatomy and radiology, with LODD data, in particular with LinkedCT to get further access to DrugBank and Diseasome (and the ClinicalTrials and Pubmed webpages, respectively). Figure 1 shows the GUI of the implemented facetted browsing tool. There are two options for an input, either a RadLex term that a drop-down menu suggests while the user types the disease name, or the disease name automatically taken from the RadLex image annotation. The backend runs a query

written in SPARQL to interlink LinkedCT, Diseasome, DrugBank, and Daily-Med to gain useful information that might help the radiologist to make a decision about a treatment or in order to suggest alternative medications. Mappings between Radlex and LinkedCT do not exist; we obtained best results when using string-based ontology matching approaches while mapping the Radlex term (verified by the search-as-you-type functionality with access to a Radlex API) onto individual tokens (words) in the trial description which we indexed for all 90K keywords to get the trial URIs (cf. http://linkedct.org/resource/keyword/). Links to Diseasesome and DrugBank are largely available but also in a n to n relationship. Our tool allows a user to filter thousands of results according to the bast-ranked string matches and linked data relations in a convenient way. Clinicians can retrieve the trails they might be interested in in just a few seconds.
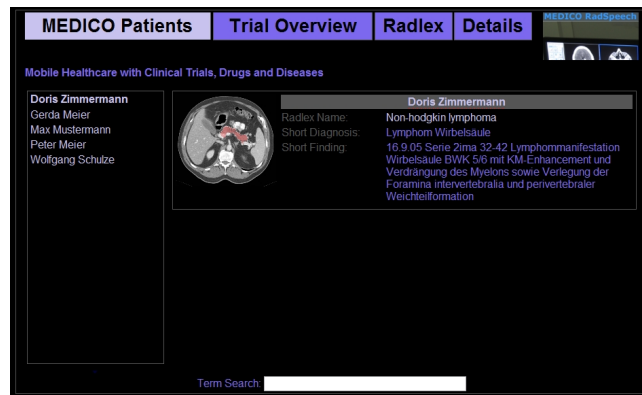

Fig. 1. LODD Application (iPad), http://digitaleveredelung.dfki.de/MEDICO/

For example, a radiologist types a disease name "carcinoma". Then he or she is probably interested in the clinical trails of a specific study type, drug, and/or drug type which we aggregated into different facets according to the disease mapping function or the provided links to Diseasome, DrugBank, and DailyMed. For genetically caused diseases, one can also look at the disease from the genetic point of view (at least the gene name can help to combine or compare therapies). Most of the development time has been spent on (1) the access to non-conformed Linked Data RDF results in the form of conversion programmes for JSON and XML results; (2) the provision of the online SPARQL queries for our Virtuoso server (http://virtuoso.openlinksw.com/); and (3) the implementation of a graphical user interface from scratch by using open-source knowledge management tools such as Exhibit, see http://www.simile-widgets.org/exhibit/.