

articles in HTML by means of the RASH format⁹. RASH is a markup language for writing research articles that restricts the use of HTML elements to only 31 elements. It is possible to include also RDF statements as RDFa annotations and/or as Turtle, JSON-LD and RDF/XML triples by using the tag “script”. In addition, RASH strictly follows the Digital Publishing WAI-ARIA Module 1.0 [5] for expressing structural semantics on various markup elements used. RASH is accompanied by a series of tools for helping the creation/conversion/enrichment of such documents from/into different formats. In addition to the converters, which will be introduced in the following section, there are:

- a script to enable users to check RASH documents against the specific requirements in the RASH RelaxNG grammar and HTML;
- a visualisation tool for showing RASH documents on browsers with two different layouts (Web-based and Springer LNCS – the latter is partially based on *Linked Research* [1] CCSS);
- a suite that automatically annotates RASH elements with their actual (structural) semantics according to the *Document Components Ontology (DoCO)* [3].

All the converters and tools are all available on the RASH Framework repository on GitHub. It is worth mentioning that RASH was already proposed as one of the possible formats for HTML submissions in several academic events¹⁰.

3. RASH ONLINE CONVERSION SERVICE

We created an online conversion tool called *ROCS (RASH Online Conversion Service)* for supporting authors in writing RASH documents and preparing submissions to be easily processed by current journals, workshops and conferences. The abstract architecture of the tool is shown in Fig. 1, while the tool is available at <http://dasplab.cs.unibo.it/rocs>. ROCS allows converting an ODT document, written according to specific guidelines, into RASH and, then, into LaTeX according to either the Springer LNCS or the ACM IPCS layouts. Such ODT guidelines¹¹ are very simple and use only the basic features available in OpenOffice Writer, without using any external tool or plugin.

ROCS is actually based on several other software artefacts that are included in the RASH Framework, i.e.:

- an XSLT 2.0 file to perform conversions from OpenOffice documents into RASH documents;
- a Java-based application that allows converting OpenOffice documents into RASH documents;
- XSLT 2.0 files for converting RASH documents into LaTeX according to ACM ICPS and Springer LNCS.

ROCS allows users to upload three kinds of file, i.e., an ODT document, a HTML file compliant with RASH, and a ZIP archive which contains an HTML file compliant with

⁹<https://rawgit.com/essepuntato/rash/master/documentation/index.html>

¹⁰<https://github.com/essepuntato/rash/#venues-that-have-adopted-rash-as-submission-format>

¹¹<https://rawgit.com/essepuntato/rash/master/documentation/rash-in-odt.odt>

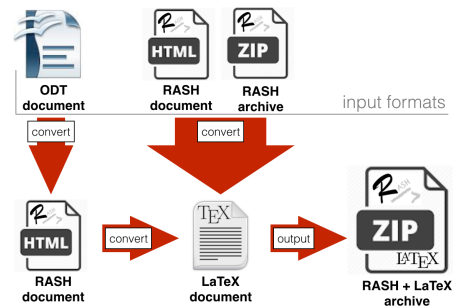


Figure 1: The architecture of ROCS.

RASH and related files (i.e., CSSs, javascript files, fonts, images). It returns a ZIP archive containing the original document plus all its converted versions, i.e., RASH, if an ODT file was given, and the LaTeX file.

The main advantage of having the paper both in RASH and in LaTeX is that it is very easy for RASH to be adopted by workshops, conferences or journals. Since, the program committee, the reviews and the editors will also have access to a LaTeX or a PDF version of the paper, the RASH file is an addition that does not preclude any current workflows. Of course, the hope is that the inherent advantages of an HTML-based format such as RASH will eventually persuade stakeholders to adopt the HTML version whenever it will be possible, keeping the alternatives as fallback options.

4. CONCLUSIONS

In this poster paper we introduced *ROCS*, the *RASH Online Conversion Service* that allows the conversion of ODT documents to RASH, an HTML-based format, and LaTeX. We are currently developing additional XSLT documents in order to convert DOCX documents into RASH and to convert RASH documents into several LaTeX and XML-based formats for scholarly communications, such as ACM journals and EPUB3 [2], and to convert LaTeX files into RASH documents as well.

5. REFERENCES

- [1] Capadisli, S., Riedl, R., Auer, S. (2015). Enabling Accessible Knowledge. In Proceedings of CeDEM 2015. <http://csarven.ca/enabling-accessible-knowledge>
- [2] Conboy, G., Garrish, M., Gylling, M., McCoy, W., Makoto, M., Weck, D. (2014). EPUB 3 Overview. Recommended Specification 26 June 2014. <http://www.idpf.org/epub/301/spec/epub-overview.html>
- [3] Constantin, A., Peroni, S., Pettifer, S., Shotton, D., Vitali, F. (2016). The Document Components Ontology (DoCO). In Semantic Web, 7 (2). <http://dx.doi.org/10.3233/SW-150177>
- [4] Di Iorio, A., Nuzzolese, A. G., Osborne, F., Peroni, S., Poggi, F., Smith, M., Vitali, F., Zhao, J. (2015). The RASH Framework: enabling HTML+RDF submissions in scholarly venues. In Proceedings of the poster and demo session of ISWC 2015. http://ceur-ws.org/Vol-1486/paper_72.pdf
- [5] Garrish, M., Siegman, T., Gylling, M., McCarron, S. (2015). Digital Publishing WAI-ARIA Module 1.0. W3C Working Draft 19 November 2015. <http://www.w3.org/TR/dpub-aria-1.0/>