libSBOLj 2.0: A Java Library to Support SBOL 2.0

Zhen Zhang[1], Tramy Nguyen[1], Nicholas Roehner[2], Goksel Misirli[3], Matthew Pocock[3], Ernst Oberortner[4], Jacob Beal[5], Kevin Clancy[6], Anil Wipat[3], Chris Myers[1]

[1]University of Utah
[2]Boston University
[3]Newcastle University
[4]DOE Joint Genome Institute
[5]Raytheon BBN Technologies
[6]ThermoFisher Scientific

COMBINE 2015
October 13, 2015

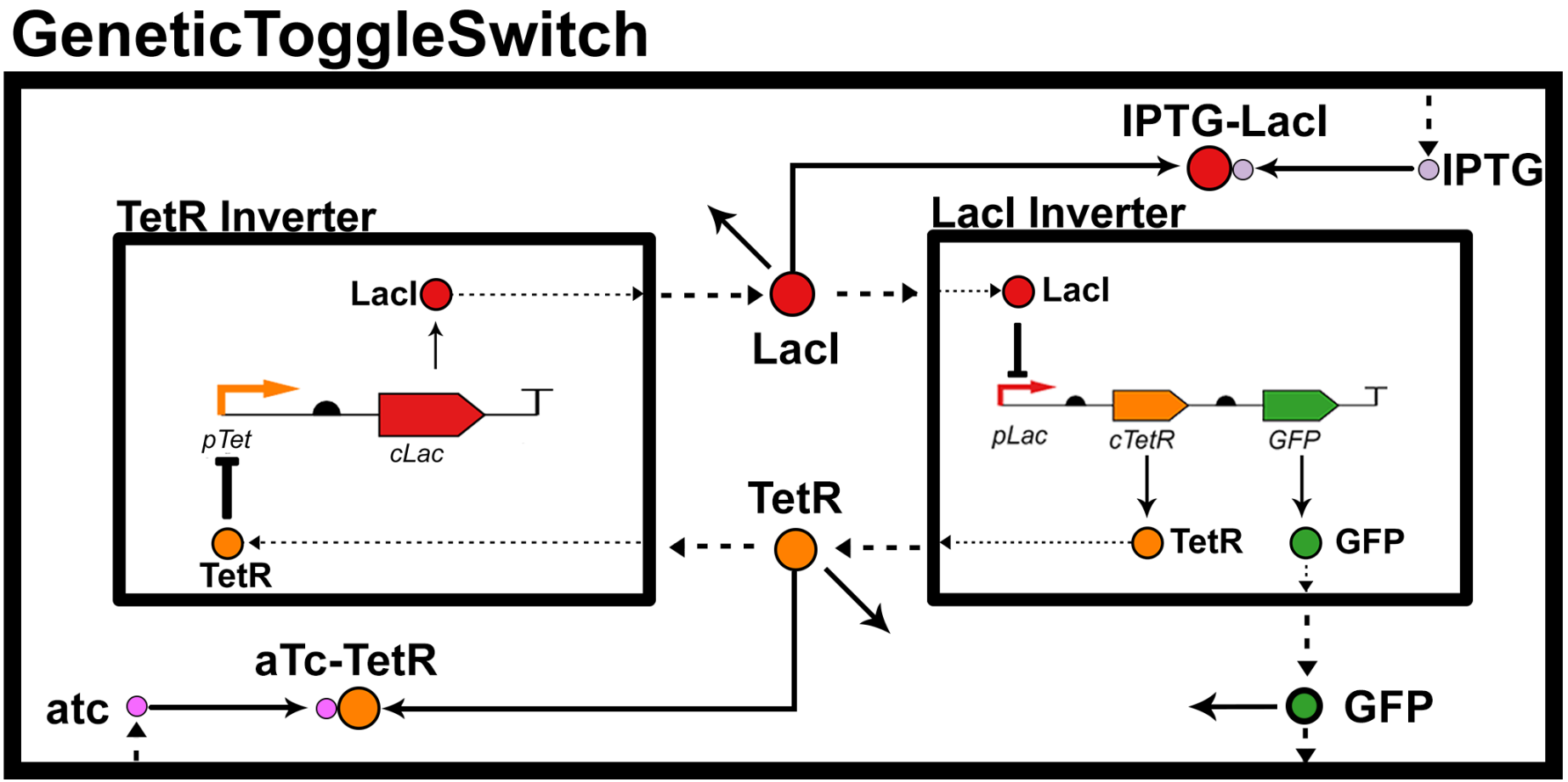
libSBOLj 2.0

- Crucial to the success of a standard is software infrastructure to support developers' integration of the standard within their tools.
- `libSBOLj` is a native Java implementation of the SBOL data structure, which provides an *application programmers interface* (API) for tool developers to interact with SBOL data objects.
- Leveraging `libSBOLj` many software tools now support SBOL 1.1.
- `libSBOLj 2.0` will ease the adoption of SBOL 2.0 by tool developers.
- `libSBOLj 2.0` is a native Java implementation of the SBOL 2.0 data model, enriched with an API to instantiate and link data objects.
- Library distribution includes detailed documentation of the class definitions and the methods provided by the API.

