Recent Advances in the Synthetic Biology Open Language

Michal Galdzicki Ernst Oberortner Matthew Pocock Jacqueline Quinn Mandy Wilson
Evan Appleton Bryan Bartley Jacob Beal Swapnil Bhatia Robert Cox Raik Grünberg
Goksel Misirli Hector Plahar Nicholas Roehner Larisa Soldotova Guy-Bart Stan
Doug Densmore Chris J. Myers Herbert Sauro Anil Wipat

1. INTRODUCTION

A significant concern in the synthetic biology community is the difficulty in reproducing results reported in the literature [5]. To address this problem, in 2008, a small group of researchers proposed the development of the *synthetic biology open language* (SBOL), an open-source standard for the exchange of genetic designs. In 2011, the first version of the SBOL core data model was released [2]. In 2013, the first version of a standard for visualization of genetic designs expressed in SBOL was also released [6]. Leveraging `libSBOLj`, a java-based library for SBOL's core data model, 18 software tools now support SBOL. While this represents excellent progress, there is still a lot of work to do. In April 2013, several members of the SBOL developers group met at Newcastle University to discuss SBOL's next steps. This abstract briefly describes two significant outcomes of this meeting: a plan to improve SBOL adoption and several significant enhancements to SBOL's data model.

2. SBOL ADOPTION PLAN

As mentioned above, SBOL evolved out of a desire to improve the reproducibility of research results. Figure 1 summarizes a plan discussed at the workshop that leverages SBOL to achieve this goal. We are working with the editors of the journal ACS Synthetic Biology to encourage authors of papers describing genetic designs to submit information about their design. The *Joint Bio Energy Institute* (JBEI) has created a repository called `JBEI-ICE` for storing information about genetic designs [3]. It has been proposed that designs be deposited at a public `JBEI-ICE` repository and that their EntryID be provided to the journal upon paper submission. This would be akin to submissions of DNA sequences to `GenBank` or `EMBL` but in this case authors submit designs. The submitted information would remain private, but reviewers of their paper would be given access in order to access the quality of the provided information. Upon paper acceptance, this submitted information would become publicly accessible. The `JBEI-ICE` tool supports export of SBOL RDF/XML files, so this would enable both the authors and other users to import these designs into their *genetic design automation* (GDA) tools. In the future, these GDA tools may be used directly to submit the designs to the repository. Another added benefit is the ability to export graphic design images in the SBOL visual standard using Graphviz or the software tool `Pigeon` [1]. We hope that these generated images are ultimately used in the publication of the design.

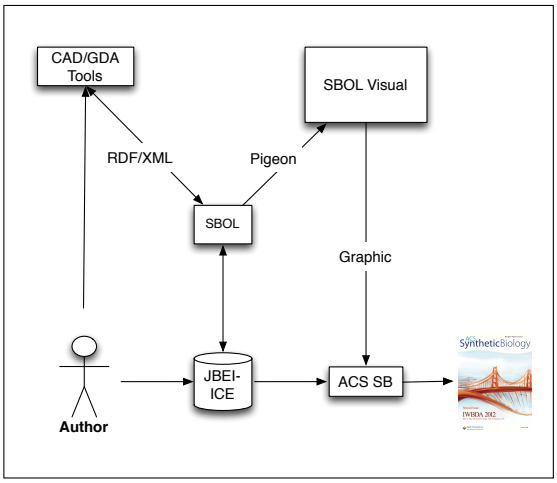

Figure 1: SBOL adoption plan.

3. ENHANCEMENTS TO SBOL

To improve the utility and adoption of SBOL, it is essential that it is able to capture all the relevant information for a genetic design. The first version of SBOL enables exchange of annotated DNA sequences. In particular, an SBOL file is composed of *DNA components* which have a *unique reference identifier* (URI), an id, a name, a description, a type taken from the Sequence Ontology, and a *DNA sequence*. These DNA components are annotated using *sequence annotations* which specify positions within a DNA sequence that are to be annotated with simpler DNA components.

There are several aspects of genetic design that cannot be exchanged using the first version of SBOL. At the workshop, several enhancements to the SBOL data model were proposed as shown in Figure 2 (note that most objects include URI, id, name, and description fields which are not shown to save space). First, the *component* class is generalized to allow it to not only represent annotated DNA sequences, but also annotated RNA and protein sequences. Next, the *device* class has been added to allow the description of groups of components that together perform a desired function. Devices can be either *primitive* (i.e., composed of components) or *composite* (i.e., composed of other devices). Next, the *system* class is added as a way to group a collection of devices with a description of their *context* under which they are to be used. The context class is used to provide information about the experimental methods including the

