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 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). A dialog highlights in red how a model selected in the editor is altered compared with a reference model.

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.
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://sysbioupps.prodyhms.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://combine.net] 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-ins 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/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