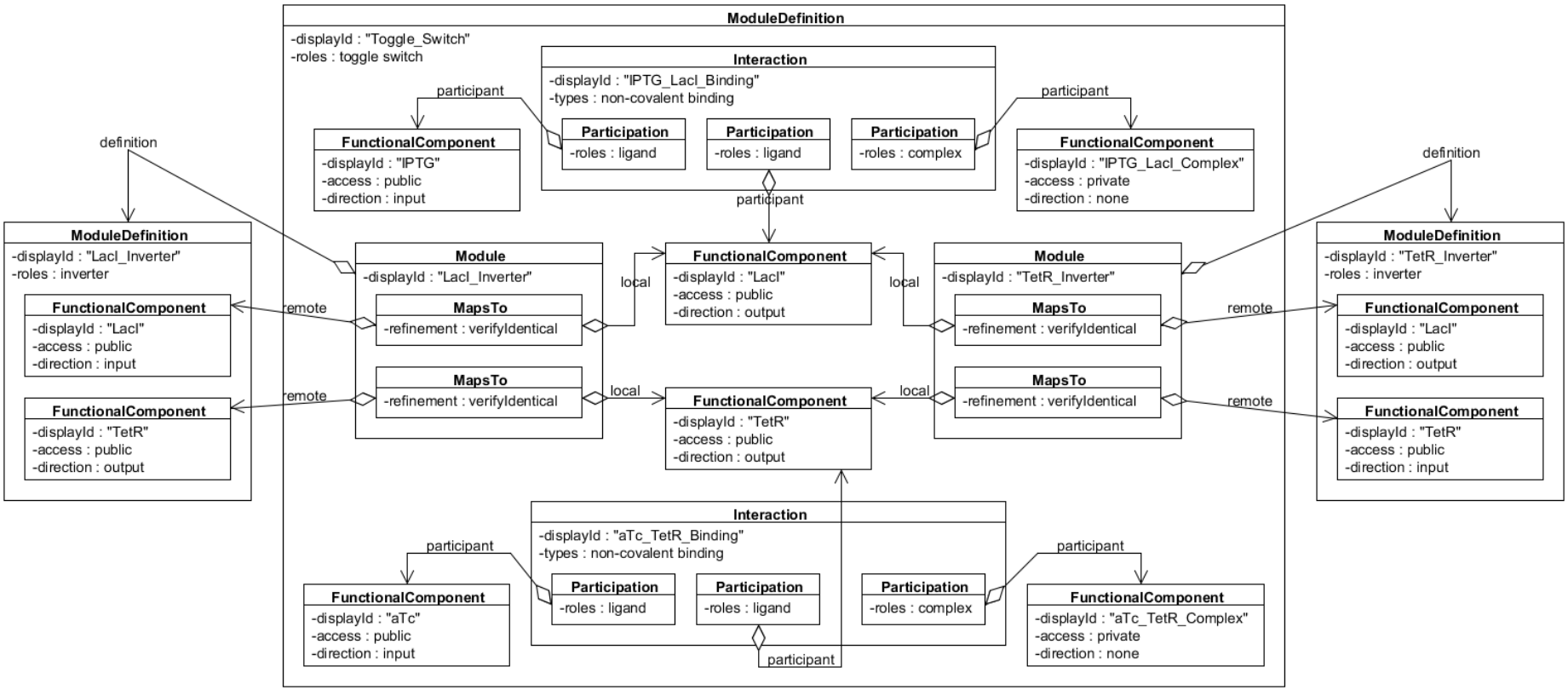
SBOL 2.0 Data Model

- `libSBOLj 2.0` organizes all SBOL data within an *SBOL document*.
- Includes a list of each type of *top level* object: *collections*, *modules*, *components*, *sequences*, *models*, and *generic top level* objects.
- These lists are organized as hash maps to allow for easy search by their *unique reference identifiers* (URIs) and validation that they are distinct.
- Library includes methods for creating, updating, accessing, and removing these data objects, as well as, their child objects.

Genetic Toggle Switch (Gardner et al. 2000)



Hierarchical UML Diagram for Part of the Toggle Switch



Compliant URIs

- `libSBOLj 2.0` provides added functionality and performance enhancements for those using *compliant URIs*.

$$\text{http://}\langle\text{prefix}\rangle/\langle\text{displayId}\rangle/\langle\text{version}\rangle$$

- Prefix is *uniform resource locator* (URL) established by owner of object.
- DisplayId must uniquely refer to an SBOL object in its namespace.
- Multiple versions of this SBOL object must have a unique version field.
- A child of a top level object adopts the URI of its parent object except that its displayId is inserted between its parent's displayId and version.
- Similarly, a child of this child object would add its displayId to the URI.
- There are several implementation advantages to using compliant URIs.
  - Easier to read, which facilitates debugging.
  - Enables distributed architecture of hash maps to maintain uniqueness and provide an efficient mechanism to search for an object by its URI.

Persistent Identity

- Each SBOL object in addition to having an identity URI also has a *persistent identity* URI which is simply its URI without the version.
- Using the persistent identity, another object can refer to the latest version of this object.
- The latest version is determined using MAVEN versioning conventions.

Serialization

- `libSBOLj 2.0` enhances the efficiency and extensibility of serialization.
- Supports reading/writing data using RDF/XML, Turtle, and JSON formats.
- Software tools that need to store data that is not currently encoded within SBOL can do so using generic top level objects and custom annotations.
- When the library reader encounters a tag for a top level object that it does not recognize, this data is stored within a generic top level object.
- Within top level objects, when a tag is not recognized the data is stored within a custom annotation object.
- Tools using our library that do not recognize custom data will round-trip the data unmodified when writing and reading SBOL files.
- Tools that would like to make use of this data can interpret and manipulate the raw data, which is stored in a tree-like data structure.

