The Synthetic Biology Open Language Supports Integration of the Engineering Life-Cycle for Synthetic Biologists

Bryan Bartley[1], Christian Atallah[2], Alasdair Humphries[2], Vishwesh Kulkarni[3], Curtis Madsen[4], Goksel Misirli[5], Angel Goni-Moreno[2], Tramy Nguyen[6], Ernst Oberortner[7], Nicholas Roehner[1], Meher Samineni[6], Zach Zundel[6], Jacob Beal[1], Chris Myers[6], Herbert M Sauro[8], Anil Wipat[2]
[1]Raytheon BBN Technologies, [2]Newcastle University, [3]University of Warwick, [4]Boston University, [5]Keele University, [6]University of Utah, [7]DOE Joint Genome Institute, [8]University of Washington
bryan.a.bartley@raytheon.com

MOTIVATION

A critical bottleneck for large-scale engineering collaboration in synthetic biology has been the inability to integrate data through successive stages of the design-build-test-learn (DBTL) engineering life-cycle. These workflows generate large volumes of data and physical artifacts (e.g., DNA samples and cell stocks) that are difficult to organize, track, and manage without systematized, automated tool chains.

The DBTL cycle is a generalized, iterative framework for engineering problem-solving—something like a scientific method for engineers. In the context of synthetic biology, the DBTL cycle may include processes such as pulling data about biological parts from online databases, assembling new genetic programs from DNA sequences, synthesizing and assembling DNA, performing quality control, measurement and model-based characterization of a DNA part's encoded behavior, submitting characterized parts to inventories, and publishing data sheets. Ideally, each cycle generates new knowledge that feeds back into new cycles in the form of alternative approaches, reformulated problems, or forward specifications for future designs.

In this abstract, we describe how the *Synthetic Biology Open Language* (SBOL) data exchange standard has recently been extended to enable documentation, automation, and integration of DBTL pipelines to support large-scale, distributed research and development in synthetic biology.

REPRESENTING WORKFLOWS

Distributed collaboration in synthetic biology requires integrating a diverse network of resources, including software tools, automation, instrumentation, databases, and repositories. In order for these resources and, by extension, the collaborative, scientific communities they support to communicate and operate efficiently, data standards are needed. This abstract reports recent developments in data exchange using the SBOL standard that support large-scale collaboration among diverse communities of experimental and computational synthetic biologists.

SBOL is a data exchange standard intended to support reuse and reproducibility of prior scientific work by synthetic biologists and developed through an open, community-wide process. SBOL defines a high-level data model that represents important conceptual knowledge for human users, while, at a lower level, data is serialized in a machine-readable RDF/XML format that enables semantic interoperability between distributed resources. Since the standard

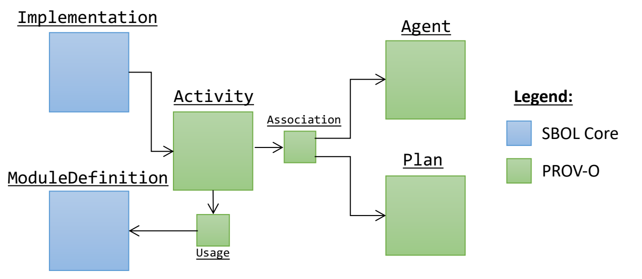

Figure 1: PROV-O and SBOL classes used to represent the design-build-test-learn (DBTL) engineering life-cycle.

was originally released [3], its scope has expanded [8], most recently with SBOL 2.2, which includes support for integration of computational workflows with experimental workflows [2].

