

Review of: "CRAFTS: Configurable REST APIs For Triple Stores"

Silvio Peroni¹

¹ University of Bologna

Potential competing interests: The author(s) declared that no potential competing interests exist.

Metadata

Title: CRAFTS: Configurable REST APIs For Triple Stores

Author: Guillermo Vega-Gorgojo

Submitted to: Semantic Web

Preprint: <http://www.semantic-web-journal.net/content/crafts-configurable-rest-apis-triple-stores-0>

Review

In this article, the author presents a specific tool, i.e. CRAFTS, that enables interfacing with one or more SPARQL endpoints to provide REST APIs that web developers can use to interact with RDF data. The article presents a detailed description of the tool and its main characteristics (Section 2), accompanied by some examples of the use of the tool to produce REST APIs. The final sections are dedicated to showing some statistics about the use of CRAFTS and discussing some related works.

Generally speaking, the release of these kinds of tools is very welcomed in the Semantic Web community since such tools enable broad adoption of Semantic Web technologies for consuming and storing data hidden behind common Web-based standards and practices such as Web REST APIs. Thus, it is essential to present these tools in articles to share their existence with a broad audience.

However, while the description of the tool and its use has been addressed appropriately in the paper, not all the requirements of this particular kind of paper (i.e. reports on tools and systems) have been appropriately addressed. In particular, the quality and impact dimensions (see <http://www.semantic-web-journal.net/authors#types>) must be supported with appropriate evidence.

Quality dimension

While the description of the tool's features is well introduced and the overall organisation of its main components, the article lacks an appropriate description of implementation details. A UML diagram presenting the various classes and methods comprising the tool would enable a programmer to understand

the rationale of the software organisation and the relationships between the various classes.

In addition, no evidence shows how to set up a CRAFTS instance with a particular API in the article. How can I run such an instance on my machine? Can I integrate CRAFTS in my Javascript code by creating an object of a particular class and then using that object to communicate with the SPARQL endpoint? Which level of customizability is available in terms of extending CRAFTS behaviour, formats supported, etc.? How can I customise it? I know that, probably, part of the answers to the question above is included in the documentation, but it would be worth having an overview of these aspects in the paper.

Another important aspect related to this dimension is the missing analysis of the other tools available in the community developed to address the same task, such as RAMOSE (full disclosure: I am one of its developers), OBA, grlc, and BASIL. There is a claim in the introduction that states that these tools cannot comply with some crucial requirements such as reading and writing LOD data from multiple triple stores using JSON as exchange format – while, implicitly, only CRAFTS appears to be able to comply with these requirements. However, the paper has not demonstrated this since there is no comparison between CRAFTS and the other tools, at least in terms of features they provide, what they miss, and how they handle things. For example, in ^[1], there is a comparison table at the end that presents some dimensions for comparison. It would be good to start from there, extend the dimensions for comparing the tools, and develop a new extended and revised table with more information that highlights the added value provided by CRAFTS.

There is another aspect that is worthy of note, which is that of limitations. Usually, no tool can address a set of requirements in the best possible way. Thus, it is also essential to present the limits of a tool, particularly when compared with others.

Last but not least, CRAFTS is a software and, as such, needs to be appropriately cited. Since GitHub is used to make the code available, it would be appropriate to link the repository with Zenodo to have a citable DOI for each release of the tool. More information is available at <https://guides.github.com/activities/citable-code/>. Once with the DOI, the CRAFTS version presented in the paper should be appropriately cited in the text, and a bibliographic reference should be included in the reference list.

All these things enable a reader to assess (either directly or indirectly) the quality of the tool.

Impact dimension

According to the reviewing guidelines of the Semantic Web Journal, the impact of a tool can be measured according to one of the following three categories:

1. impact beyond the author's range of influence (e.g. uptake of the work by other research groups);
2. impact within the author's range of influence (e.g. use of the results in a collaborative research project);
3. potential impact (e.g. the work if a follow-up of a previously impactful contribution).

According to the paper, CRAFTS lies in category 2) since all the projects presented in section 4 involve the tool's author. In this case, the author has to make, in the paper, a compelling case that impact according to 1) will be had very soon starting from the evidence collected according to 2).

The evidence presented in the article is not enough for claiming reaching impact according to a). Indeed, it includes only a few quantitative statistics that prevent a reader from having a clear picture of the use of such tool or of the APIs produced by using the tool against the regular use of a SPARQL endpoint for addressing the same tasks of reading and writing LOD data.

Playing Devil's advocate for a while, all the projects introduced in section 4 provide a Web interface that enables users who are not experts of the underlying technology to interact with the data shown by the Web interface. Thus, the complexity of the underlying model (being either pure RDF data queryable via SPARQL or just simple Web REST APIs) is already hidden by the Web interface since a typical Web user would not be able to understand how to use such technologies otherwise. Considering this, what have been the benefits provided by CRAFTS against SPARQL? Why using CRAFTS in these contexts is better than using SPARQL (which is undoubtedly more flexible) for the users of the Web applications provided in the three projects presented in section 4? I can see that CRAFTS can be beneficial to a developer who has to build such Web applications and interfaces and is not an expert in Semantic Web technologies. However, these possible benefits for such a specific kind of user have not been measured. Therefore, it is impossible to understand if CRAFTS indeed is an added value in the process (e.g. it simplifies the development of Web applications interacting with LOD data) or it is just a tool of the chain of software that provides only an alternative way of reading/writing data that are otherwise accessible via SPARQL.

In addition, the data in Table 7 are not sufficient to understand the tool's impact. First, they are aggregated data, and it is not clear when they have been collected. It would be more effective to show them in a diagram for each project presenting the evolution of such numbers (i.e. craftsmen, LOD users, etc.) in time, month by month. This will allow one to observe if there is any trend in using the REST API created via CRAFTS.

Also, the labels used in Table 7 should be better explained. What do you mean with craftsmen, LOD users, sessions, etc.? In particular, LOD users is the most problematic to me since it recalls that the users involved are expert users of LOD data. Why should these users need to use such proxy REST APIs provided by CRAFTS instead of interacting directly with the SPARQL endpoints?

Conclusion

While the tool is of interest to the Semantic Web community, the article submitted does not provide the necessary evidence to claim the quality and impact of the tool. Therefore, I think it is necessary to essentially revise the article, even by collecting additional evidence, to have enough material to deserve publication in the Semantic Web Journal.

References

1. [^] *Marilena Daquino, Ivan Heibi, Silvio Peroni, David Shotton. (in print). [Creating RESTful APIs over SPARQL endpoints using RAMOSE](#). Semantic Web.*