

The Synthetic Biology Open Language 2.0

Bryan Bartley
University of Washington, US
bbartley@uw.edu

Goksel Misirli
Newcastle University, UK
goksel.misirli@ncl.ac.uk

Jacob Beal
Raytheon BBN Tech., US
Jakebeal@bbn.com

Nicholas Roehner
Boston University, US
nroehner@bu.edu

Kevin Clancy
ThermoFisher Scientific, US
kevin.clancy1@thermofisher.com

Matthew Pocock
Turing Ate My Hamster, UK
turingatemyhamster@gmail.com

1. INTRODUCTION

The initial version of the Synthetic Biology Open Language (SBOL) was designed for the exchange of information about biological designs at the DNA level. As the field of synthetic biology matures, however, there is a clear need to extend SBOL to capture the function of biological designs and their structure beyond annotated DNA sequences [2]. To support the specification of increasingly complex and diverse biological designs, standards need to represent data on both biological structure and function in a modular, hierarchical fashion. These include data on biological interactions, which are especially important for the functional composition of biological components, and meta-data on computational models, which are important for linking biological designs to more detailed descriptions of their behavior in specific biological contexts.

SBOL 1.1 provides entities to represent biological systems as composite DNA designs [1]. In particular, biological parts are represented in SBOL 1.1 using `DnaComponent` entities. These entities can be reused in different designs, constituting building blocks of larger and more complex `DnaComponent` entities.

SBOL 2.0 builds conceptually upon the DNA-centric SBOL 1.1 data model in two directions. First, SBOL 2.0 generalizes the concept of a DNA component to support a wide range of biological components, including RNA, proteins, and metabolites. This generalization enables the structural diversity of biological designs to be fully captured. Second, SBOL 2.0 introduces a functional data model to complement its structural data model, thereby enabling specification of the dynamic interactions and processes of a design in a lightweight manner, without commitment to any specific quantitative modeling framework. Ultimately, SBOL 2.0 provides a system of hierarchical constructs for describing both the structure and function of modular biological designs.

2. SBOL 2.0 DATA MODEL

As shown in Figure 1, SBOL 2.0 offers a rich set of design entities, including `ComponentDefinitions`, `Sequences`, `ModuleDefinitions`, `Models`, `Collections` (not shown), and `GenericTopLevels` (not shown). These entities enable the design of biological systems from composable, modular, and reusable building blocks. Examples can be found in the SBOL 2.0 specification, online at <http://sbolstandard.org/>.

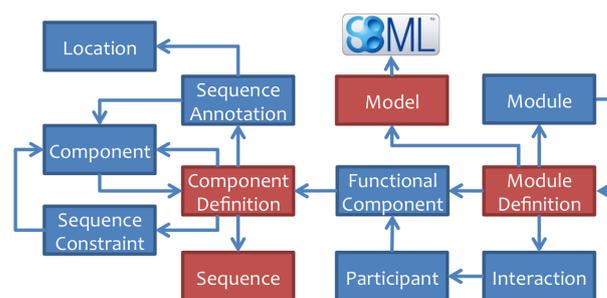


Figure 1: Red boxes represent the top level entities that may encapsulate entities represented in blue boxes. Arrows indicate property relationships (un-labeled for simplicity). The left half of the diagram is a generalization of SBOL 1.1 to include molecules other than DNA, while the right half is entirely new.

Component Definitions. Biological components are represented in SBOL 2.0 using the `ComponentDefinition` entity, which provides an improved representation of component compositions and their associated structural constraints. In SBOL 1.1, sub-components are represented via `SequenceAnnotations`. However, this representation requires even small regions of DNA, such as start codons, to be defined as reusable components. SBOL 2.0 `SequenceAnnotations`, on the other hand, simply indicate regions of interest that can refer to sub-components if desired. These sub-components are represented by `Component` entities. Furthermore, additional entities are introduced to represent different types of `Locations` for `SequenceAnnotations`, such as a cuts between adjacent base pairs and ranges. As in SBOL 1.1, SBOL 2.0 also supports the representation of partial designs, in which precise locations may not be known. Rather than use `SequenceAnnotations` to explicitly encode sub-component ordering, SBOL 2.0 represents this and other biological structural relationships between sub-components using `SequenceConstraint` entities.

Beyond DNA, the `ComponentDefinition` entity of SBOL 2.0 can also be used to represent different types of biological entities, such as RNA, protein, metabolites, small molecules, and complexes. The types and roles of these entities reference existing data in the form of terms from ontologies. For example, the roles of a `ComponentDefinition` can define whether it is a promoter or coding sequence by referring to terms from the Sequence Ontology.

Sequences. In SBOL 2.0, more general sequence information can be attached to different types of `ComponentDefinitions`. The International Union of Pure and Applied Chemistry (IUPAC) encodings are used to specify the nucleotide and amino acid `Sequences` of DNA, RNA and protein components. The Simplified Molecular Input Line Entry System (SMILES) encoding is recommended to specify the `Sequences` (atomic structures) of small molecules.

