The Planetary System: Executable Science, Technology, Engineering and Math Papers

Christoph Lange, Michael Kohlhase, Catalin David, Deyan Ginev, Andrea Kohlhase, Bogdan Matican, Stefan Mirea, and Vyacheslav Zholudev

Computer Science, Jacobs University Bremen, Germany
{ch.lange,m.kohlhase,c.david,d.ginev,a.kohlhase,
b.matican,s.mirea,v.zholudev}@jacobs-university.de

Abstract. Executable scientific papers contain not just layouted text for reading. They contain, or link to, machine-comprehensible representations of the scientific findings or experiments they describe. Client-side players can thus enable readers to "check, manipulate and explore the result space" [1]. We have realized executable papers in the STEM domain with the Planetary system. Semantic annotations associate the papers with a content commons holding the background ontology, the annotations are exposed as Linked Data, and a frontend player application hooks modular interactive services into the semantic annotations.

1 Application Context: STEM Document Collections

The Planetary system [2] is a semantic social environment for document collections in Science, Technology, Engineering and Mathematics (STEM). STEM documents have in common that they describe concepts using mathematical formulæ, which are composed from mathematical symbols – operators, functions, etc. –, which have again been defined as more foundational mathematical concepts in mathematical documents. Thus, there is a dynamically growing ontology of domain knowledge. The domain knowledge is structured along the following, largely independent dimensions [3,4]: (i) logical and functional structures, (ii) narrative and rhetorical document structures, (iii) information on how to present all of the former to the reader (such as the notation of mathematical symbols), (iv) application-specific structures (e.g. for physics), (v) administrative metadata, and (vi) users' discussions about artifacts of domain knowledge.

We have set up Planetary instances for the following paradigmatic document collections: (i) a browser for the ePrint arχiv [5], (ii) a reincarnation of the PlanetMath mathematical encyclopledia [6] (where the name Planetary comes from), (iii) a companion site to the general computer science (GenCS) lecture of the second author [7,8], and (iv) an atlas of theories of formal logic [9]. This list is ordered by increasing machine-comprehensibility of the representation and thus, as explained below, by increasing "executability" of the respective papers. All instances support browsing and fine-grained discussion. The PlanetMath and GenCS collections are editable, as in a wiki[1], whereas the arχiv and Logic Atlas

[1] Planetary reuses technology of our earlier semantic wiki SWiM [10].

G. Antoniou et al. (Eds.): ESWC 2011, Part II, LNCS 6644, pp. 471–475, 2011.

corpora have been imported from external sources and are presented read-only. We have prepared demos of selected services in all of these instances.

2 Key Technology: Semantics-Preserving Transformations

Documents published in PLANETARY become flexible, adaptive interfaces to a *content commons* of domain objects, context, and their relations. This is achieved by providing an integrated user experience through a set of interactions with documents based on an extensible set of client- and server side services that draw on explicit (and thus machine-understandable) representations in the content commons. We have implemented or reused ontologies for all structures of STEM knowledge ([4] gives an overview). Annotations of papers with terms from these ontologies act as hooks for local interactive services. By translation, PLANETARY makes the structural ontologies editable in the same way as the papers, so that the community can adapt and extend them to their needs.

The sources of the papers are maintained in LaTeX or the semantic mathematical markup language OMDoc [11]. For querying and information retrieval, and interlinking with external knowledge – including discussions about concepts in the papers, but also remote Linked Datasets –, we extract their semantic structural outlines to an RDF representation, which is accessible to external services via a SPARQL endpoint and as Linked Data [8]. For human-comprehensible presentation, we transform the sources to XHTML+MathML+SVG [8]. These papers gain their "executability" from embedded semantic annotations: Content MathML[2] embedded into formulæ [13], and an RDFa subgraph of the above-mentioned RDF representation embedded into XHTML and SVG.

The amount of semantic annotations depends on the source representation: (i) The arχiv corpus – 500+K scientific publications – has LaTeX sources, most of which merely make the section structure of a document machine-comprehensible, but hardly the fine-grained functional structures of mathematical formulæ, statements (definition, axiom, theorem, proof, etc.), and theories. We have transformed the papers to XHTML+MathML, preserving semantic properties like formula and document structure [5]. (ii) The PlanetMath corpus is maintained inside PLANETARY; it additionally features subject classification metadata and semi-automatically annotated concept links [14], which we preserve as RDFa. (iii) The GenCS corpus is maintained in STeX, a semantics-extended LaTeX [15], inside PLANETARY. STeX makes explicit the functional structures of formulæ, statements, and theories, narrative and rhetorical structures, information on notation, as well as – via an RDFa-like extensibility – arbitrary administrative and application-specific metadata. This structural markup is preserved as Content MathML and RDFa in the human-comprehensible output. In this translation, OMDoc, an XML language semantically equivalent to STeX, serves as an intermediate representation. (iv) The Logic Atlas is imported into PLANETARY from an external OMDoc source but otherwise treated analogously to the GenCS corpus.