Serialization Example

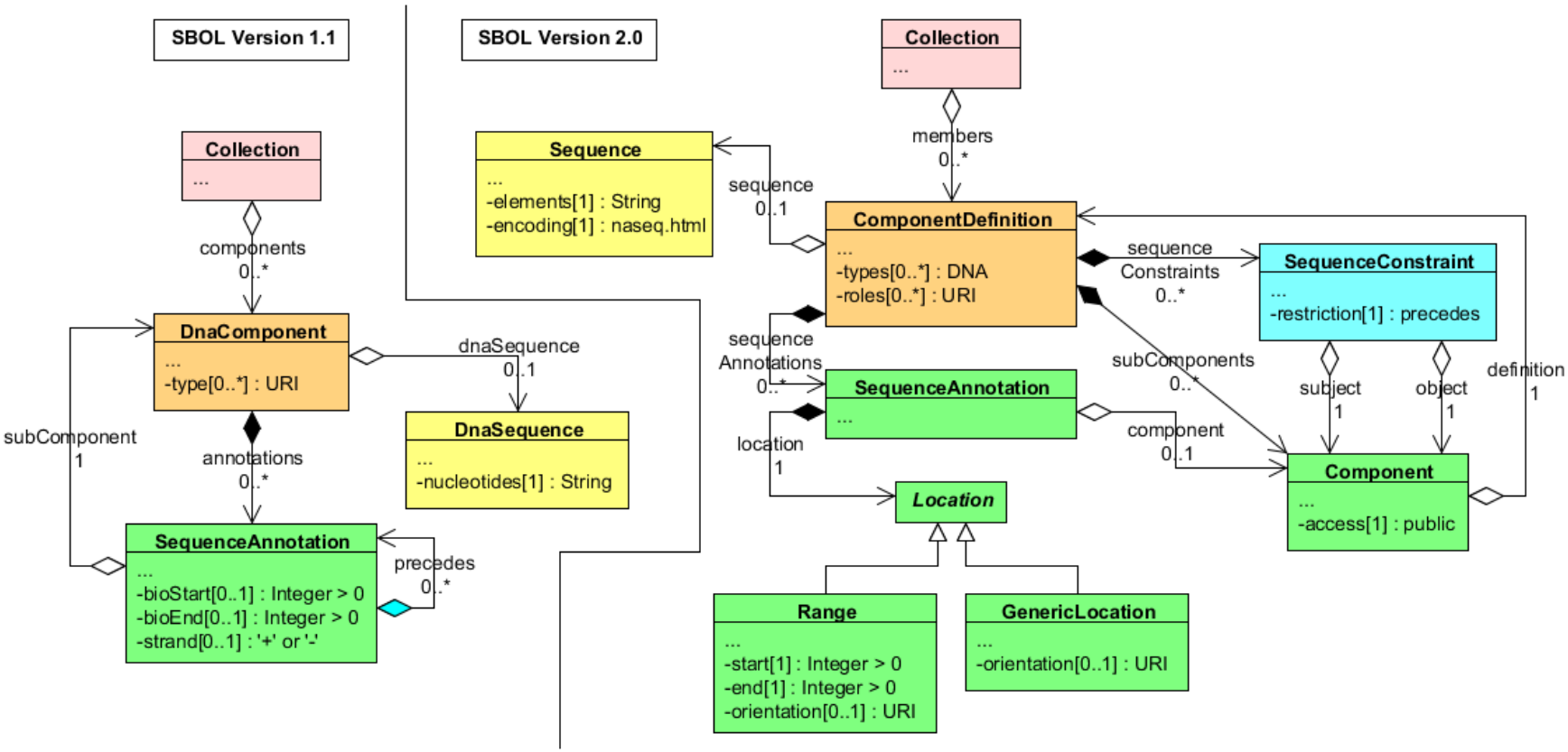
```
<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:myerslab="http:www.async.ece.utah.edu"
  xmlns:sbol="http://sbols.org/v2#">
  <sbol:Collection rdf:about="http://www.async.ece.utah.edu/My_Parts/1.0">
    <dc:title>My_Parts</dc:title>
    <dc:description>These_are_my_parts.</dc:description>
    <sbol:member rdf:resource="http://www.async.ece.utah.edu/pLac/1.0"/>
    ...
  </sbol:Collection>
  <sbol:ModuleDefinition rdf:about="http://www.async.ece.utah.edu/LacI_Inverter/1.0">
    <sbol:role rdf:resource="http://www.async.ece.utah.edu/inverter"/>
    <sbol:functionalComponent>
      <sbol:FunctionalComponent rdf:about="http://www.async.ece.utah.edu/LacI_Inverter/LacI/1.0">
        <sbol:definition rdf:resource="http://www.async.ece.utah.edu/LacI/1.0"/>
        <sbol:access rdf:resource="http://sbols.org/v2#public"/>
        <sbol:direction rdf:resource="http://sbols.org/v2#input"/>
      </sbol:FunctionalComponent>
    </sbol:functionalComponent>
    ...
```

