Reviewer's report

Title: AGUIA - Autonomous Graphic User Interface Assembly - for clinical trial semantic data services

Version: 2  Date: 1 July 2010

Reviewer: Prakash Nadkarni

Reviewer's report:

Major Compulsory Revisions
-----------------------------

The paper has improved marginally, but many unanswered questions remain.

The use-case for RDF appears to be that, in situations where database schemas are highly volatile, RDF provides the flexibility of entity-attribute-value (EAV) modeling, where one can simulate virtual schemas in metadata. (RDF is a kind of EAV data store.) In the authors’ scenario, the metadata is used to capture not only the virtual schema but also information about how data is to be presented.

The use of metadata to capture presentation information is not novel: frameworks such as Ruby on Rails (which also support a REST approach) have been doing this for a while: the only difference is that they configuration files rather than RDF stores to persist their metadata. In the reviewer’s own experience, because the metadata schema itself is not volatile, a 3NF relational sub-schema can also be used to capture presentation metadata.

What would make the authors’ paper viable (or not) is not the fact that they use RDF per se, but how they allow a user (who is possibly a non-programmer) to specify various aspects of the desired interface, both at the level of the “class” of data – e.g., chemotherapy, demographics, visits, histology – as well as individual fields within a class – e.g., gender, date of birth, tumor grade. Also, since classes of data are inter-related – e.g., one patient has multiple visits – one typically wants to see summary information of the “many” data – e.g., visits – when looking at a form containing a record of the “one” data – e.g., demographics of a single patient.

This information is simply not present in the paper. Even then, I would have given it a pass if, after inspecting the sample Website (lbl.mdanderson.org/edu) , I could see a unifying coherence in the user interface. Unfortunately, after trying out the sample Website, I am convinced more than ever that this interface has not been used for extended periods (if it has been used at all in production) by anyone other than the paper’s authors.

• The site was clearly test data – patients with the name PatientX, and surgeons with the name Surgeon2. Of course, I don’t expect to see real data at a site (HIPAA considerations), but the data should at least be realistic. While I was able
to look at the details of the Histology class (which has the fields Adenocarcinoma subtype, tumor grade, etc.), I could not even look at the details of a single histology record – all I got appeared to be an audit trail of changes made to this class by individual users. Several classes have zero records.

- There is a significant discrepancy between the video and the site. While the video demonstrates the ability to add a record through a forms-based interface, I searched in vain for a means of doing that on the site. (There was a means of uploading tab-delimited files, hardly an effective means of providing interactive data validation, which all of the existing GUI frameworks provide.) Indeed, it is not clear how robust the data-validation component of the framework is. If a video differs from a site, I have to assume that it is the site that represents reality.

- In their response to the reviewer’s complaint about the seemingly random order of folders (each folder corresponds to a class of data), the authors stated, “The folder tree now can be sorted by the user through of insertion of a number after the name of collection. If the user doesn’t set any order, the folders will be ordered alphabetically.” I could not do this – presumably some one with administrator privileges could. In any case, re-ordering folders by making a permanent change to the class name is one of the most counter-intuitive (and destructive) ways of reordering data. The logical means of ordering data (which would allow customization by user – the pathologists would want to access different information rapidly than the radiologists – would be to maintain ordering metadata that recorded the preferred order of display. The fact that the authors haven’t considered this leaves me wondering how sophisticated their display rules are. (Individual fields within a class are also required to be prefixed with numbers to force presentation order – thus, the field “tumor grade” is really “03_tumor_grade”. Can’t the authors separate the intended presentation order from the field’s name itself?

Needless to say, there seems to be no means of viewing data from more than one class simultaneously - e.g., demographics in a form, and summary histology information in a sub-form/table. Most rival frameworks have this capability.

- Since the display rules were not described anywhere in the paper, I tried inspecting the GUI_Rules folder on this site. I got back an enormous list of rules, but was unable to inspect the details of even one of them – other than the columns ID, “Notes”, Created On, Created By. The essential information – what is the rule’s definition? – was unavailable. How is a power user who is not an RDF guru to go about the task of specifying aspects of the user interface?

- There was a “SPARQL” tab. Is the typical bench scientist or clinician expected to know SPARQL to query the system? I would have expected a query-by-example type interface that generates SPARQL behind the scenes.
In summary, this interface fails to meet the most basic usability requirements - the ability to list data in summary and in detail (one record at a time) and allow its interactive editing with validation.

I therefore have to judge the paper, not by the authors’ good intentions (including the provision of their effort as open-source), or the fact that they employ RDF or REST principles, but by the end-results, in terms of a convincing demonstration of what the framework can actually do. Even if they made the case (which they do not) that their framework was mature, they would need to contrast its strengths and limitations with other perfectly good, existing alternatives. (Many of these, such as Ruby on Rails, are also open-source.)

**Level of interest:** An article of limited interest

**Quality of written English:** Not suitable for publication unless extensively edited

**Statistical review:** No, the manuscript does not need to be seen by a statistician.

**Declaration of competing interests:**

'I declare that I have no competing interests'