In order to describe multi-stage workflows, SBOL leverages the *World Wide Web Consortium* (W3C) *Provenance Ontology* (PROV-O) [7]. Provenance may be defined as a form of structured meta-data that describes the execution of processes in which one artifact is transformed into another. This record is essential for understanding where data comes from, deciding whether it should be trusted, and integrating it with other information sources. In PROV-O, workflows are represented as a directed, acyclic graph linked by **Activities** executed by **Agents** (e.g., persons, robotics, and/or software tools) according to a **Plan** upon **Usages** of other artifacts (Figure 1). The PROV-O data model is deliberately generic. Although it has heretofore been used primarily for describing computational workflows, it has been adapted for use in SBOL to describe experimental workflows as well.

More specifically, SBOL classifies provenance **Activities** according to a simple DBTL ontology. Data objects produced by "design" **Activities** represent an engineer's intended design and are purely conceptual (these are typically **ComponentDefinition** or **ModuleDefinition** objects). Data objects produced by "build" **Activities** represent physical artifacts such as DNA samples or cell lines (these are typically **Implementation** objects). These in turn may be subject to "test" **Activities**, an experimental measurement that results in new data objects (these are typically an **Attachment** or a **Collection** of **Attachment** objects). These data are then subject to reduction and analysis through "learn" **Activities** (learned objects can be any other type of object representing what has been learned). Workflows in SBOL may consist of any number of specialized steps and protocols, but by and large they are expected to fit into the

DBTL abstraction, as this is how workflows are often presented in synthetic biology literature [1, 4].

Figure 2 represents one complete iteration through a hypothetical DBTL cycle. The cycle starts with a model specifying a gene circuit's desired behavior. A parts-based design is created with the iBioSim tool. Subsequently, a DNA construct is implemented in the lab by a technician and its behavior is measured using an automated plate-reader protocol. Finally, the data are fit with a mathematical model in order to characterize the observed behavior, which may or may not match the original specification. This scenario represents a model-based design approach to synthetic biology.

APPLICATIONS AND TOOL SUPPORT

SBOL 2.2 is currently being used for distributed collaboration in several large-scale synthetic biology efforts that are characterizing genetic parts and designs, including the NSF Living Computing Project and the EPSRC-funded Synthetic Portabolomics project. These efforts involve diverse communities of computational and experimental researchers. To support these efforts, software tools have been developed and upgraded to support PROV-O. These include, among others, synthetic biology design tools, such as SBOLDesigner [9], modeling tools, such as iBioSim [5], and data repositories, such as SynBioHub [6]. SynBioHub, in particular, provides a means to browse provenance histories online using its web interface. These tools leverage open source software libraries for Java (libSBOLj) [10], JavaScript (sboljs), C++ (libSBOL), and Python (pySBOL), which have been extended to support the creation and search of provenance histories using PROV-O and networked data exchange with SynBioHub instances.

CONTRIBUTIONS

SBOL 2.2 is an enabling technology that supports large-scale engineering efforts in synthetic biology [2]. Teams of synthetic biologists using experimental and computational methods may collaborate better using a growing company of resources and infrastructure that communicate with SBOL. Genetic designs, laboratory samples, and experimental data can be linked together by provenance histories, making it easier for one synthetic biologist to reuse the work of another. The pace of synthetic biology innovation will likely improve as synthetic biologists proceed through many iterations of DBTL with computer-aided and automated technologies.

ACKNOWLEDGMENTS

The authors of this work are supported by the National Science Foundation under Grant No., 1522074 (J.B., M.S., and C.M.), DBI-1355909 (H.M.S. and B.B.), and DBI-1356041 (Z.Z., T.N., and C.M.). A.W. was funded by Engineering and Physical Sciences Research Council (EPSRC) Grant Nos. EP/N031962/1 and EP/J02175X/1. A.G-M is funded by EPSRC grant EP/R019002/1. C.A. acknowledges studentship funding from the EPSRC and from Prozomix Ltd. E.O. has been funded through contract DE-AC02-05CH11231 between Lawrence Berkeley National Laboratory and the U.S. Department of Energy. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

