

SED-ED, a workflow editor for computational biology experiments written in SED-ML

Richard R. Adams*

SynthSys Edinburgh, CH Waddington Building, University of Edinburgh, Edinburgh EH9 3JD, UK

Associate Editor: Martin Bishop

ABSTRACT

Summary: The simulation experiment description markup language (SED-ML) is a new community data standard to encode computational biology experiments in a computer-readable XML format. Its widespread adoption will require the development of software support to work with SED-ML files. Here, we describe a software tool, SED-ED, to view, edit, validate and annotate SED-ML documents while shielding end-users from the underlying XML representation. SED-ED supports modellers who wish to create, understand and further develop a simulation description provided in SED-ML format.

Availability and implementation: SED-ED is available as a standalone Java application, as an Eclipse plug-in and as an SBSI (www.sbsi.ed.ac.uk) plug-in, all under an MIT open-source license. Source code is at <https://sed-ed-sedmleditor.googlecode.com/svn>. The application itself is available from <https://sourceforge.net/projects/jlibsedml/files/SED-ED/>

Contact: richard.adams@ed.ac.uk

Received on January 16, 2012; revised on February 06, 2012; accepted on February 23, 2012

1 INTRODUCTION

The simulation experiment description markup language (SED-ML) (Waltemath *et al.*, 2011) is an XML language that aims to describe computational biology experiments so that they may be reproduced and exchanged between different software tools. In brief, SED-ML Level 1 encodes uniform time course simulation descriptions. It describes from where to obtain a model, the type of simulation to run and the content and presentation of the simulation output. SED-ML is designed to be machine-readable rather than human-readable, and to most modellers, SED-ML should be largely hidden. However, there is a need for curators and SED-ML document editors to easily view, annotate and validate SED-ML files without being exposed to the underlying XML. SED-ED is an application designed to address the needs of these users.

2 IMPLEMENTATION

The information encoded by SED-ML describes a sequential series of steps to perform a computational experiment and can be conveniently represented diagrammatically as a workflow graph. SED-ED provides a graphical editor to manipulate this workflow, in which nodes depict high-level SED-ML elements and edges the

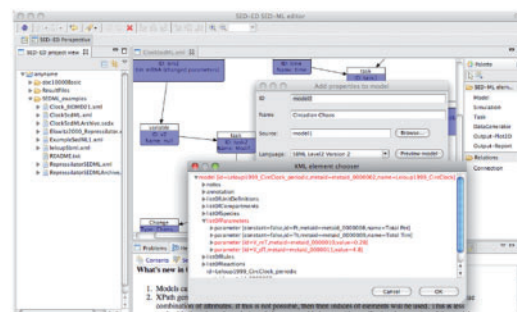


Fig. 1. A screenshot of the SED-ED editor. An overview of a SED-ML document's contents is shown in the left-hand panel and the contents displayed as a graphical workflow in the main panel. A dialog highlights in red how a model selected in the editor is altered compared with a reference model.

relations between them. Within the application window there are two main panes: the editor itself and an outline view that summarizes the SED-ML document's XML contents (Fig. 1).

SED-ED supports both *de novo* creation of SED-ML files and import of existing files, and a full graphical editor for creating elements and their interconnections. Any editor needs to provide freedom to enter data in a non-specified order, in order to be usable, while at the same time ensuring that persisted data is not corrupt or meaningless. Therefore, during the editing process it may be the case that the document is temporarily invalid (for example, because of duplicated identifiers, or invalid maths syntax). SED-ED automatically detects syntax errors and inconsistencies in the document, and displays them in a 'Problems View' from where the causes of the error can be accessed.

SED-ML re-uses several other standard representations and ontologies—HTML for annotations, XPath for identifying model elements, MathML (Ausbrooks *et al.*, 2003) to describe mathematical manipulations of both input and output variables, and the kinetic simulation algorithm ontology (KiSAO) (Courtot *et al.*, 2011) to describe simulation algorithms. SED-ED provides extensive editing support for these technologies so that users do not have to learn their details. For HTML, SED-ED provides a split-pane note editor showing the source HTML and a browser view. Mathematical statements are entered using a straightforward text-based syntax, based on the C programming language syntax. An embedded KiSAO ontology helps the user to supply correct, unambiguous information regarding applicable simulation procedures. XPath expressions are used in two circumstances in SED-ML: to identify variables and to identify the target of changes

*To whom correspondence should be addressed.

to be applied to the model to modify its behaviour. In the former case, SED-ED auto-generates XPath expressions based on the user's selection of a model element in a tree view of the model. This feature is available for any XML-based modelling language. Similarly, a point-and-click viewer identifies model elements targeted for changes. Once a change has been described, the differences compared with the original reference model are highlighted in colour.