[2] Or the semantically equivalent OpenMath [12].

3 Demo: Interactive Services and the Planetary API

Our demo focuses on how Planetary makes STEM papers executable – by hooking interactive services into the annotations that the semantics-preserving translations put into the human-comprehensible presentations of the papers. Services are accessible locally via a context menu for each object with (fine-grained) semantic annotations – e.g. a subterm of a formula –, or via the "`InfoBar`", as shown in fig. 1. The menu has one entry per service available in the current context; the `InfoBar` indicates the services available for the information objects in each line of the paper. In the image on the right of fig. 1, we selected a subterm and requested to fold it, i.e. to simplify its display by replacing it with an ellipsis. The `FoldingBar` on the left, similar to source code IDEs, enables folding document structures, and the `InfoBar` icons on the right indicate the availability of local discussions. Clicking them highlights all items with discussions; clicking any of them yields an icon menu as shown in the center. The icon menu for the discussion service allows for reporting problems or asking questions using a STEM-specifically extended argumentation ontology [16]. The richer semantic markup of the GenCS and Logic Atlas collections enable services that utilize logical and functional structures – reflected by a different icon menu. Fig. 2 demonstrates looking up a definition and exploring the prerequisites of a concept. The definition lookup service obtains the URI of a symbol from the annotation of a formula and queries the server for the corresponding definition. The server-side part of the prerequisite navigation service obtains the transitive closure of all dependencies of a given item and returns them as an annotated SVG graph. Computational services make mathematical formulæ truly executable: The user can send a selected expression to a computer algebra web service for evaluation or graphing [17], or have unit conversions applied to measurable quantities [18]. Finally, besides these existing services, we will demonstrate the ease of realizing additional services – within the Planetary environment or externally of it. The API for services running as scripts in client-side documents is essentially defined

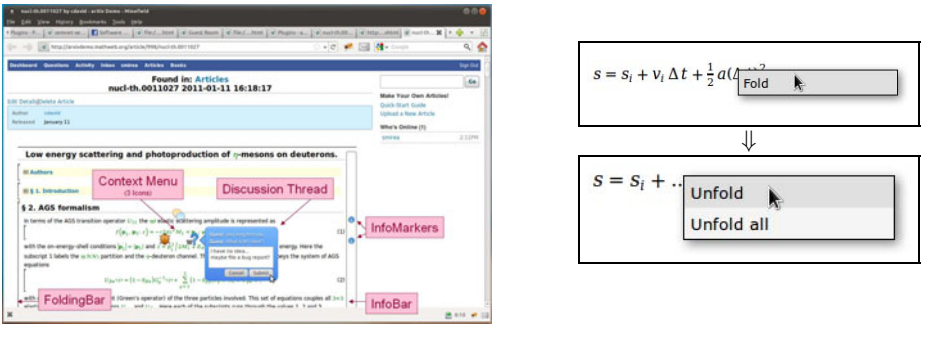

Fig. 1. Interacting with an arχiv article via `FoldingBar`, `InfoBar`, and localized discussions. On the right: localized folding inside formulæ

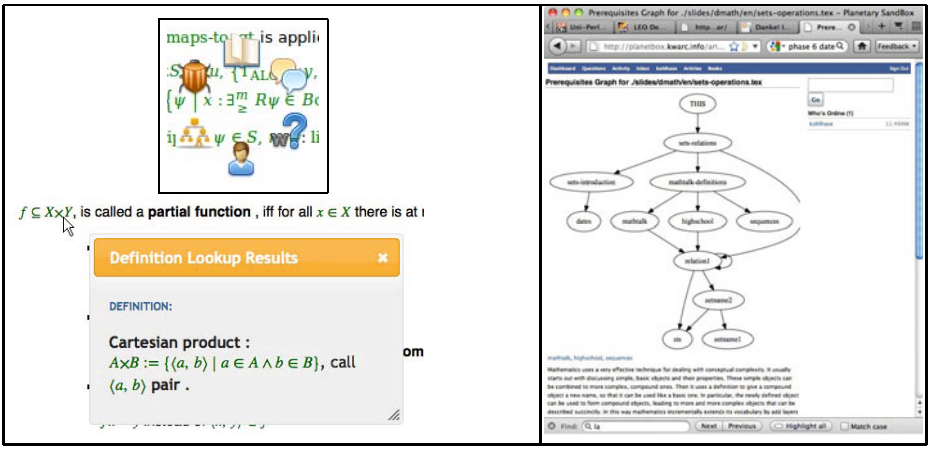

Fig. 2. Definition Lookup and Prerequisites Navigation

by the in-document annotations, the underlying structural ontologies that are retrievable from the content commons, the possibility to execute queries against the content commons, and the extensibility of the client-side user interface.