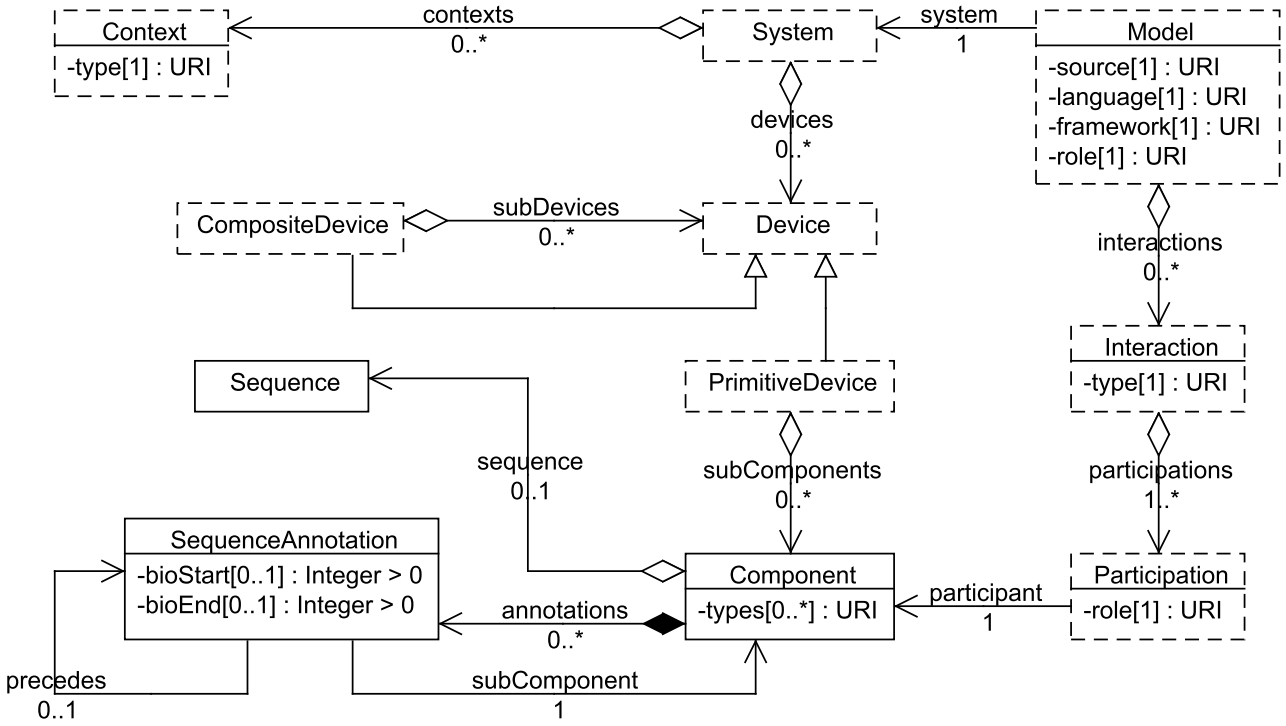

Figure 2: Proposed enhancements to SBOL's data model.

strain of the host, the medium in which the host resides, the container in which the medium is stored, the environmental conditions, and the measurement device used to study the system. Precise details about the experimental context are often essential to the reproducibility of laboratory results.

The final significant enhancement proposed is a mechanism for attaching models to genetic designs. There has been significant work in the development of standards for modeling biological systems, such as the *systems biology markup language* (SBML) [4]. The SBOL community does not wish to duplicate these efforts. Therefore, the *model* class allows one to reference a model for a system expressed in an external file using any modeling language of choice. The model class includes a URI to reference an external model file, a URI to identify the language of the model (SBML, CellML, matlab, BNGL, etc.), a URI to identify the modeling framework (ODE, stochastic, Boolean, etc.), and a URI to identify the purpose of the model (specification, back annotation, etc.). One important aspect of modeling for genetic designs is information about the interactions between the components in the design. For example, a protein component may repress a DNA component of type promoter. Typically, this information is stored in the model file, but it is useful for the user to have ready assess to the information, for example, to draw regulation constructs in a genetic circuit diagram. An *interaction* class is proposed to provide this type of information to the user. Each interaction includes a URI from the Systems Biology Ontology to specify its type (repression, activation, etc.). An interaction includes a number of component participants which are each given a URI for their role (repressor, repressed, etc.). While typically, these interactions are stored in an accompanying model file, if the source of the model is the SBOL file itself, then these interactions can be stored within the SBOL file.

4. CONCLUSION

While this abstract describes some exciting proposals being discussed in the SBOL community, it should be emphasized that they are just proposals, and we would value further input from people in the synthetic biology community. Many problems remain to be solved, and we look forward to further stimulating discussion at IWBDA.

5. ACKNOWLEDGMENTS

There are over 80 members of the SBOL developers group. While not all members were able to participate in the SBOL workshop, we would like to acknowledge their support of SBOL through numerous lengthy discussions on the SBOL mailing list.

6. REFERENCES

[1] S. Bhatia and D. Densmore. Pigeon: a design visualizer for synthetic biology. *ACS Syn. Bio.*, April 2013.

[2] M. Galdzicki et al. Synthetic biology open language (SBOL) version 1.0.0. BBF RFC #84.

[3] T. Ham et al. Design, implementation and practice of jbei-ice: an open source biological part registry platform and tools. *Nucleic Acids Res.*, 40(18), 2012.

[4] M. Hucka et al. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinfo.*, 9(4), 2003.

[5] J. Peccoud et al. Essential information for synthetic dna sequences. *Nature Biotechnology*, 29(22), 2011.

[6] J. Quinn et al. Synthetic biology open language visual (SBOL Visual) version 1.0.0. BBF RFC #93.