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-sedmeditor.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

*To whom correspondence should be addressed.
This article presents the SED-ED application for working with SED-mbine.org/) initiative to coordinate the development of the various more widely adopted—SED-ML is part of the Combine (http://co.
users.org/SED-ML%20Web%20Tools/]. This approach to presenting a SED-ML document’s contents to the user, simulation capability. However, while SED-ED uses a graphical 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://com.
mine.org/mission 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://jibsedml.sourceforge.net/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