Softw Syst Model (2016) 15:303–305
DOI 10.1007/s10270-016-0524-2

EDITORIAL

The evolution of model editors: browser- and cloud-based solutions

Jeff Gray[1] · **Bernhard Rumpe**[2]

Published online: 5 April 2016

In March 2006, Google purchased Upstartle to gain access to their browser-based word processor called Writely [1]. This acquisition from over a decade ago led to what we now know as Google Docs, which ushered in a new form of collaborative authoring tools. The idea of using a Web browser as an editing platform, coupled with the storage options available within the cloud, provides powerful new capabilities that have transformed the way we interact with colleagues to design and create documents, as well as all other sorts of artifacts. Specialized text processing solutions, like the LaTeX-focused Overleaf environment [2], bring a fresh new approach to collaboration using long-standing traditional tools. Furthermore, browser- and cloud-based authoring tools have penetrated many domains. For example, in computer science education, tools such as Scratch help new programmers learn block-based coding in a browser, where programs are stored in the cloud with a large repository (over 13.M shared Scratch programs are available at the time of this writing) of user-shared examples [3].

There are multiple benefits of combining browser-based authoring environments with a cloud service. An obvious advantage is the platform independence that can be achieved through a browser, allowing the tool implementers to focus more on the core tool features, instead of the implementation morass of reproducing the same tool functionality across different platforms (browser incompatibility issues notwithstanding). Cloud services not only allow resources and intellectual property to be accessed from anywhere, but also permit the sharing of those resources with collaborators. The benefit to the user is more flexibility in accessing the intellectual resources from multiple devices while collaborating with colleagues in real time. The first time seeing a colleague type in a shared Google Doc file at the same time as your own edits is often a watershed moment for many users. The opportunity to perform real-time updates will shorten feedback loops for constructive collaboration considerably and thus improve agile development approaches.

It is exciting to see these same types of collaborative environments emerging in the MDE tooling space. In fact, the CloudMDE workshop at MODELS (2012, 2014, and 2015) [4] has focused on ways that MDE can bring benefits to the cloud, and how the cloud can benefit MDE (this latter idea is more related to this editorial discussion). A survey of various collaborative modeling tools that are emerging can be found in [6]. Below is a sample of just a few of the commercial and research tools available that each offer some supportive nuance enabled by in-browser cloud-supported collaboration.

- GenMyModel is a commercial tool that allows collaborative modeling for UML, BPMN, RDS, and flowcharts in a "Google Drive-style collaboration" [6]. All collaborative updates are made in real time without the need for manual lock–unlock coordination. Furthermore, all changes are checked for compliance with the underlying metamodel. Distributed team communication, coordination, and decision making are supported through synchronous and asynchronous chat features.
- AtoMPM [7], available since 2013, is one of the earliest in-browser modeling environment generators. The AtoMPM implementation has been optimized for DSLs with a graphical concrete syntax using W3C standards,

✉ Bernhard Rumpe
bernhard.rumpe@sosym.org

Jeff Gray
jeff.gray@sosym.org

[1] University of Alabama, Tuscaloosa Alabama, USA

[2] RWTH Aachen University, Aachen, Germany

such as SVG, JavaScript, and HTML, to support graphical concrete syntax. A strength of AtoMPM is its support for the specification, execution, and debugging of model transformations within the browser, allowing the user to debug their model transformations using both stepwise and omniscient debugging features. The collaborative features of AtoMPM are based on a scalable multi-view/multi-user implementation. Future work includes support for textual concrete syntax and hybrid representation (graphical + textual), as well as code generation options.

- WebGME [8] provides in-browser model editing capabilities that support sharing of model instance by several users. A branching scheme similar to version control systems is used by WebGME to allow multiple contributors to edit the same model. Model changes in WebGME are broadcast to all collaborators with affordances and visualizations to detect, retry, or reject concurrent edits. Future features will be added to WebGME to merge branches, which would allow a project to be forked, modified, and then integrated back into the master branch.
- MDEForge [9] is an extensible in-browser modeling platform. It fosters community-based modeling repositories that support the development, analysis, and reuse of modeling artifacts. An advanced query mechanism is provided to find desired artifacts within the repository. MDEForge also enables the adoption of model management tools as Software as a service that can be remotely used without overwhelming the users with intricate and error-prone installation and configuration procedures.

The current status of many of the MDE tools in this space is still immature, with much potential for future extension, robustness, and adoption. It will be interesting to see what development domains may benefit the most from such a fluently collaborative approach. We also need to understand the differences between real-time collaboration compared to a more version controlled approach provided by git and svn, where the developer can exchange the newest versions of development assets, but can also work in a closed, unconnected form as long as desired or necessary.