Module Definitions. A `ModuleDefinition` entity can be used to link several entities to represent the function of a biological system design. Each `ModuleDefinition` includes `FunctionalComponents`, which are defined by `ComponentDefinitions`, and the `Interactions` between these components. Information about `Interactions` is crucial to specify the qualitative functional details of a design. Each `Interaction` has one or more `Participations` that elaborate on the roles of participant `FunctionalComponents`.

Each `ModuleDefinition` can also indicate its inputs and outputs, thereby informing its composition and reuse by parent entities. For example, a parent `ModuleDefinition` can import other `ModuleDefinitions` as `Modules` and map the inputs/outputs of these sub-modules to its own. This approach aids machine reasoning and automation to compose modules into designs for complex biological systems.

Models. `Model` entities document references to actual sources for quantitative or qualitative models. Each model entity includes the model source, framework, and language. Although Figure 1 shows an SBML model linked to a `Model` entity, it is important to note that the model can be encoded in any language, such as CellML, Matlab, etc.

Extension via Annotations. In addition to the entities described here, SBOL provides an annotation framework for application-specific information. Namely, each entity in an SBOL file can be annotated with Resource Description Framework (RDF) properties. Furthermore, application-specific entities can be included as RDF documents. SBOL libraries make these custom annotations and documents available to tools as generic properties and `GenericTopLevel` entities that are preserved during subsequent read and write operations.

3. SERIALIZATION AND LIBRARIES

SBOL documents are serialized using RDF, taking advantage of the rich tool ecosystem for this Semantic Web technology. Unique Uniform Resource Identifiers (URIs) identify each entity in a SBOL document. Libraries to read and write SBOL 2.0 documents are available in several languages, with ongoing support and development by the SBOL community. The Java library, libSBOLj 2.0 [3], is the most mature. This library is backwards compatible and can import SBOL 1.1 data into SBOL 2.0 data objects. Other ongoing library development efforts include Scala and C libraries.

4. CONTINUED DEVELOPMENT

Beyond the extensions added by SBOL 2.0, the SBOL standard is undergoing continuous development to represent more information about different types of biological system

designs. In some cases, there is not yet sufficient scientific consensus for effective standards development. Currently, the most pressing area for development is capturing data on biological context, such as experimental conditions, chassis, and growth media. Such information is not yet captured in the core objects of the standard, but can be encoded for testing as annotations and `GenericTopLevel` entities.

In this and other extension initiatives, SBOL uses existing standards and resources whenever possible. For example, SBOL already leverages existing ontologies for terms to define the types, roles, and other properties of entities in the SBOL data model. In general, SBOL 2.0 links to these external resources via placeholders and provides guidelines for their use with limited enforcement.

Finally, the development of SBOL is carried out openly and iteratively with the community feedback. SBOL is also now part of COMBINE, an initiative to coordinate the development of standards for computational modeling in biology, which aids in the application of best practices for the development of data standards.

5. ACKNOWLEDGMENTS

Beyond the listed authors, contributions to the SBOL standard have been made by many individuals and organizations that participate in the SBOL Developers Group. The work reported here has been partially supported by the National Science Foundation under Grant Number DBI-1356041 and DBI-1355909, and the Engineering and Physical Sciences Research Council under grant EP/J02175X/1. 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 our funding agencies.

6. ADDITIONAL AUTHORS

Tramy Nguyen, Zhen Zhang, Chris Myers (U. of Utah, US, {tramy.nguyen, zhen.zhang, chris.john.myers}@utah.edu), John Gennari, Herbert Sauro (U. of Washington, US, {gennari,hsauro}@uw.edu), Curtis Madsen, Anil Wipat (Newcastle U., UK, {curtis.madsen,anil.wipat}@ncl.ac.uk), Ernst Oberortner (DOE Joint Genome Institute (JGI), US, eoberortner@lbl.gov), Michael Bissell (Amyris, Inc., US, bissell@amyris.com)

7. REFERENCES

- [1] M. Galdzicki et al. The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature Biotechnology*, 32(6):545–550, 2014.
- [2] N. Roehner, E. Oberortner, M. Pocock, J. Beal, K. Clancy, C. Madsen, G. Misirli, A. Wipat, H. Sauro, and C. J. Myers. Proposed data model for the next version of the Synthetic Biology Open Language. *ACS Synthetic Biology*, 4(1):57–71, 2015.
- [3] Z. Zhang, T. Nguyen, N. Roehner, G. Misirli, M. Pocock, E. Oberortner, J. Beal, K. Clancy, A. Wipat, and C. Myers. libSBOLj 2.0: A Java library to support SBOL 2.0. In *IEEE SY-Bio Workshop to Address Topics in Systems and Synthetic Biology*, Dallas, US, Mar. 2015.