In SED-ML, models can be referenced using any valid uniform resource identifier (URI). SED-ED will retrieve the contents of models automatically that are referenced in the local file system, by a URL, or from MIRIAM identifiers (Laibe and Novère, 2007) to the BioModels database (Li *et al.*, 2010).

For large SED-ML files, workflows can be complex. SED-ED has an inbuilt search that will highlight elements in the editor and the view can also be filtered, for example, to show only those elements that are used to generate a particular output.

The SED-ML specification (<http://sed-ml.org>) suggests the use of a binary archive file format to conveniently exchange resources that are currently unavailable from public databases. SED-ED can directly edit and view these archive files and can also generate archive files from the original resources.

SED-ED currently does not include the ability to actually simulate a model using the protocol described in a SED-ML document. However, the SED-ED plugin for SBSI (www.sbsi.ed.ac.uk) integrates with that tool's simulation capability, enabling the execution of both regular and archived SED-ML files, providing that the algorithm and modelling language are supported. Currently, SBSI is limited to execution of System Biology Markup Language (SBML, Hucka *et al.*, 2003) models.

3 DISCUSSION

This article presents the SED-ED application for working with SED-ML files. At present, software support for SED-ML is at an early stage, with varying degrees of conformance to the specification [e.g. CellDesigner 4.2 (Funahashi *et al.*, 2003), SED-ML Web Tools (<http://sysbioapps.dyndns.org/SED-ML%20Web%20Tools/>)]. This latter tool provides strong support for SED-ML, including simulation capability. However, while SED-ED uses a graphical approach to presenting a SED-ML document's contents to the user, SEDML Web Tools uses a text-based approach and a scripting language. Both these approaches have their advantages and end-users will benefit from the choice of software.

Software support is likely to increase as the standard becomes more widely adopted—SED-ML is part of the Combine (<http://co.mbine.org/>) initiative to coordinate the development of the various

community standards and formats in systems biology, which has the support of many simulation tool developers.

SED-ED has no dependencies on SBML and can therefore, be used with SED-ML documents referring to models written in any XML-based language. Future versions of SED-ED will contain plug-in points for modelling language-specific functionality.

The future development of SED-ED will naturally be tied to the development of the SED-ML language. The diagrammatic workflow representation we have chosen is adaptable to the addition of new language elements and will also become increasingly useful as the complexity of SED-ML workflows increases.

SED-ED is implemented as a collection of OSGi-compatible Java modules, and can therefore be readily integrated into an application environment that uses this technology.

Full user documentation is incorporated into the application, and an introductory tutorial is available online at <http://jlibsdml.sourceforge.net/sed-ed/tutorial.html>.

ACKNOWLEDGEMENTS

Dagmar Waltemath provided invaluable comments on the manuscript, Anna Zhukova suggested several improvements to the software and Frank Bergmann proposed the SED-ML archive format.

Funding: UK Biotechnology and Biological Sciences Research Council, and Engineering and Physical Sciences Research Council, to SynthSys, a Centre for Integrative Systems Biology (grant number BB/D019621/1).

Conflict of Interest: none declared.

REFERENCES

- Ausbrooks, R. *et al.* (2003) Mathematical Markup Language (MathML) version 2.0. W3C recommendation. World Wide Web Consortium. <http://www.w3.org/TR/MathML2/> (6 March 2012, date last accessed).
- Courtot, M. *et al.* (2011) Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol.*, **7**, 543.
- Funahashi, A. *et al.* (2003) CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO* **1**, 159–162.
- Hucka, M. *et al.* (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.
- Laibe, C. and Le Novère, N. (2007) MIRIAM resources: tools to generate and resolve robust cross-references in Systems Biology. *BMC Syst. Biol.*, **1**, 58.
- Li, C. *et al.* (2010) BioModels database: an enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Syst. Biol.*, **4**, 92.
- Waltemath, D. *et al.* (2011) Reproducible computational biology experiments with SED-ML – The Simulation Experiment Description Markup Language. *BMC Syst. Biol.*, **5**, 198.