What are your thoughts on this topic? Are there other research or commercial efforts in this space that you would like the SoSyM readership to know about? If so, please see the next section about how to tweet your comments to the SoSyM handle!

1. Writely Blog. https://googleblog.blogspot.com/2006/03/writely-so.html
2. Overleaf. https://www.overleaf.com/
3. Scratch. https://scratch.mit.edu/
4. CloudMDE workshop series at MODELS. http://cloudmde.como.polimi.it/
5. Juri Di Rocco, Davide Di Ruscio, Ludovico Iovino, Alfonso Pierantonio, "Collaborative Repositories in Model-Driven Engineering," *IEEE Software*, May/June 2015, 32(3), pp. 28–34.
6. GenMyModel. https://www.genmymodel.com/
7. Jonathan Corley, Eugene Syriani, Huseyin Ergin, and Simon Van Mierlo (2016) "Cloud-based Multi-View Modeling Environments" In Cruz, A.M., Paiva, S. (Eds.) *Modern Software Engineering Methodologies for Mobile and Cloud Environments*, IGI Global.
8. Miklós Maróti, Tamás Kecskés, Róbert Kereskényi, Brian Broll, Péter Völgyesi, László Jurácz, Tihamer Levendovszky, Ákos Lédeczi, "Next Generation (Meta) Modeling: Web- and Cloud-based Collaborative Tool Infrastructure," *MPM@MoDELS 2014*, pp. 41–60.
9. Francesco Basciani, Juri Di Rocco, Davide Di Ruscio, Amleto Di Salle, Ludovico Iovino, Alfonso Pierantonio, "MDEForge: An Extensible Web-Based Modeling Platform," *CloudMDE@MoDELS 2014*, pp. 66–75.

1 SoSyM social media

We encourage authors and readers to follow the official SoSyM twitter handle (@sosym_journal). All papers that have been accepted and appear online first will be announced on Twitter. We will also announce journal special issue calls and other topics of interest to SoSyM readers. Please let us know your thoughts on how we can best serve the readership through social media (by, of course, tweeting to @sosym_journal)!

2 Editor updates

We are grateful for the many years of service that Brian Henderson-Sellers has offered to the SoSyM readership in his role as Editor. Brian will be ending his service to the journal with this issue. We are excited to welcome Benoit Combemale as a new SoSyM Editor. Thanks Brian, and welcome Benoit!

3 Content of this issue

This issue contains three papers and a guest editorial for the special section that emerged from the 2013 conference on Business Process Modeling, Development, and Support (BPMDS). This issue also contains the following 10 Regular Papers, as enumerated below.

Three of the papers in this issue were selected as the "SoSyM Journal-First" papers for 2015. These papers were first submitted and accepted in the normal SoSyM paper

selection process and then presented in October 2015 at MODELS 2015. The papers by Farwick et al., Song et al., and Rago et al. were the first three papers presented under this new model that we will continue in future collaboration with the MODELS conference series.

- "ReFlO: an interactive tool for pipe-and-filter domain specification and program generation" by Rui C. Gonçalves, Don Batory, and João L. Sobral
- **A SoSyM Journal-First paper**
  "A situational method for semi-automated Enterprise Architecture Documentation" by Matthias Farwick, Christian M. Schweda, Ruth Breu, and Inge Hanschke
- "Component-based verification using incremental design and invariants" by Saddek Bensalem, Marius Bozga, Axel Legay, Thanh-Hung Nguyen, Joseph Sifakis, and Rongjie Yan
- **A SoSyM Journal-First paper**
  "Formalizing and verifying stochastic system architectures using Monterey Phoenix" by Songzheng Song, Jiexin Zhang, Yang Liu, Mikhail Auguston, Jun Sun, Jin Song Dong, and Tieming Chen
- "View-based model-driven software development with ModelJoin" by Erik Burger, Jörg Henss, Martin Küster, Steffen Kruse, and Lucia Happe
- "Extracting finite state representation of Java programs" by Tamal Sen and Rajib Mall
- "Heuristic search for equivalence checking" by Nicoletta De Francesco, Giuseppe Lettieri, Antonella Santone, and Gigliola Vaglini
- "A framework for the operationalization of monitoring in business intelligence requirements engineering" by Corentin Burnay, Ivan J. Jureta, Isabelle Linden, and Stéphane Faulkner
- "Service feature modeling: modeling and participatory ranking of service design alternatives" by Erik Wittern and Christian Zirpins
- **A SoSyM Journal-First paper**
  "Identifying duplicate functionality in textual use cases by aligning semantic actions" by Alejandro Rago, Claudia Marcos, and J. Andres Diaz-Pace