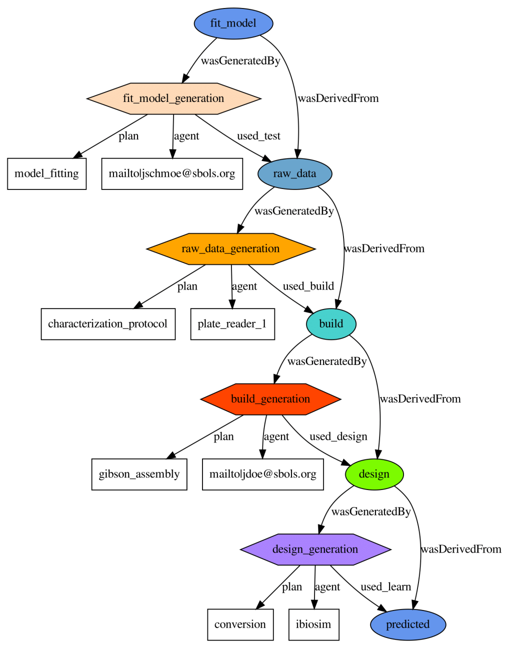

Figure 2: A hypothetical DBTL workflow representing model-based design. This figure is rendered using a Python tool for rendering SBOL 2.2 workflows (https://github.com/chrisAta/sbol-provo-viz).

REFERENCES

[1] E. D. Carlson. Creating ribo-t:(design, build, test) n, 2015.

[2] R. S. Cox, C. Madsen, J. A. McLaughlin, T. Nguyen, N. Roehner, B. Bartley, J. Beal, M. Bissell, K. Choi, K. Clancy, et al. Synthetic biology open language (SBOL) version 2.2.0. *Journal of integrative bioinformatics*, 2018.

[3] M. Galdzicki, K. P. Clancy, E. Oberortner, M. Pocock, J. Y. Quinn, C. A. Rodriguez, N. Roehner, M. L. Wilson, L. Adam, J. C. Anderson, et al. The synthetic biology open language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature biotechnology*, 32(6):545, 2014.

[4] C. A. Hutchison, R.-Y. Chuang, V. N. Noskov, N. Assad-Garcia, T. J. Deerinck, M. H. Ellisman, J. Gill, K. Kannan, B. J. Karas, L. Ma, et al. Design and synthesis of a minimal bacterial genome. *Science*, 351(6280):aad6253, 2016.

[5] C. Madsen, C. J. Myers, T. Patterson, N. Roehner, J. T. Stevens, and C. Winstead. Design and test of genetic circuits using iBioSim. *IEEE Design Test of Computers*, 29(3):32–39, June 2012.

[6] J. A. McLaughlin, C. J. Myers, Z. Zundel, G. Misirli, M. Zhang, I. D. Ofiteru, A. Goñi Moreno, and A. Wipat. SynBioHub: A standards-enabled design repository for synthetic biology. *ACS synthetic biology*, 7(2):682–âĂŞ688, 2018.

[7] P. Missier, K. Belhajjame, and J. Cheney. The W3C PROV family of specifications for modeling provenance metadata. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 773–776. ACM, 2013.

[8] N. Roehner, J. Beal, K. Clancy, B. Bartley, G. Misirli, R. GrÃijnberg, E. Oberortner, M. Pocock, M. Bissell, C. Madsen, et al. Sharing structure and function in biological design with SBOL 2.0. *ACS synthetic biology*, 5(6):498–506, 2016.

[9] M. Zhang, J. A. McLaughlin, A. Wipat, and C. J. Myers. SBOLDesigner 2: an intuitive tool for structural genetic design. *ACS synthetic biology*, 6(7):1150–1160, 2017.

[10] Z. Zhang, T. Nguyen, N. Roehner, G. Misirli, M. Pocock, E. Oberortner, M. Samineni, Z. Zundel, J. Beal, K. Clancy, et al. libSBOLj 2.0: a java library to support SBOL 2.0. *IEEE life sciences letters*, 1(4):34–37, 2015.