Serialization Example (cont)

```
    <sbol:interaction>
      <sbol:Interaction
           rdf:about="http://www.async.ece.utah.edu/LacI_Inverter/pLac_Repression/1.0">
        <sbol:type rdf:resource="http://identifiers.org/sbo/SBO:0000169"/>
        <myerslab:parameter rdf:resource="http://www.async.ece.utah.edu/Kr_LacI">
        <sbol:participation>
          <sbol:Participation
           rdf:about="http://www.async.ece.utah.edu/LacI_Inverter/pLac_Repression/p1/1.0">
            <sbol:role rdf:resource="http://www.async.ece.utah.edu/repressed"/>
            <sbol:participant
               rdf:resource="http://www.async.ece.utah.edu/LacI_Inverter/pLac/1.0"/>
          </sbol:Participation>
        </sbol:participation>
        <sbol:participation>
          <sbol:Participation
           rdf:about="http://www.async.ece.utah.edu/LacI_Inverter/pLac_Repression/p2/1.0">
            <sbol:role rdf:resource="http://identifiers.org/sbo/SBO:0000020"/>
            <sbol:participant
               rdf:resource="http://www.async.ece.utah.edu/LacI_Inverter/LacI/1.0"/>
          </sbol:Participation>
        </sbol:participation>
      </sbol:Interaction>
    </sbol:interaction>
    ...
  </sbol:ModuleDefinition>
  <myerslab:Parameter rdf:about="http://www.async.ece.utah.edu/Kr_LacI">
    <myerslab:type rdf:resource="http://identifiers.org/sbo/SBO:0000281"/>
    <myerslab:value>0.5</myerslab:value>
  </myerslab:Parameter>
  ...
</rdf:RDF>
```

Support for SBOL 1.1

- `libSBOLj 2.0` eases transition for developers who adopted SBOL 1.1.
  1. `libSBOLj 1.1` is included unmodified to manage SBOL 1.1 data objects.
  2. New file reader automatically detects the SBOL version, and if SBOL 1.1 data is detected, it is automatically converted to an SBOL 2.0 data object.



More Information

- `libSBOLj 2.0` is open source under the Apache 2.0 License.
- `http://sbolstandard.org/libsbol/get-the-libraries/java/`, includes links to:
  - Current development snapshot on GitHub.
  - The latest release, v2.0-beta.
  - The GitHub issue tracker.
  - JavaDocs, a getting started tutorial, and example code.
- Questions about the library can be sent to: `libsbol-team@googlegroups.com`.

Conclusion

- Standards are an important enabler for progress in synthetic biology.
- `libSBOLj 2.0` provides an API to support the adoption of the SBOL 2.0 data model.
- It provides utilities and performance enhancements for compliant URIs.
- Serialization is more efficient and flexible.
- Provides full backward compatibility support for `SBOL 1.1`.

Acknowledgments (libSBOLj 2.0)

Tramy Nguyen (Utah)

Nicholas Roehner (Boston)

Goksel Misirli (Newcastle)

Matthew Pocock (Newcastle)

Ernst Oberortner (DOE JGI)

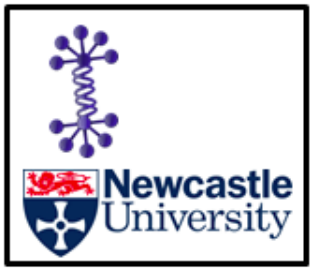
Jacob Beal (Raytheon)

Kevin Clancy (ThermoFisher)

Anil Wipat (Newcastle)

Chris Myers (Utah)

Acknowledgments (SBOL)



Supported by National Science Foundation Grants DBI-1356041 and DBI-1355909, and the Engineering and Physical Sciences Research Council under Grant Number EP/J02175X/1.