4 Related Work

Like a **semantic wiki**, PLANETARY supports editing and discussing resources. Many wikis support LaTeX formulæ, but without fine-grained semantic annotation. They can merely *render* formulæ in a human-readable way but not make them executable. The Living Document [19] environment enables users to **annotate and share life science documents** and interlink them with Web knowledge bases, turning – like PLANETARY – every single paper into a portal for exploring the underlying network. However, life science knowledge structures, e.g. proteins and genes, are relatively flat, compared to the tree-like and context-sensitive formulæ of STEM. State-of-the-art **math e-learning systems**, including ActiveMath [20] and MathDox [21], also make papers executable. However, they do not preserve the semantic structure of these papers in their human-readable output, which makes it harder for developers to embed additional services into papers.

5 Conclusion and Outlook

PLANETARY makes documents executable on top of a content commons backed by structural ontologies. Apart from mastering semantic markup – which we alleviate with dedicated editing and transformation technology – document authors, as well as authors of structural ontologies, only need expertise in their own domain. In particular, no system level programming is necessary: The semantic representations act as a high-level conceptual interface between content authors and the system and service developers. Even developers can realize considerably new services as a client-side script that runs a query against the content

commons. This separation of concerns ensures a long-term compatibility of the knowledge hosted in a Planetary instance with future demands.

References

1. Executable Paper Challenge, http://www.executablepapers.com
2. David, C., et al.: eMath 3.0: Building Blocks for a social and semantic Web for online mathematics & ELearning. In: Workshop on Mathematics and ICT (2010), http://kwarc.info/kohlhase/papers/malog10.pdf
3. Kohlhase, A., Kohlhase, M., Lange, C.: Dimensions of formality: A case study for MKM in software engineering. In: Autexier, S., Calmet, J., Delahaye, D., Ion, P.D.F., Rideau, L., Rioboo, R., Sexton, A.P. (eds.) AISC 2010. LNCS(LNAI), vol. 6167, pp. 355–369. Springer, Heidelberg (2010)
4. Lange, C.: Ontologies and Languages for Representing Mathematical Knowledge on the Semantic Web. Submitted to Semantic Web Journal, http://www.semantic-web-journal.net/underreview
5. arXMLiv Build System, http://arxivdemo.mathweb.org
6. PlanetMath Redux, http://planetmath.mathweb.org
7. Kohlhase, M., et al.: Planet GenCS, http://gencs.kwarc.info
8. David, C., Kohlhase, M., Lange, C., Rabe, F., Zhiltsov, N., Zholudev, V.: Publishing math lecture notes as linked data. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6089, pp. 370–375. Springer, Heidelberg (2010)
9. Logic Atlas and Integrator, http://logicatlas.omdoc.org
10. Lange, C.: SWiM – A semantic wiki for mathematical knowledge management. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 832–837. Springer, Heidelberg (2008)
11. Kohlhase, M.: OMDoc An open markup format for mathematical documents [Version 1.2]. LNCS (LNAI), vol. 4180. Springer, Heidelberg (2006)
12. Open Math 2.0. (2004), http://www.openmath.org/standard/om20
13. MathML 3.0., http://www.w3.org/TR/MathML3
14. Gardner, J., Krowne, A., Xiong, L.: NNexus: Towards an Automatic Linker for a Massively-Distributed Collaborative Corpus. IEEE Transactions on Knowledge and Data Engineering 21.6 (2009)
15. Kohlhase, A., Kohlhase, M., Lange, C.: sTeX – A System for Flexible Formalization of Linked Data. In: I-Semantics (2010)
16. Lange, C., et al.: Expressing Argumentative Discussions in Social Media Sites. In: Social Data on the Web Workshop at ISWC (2008)
17. David, C., Lange, C., Rabe, F.: Interactive Documents as Interfaces to Computer Algebra Systems: JOBAD and Wolfram|Alpha. In: CALCULEMUS, Emerging Trends (2010)
18. Giceva, J., Lange, C., Rabe, F.: Integrating web services into active mathematical documents. In: Carette, J., Dixon, L., Coen, C.S., Watt, S.M. (eds.) MKM 2009, Held as Part of CICM 2009. LNCS(LNAI), vol. 5625, pp. 279–293. Springer, Heidelberg (2009)
19. García, A., et al.: Semantic Web and Social Web heading towards Living Documents in the Life Sciences. In: Web Semantics 8.2–3 (2010)
20. ActiveMath, http://www.activemath.org
21. MathDox Interactive Mathematics, http://www.mathdox.org