Towards collaborative and automated development of resources for data standards in synthetic biology

Jake Sumner Ajibode¹, Jacob Beal², James Scott-Brown³ Thomas E. Gorochowski⁴, Chris J. Myers⁵, Goksel Misirli¹

¹Keele University, ²Raytheon BBN Technologies, ³University of Oxford, ⁴University of Bristol, ⁵University of Colorado Boulder

w8q52@students.keele.ac.uk, jakebeal@ieee.org, james@jamescottbrown.com

thomas.gorochowski@bristol.ac.uk, chris.myers@colorado.edu, g.misirli@keele.ac.uk

1 INTRODUCTION

Data standards in synthetic biology are becoming ever more important as the number of tools addressing different needs increases, such as designing genetic circuits and visualizing and storing the designs. The Synthetic Biology Open Language (SBOL) [4] has been developed to provide a mechanism for the electronic exchange and common understanding of these designs and related information. Moreover, SBOL Visual [1] standardizes the representation of genetic circuit designs via well-defined glyphs.

The SBOL community has adopted GitHub for online collaboration, issue tracking and the development of specifications. The community website builds on GitHub pages and is accessible via HTTP. The SBOL data and visual standards follow an incremental release process, by applying common Git related processes such as branching for incremental developments and making and tagging releases with explicit version numbering. Standard enhancement proposals are prepared by community members, discussed, and finally incorporated into minor or major SBOL versions. The changes between each version can affect existing tools and additional mechanisms are needed for documentation and verification. As the number and frequency of these new releases, there is a need to automate related changes, prepare related documentation, and verify the changes. Such an automated process can also allow testing changes even before incorporating them into formal releases.

An automated approach is becoming more urgent for SBOL Visual. Each glyph comes with an SVG file, providing information about how the glyph can be connected to the others in genetic circuit designs, with additional features such as height and width.

Metadata about each glyph is stored in a markdown file to make the information more readable and editable for human access. Each markdown file includes information about the types of biological parts or interactions that the glyph can be used for, via ontological terms from the Sequence Ontology [3] and the Systems Biology Ontology [2]. In addition, these files include allowed alternative glyphs, free text notes and other related information. These markdown files should ideally be validated after each change in the context of all other glyph files and metadata from different files to prevent human error.

We previously developed SBOL Visual Ontology version 2 (SBOL-VO2) [6] to make the metadata about these glyphs machine accessible. Moreover, we developed the visual ontology web service (VOWS) to make the glyphs searchable and to find the most relevant glyph for a provided SO or SBO term for different types of biological parts and interactions. An ontological version of SBOL2 has also been developed [5]. However, the web service is currently not suitable for supporting an automated approach for new releases of the visual ontology or testing it against a specific snapshot of the files. Moreover, these resources need to be updated to incorporate the new SBOL 3 data and visual standards.

Here, we report the latest developments related to the automation process for SBOL Visual. This work involves developing a new SBOL Visual ontology for version 3 (SBOL-VO3) and improvements to the visual ontology web service for better user experience. We then demonstrate an automated workflow approach to auto-generate the visual ontology and related documentation to release standard specifications and glyphs more quickly and robustly with error checking.

2 SBOL VISUAL ONTOLOGY 3 (SBOL-VO3)

In the new version of the SBOL Visual ontology, ontological terms related to glyphs have been grouped into four main branches to represent sequence features, molecular species, molecular interactions for binary relationships, and biological processes with multiple inputs and outputs. Sequence feature terms are linked to respective SO terms, whilst SBO is used in all other terms, to indicate the types of all molecular interactions, processes and non-DNA based molecular terms. Another new feature is to capture network level representation of genetic circuits using directed edges. SBO terms are used to provide information about the roles of molecules in molecular interactions and processes, and hence the direction of edges. The hasHead and hasTail terms indicate the direction of binary interactions. For example, 'hasHead' some
(role some SBO:0000642)’ axiom for an inhibition interaction specifies that the the direction of the interaction starts from an entity with the inhibitor role (SBO:0000642). Similarly, hasIncoming and hasOutgoing properties indicate the inputs and outputs for participating biological molecules. This visual ontology refers to terms from the latest SBOL 3 data ontology (SBOL-OWL3).

Inline with the major changes in SBOL 3, we developed the proof-of-concept SBOL-OWL3 ontology. This ontology captures the rich SBOL 3 data model, which uses a graph-based approach to represent genetic circuit designs. Hence, SBOL-VO3 and SBOL-OWL3 ontologies act as a machine-accessible source to connect the SBOL Visual glyphs with SBOL specific data entities.

3 SBOL VISUAL ONTOLOGY WEB SERVICE

In this work, we added new features to the previously developed SBOL Visual Ontology Web Service [6]. Moreover, we decoupled the web service from a particular version of an SBOL Visual ontology and other related SO and SBO ontologies. These external resources are now loaded directly from the web. Different versions of the visual ontology is held at a GitHub repository. As default, the web service uses the latest version of the SBOL Visual ontology and the latest glyph files. However, users can also specify a particular release number. Alternatively, users can provide HTTP URLs for the ontology and the base glyph folder. These URLs can also be for a specific GitHub branch to allow testing of resources under development. Different media types for glyphs can also be requested.

4 AUTOMATION

Our goal is to develop a fully automated workflow to test incremental or major updates and use them directly via the SBOL visual web service (Figure 1). To facilitate this approach, we aim to use GitHub actions. As a result, specific GitHub-related actions such as new releases or commits to a particular branch can result in the auto-generation of the SBOL Visual ontology, related documentation, and the use of the visual glyphs via the web service directly. We developed scripts that can be linked to GitHub actions so that the SBOL community to take advantage of these resources in a timely fashion.

5 CONCLUSION

Ontologies are ideal to capture domain knowledge. Here, we used this approach to provide machine accessible representation of the SBOL 3 data and visual standards in the form of ontologies. In order to help the SBOL community deal with incremental and major changes, we adopt an automated approach based on GitHub actions. This automation enables testing changes rapidly and releasing them to the community quickly. The improvements enhance the user experience via different additional options when using the web service. The SBOL community is working on parametric glyphs that can be customised for different properties such as colour, width, and height. Our goal is to incorporate additional metadata about these parameters into the SBOL Visual ontology and use the web service to serve parametric glyphs for design and layout tools in the future.

6 ACKNOWLEDGEMENTS

We thank SBOL Industrial Consortium for funding this work, and the SBOL community for continuous support sand feedback. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations.

REFERENCES


Figure 1: A use case involving the use the automated approach. A user makes a new release due to a change in metadata about an SBOL Visual glyph. As a result, the SBOL Visual ontology and the online documentation is auto-generated. Tools can start using this change immediately. Alternatively, they can start working on a specific version of the ontology and glyphs. Tools can either query the ontology file or utilise the web service to find the most suitable glyphs for given SO or SBO terms.