

BBF RFC ???: Synthetic Biology Open Language (SBOL) Version 2.0.0

Editors:

Bryan Bartley	<i>University of Washington, USA</i>
Jacob Beal	<i>Raytheon BBN Technologies, USA</i>
Kevin Clancy	<i>ThermoFischer Scientific, USA</i>
Goksel Misirli	<i>Newcastle University, UK</i>
Nicholas Roehner	<i>Boston University, USA</i>

Chair:

Herbert Sauro	<i>University of Washington, USA</i>
---------------	--------------------------------------

editors@sbolstandard.org

Additional authors, by institution:

Ernst Oberortner	<i>DOE Joint Genome Institute, USA</i>
Curtis Madsen, Matthew Pocock, Anil Wipat	<i>Newcastle University, UK</i>
Tramy Nguyen, Zhen Zhang, Chris Myers	<i>University of Utah, USA</i>
John H. Gennari	<i>University of Washington, USA</i>
Michael Bissell	<i>Amyris, Inc., USA</i>

Version 2.0.0

June XXX, 2015



Contents

1 Purpose	4
2 Relation to other BBF RFCs	7
3 Copyright and License Statement	8
4 A Brief History of SBOL	9
5 SBOL Specification Vocabulary	11
5.1 Term Conventions	11
5.2 SBOL Class Names	11
6 Overview of SBOL	13
7 SBOL Data Model	17
7.1 Understanding the UML Diagrams	17
7.2 Naming and Font Conventions	18
7.3 Data Types	18
7.4 Identified	19
7.5 TopLevel	21
7.6 Sequence	22
7.7 ComponentDefinition	23
7.7.1 ComponentInstance	27
7.7.2 Component	28
7.7.3 MapsTo	29
7.7.4 SequenceAnnotation	31
7.7.5 Location	32
7.7.6 SequenceConstraint	35
7.8 Model	37
7.9 ModuleDefinition	38
7.9.1 FunctionalComponent	40
7.9.2 Module	41
7.9.3 Interaction	42
7.9.4 Participation	44
7.10 Collection	44
7.11 SBOL Extension Mechanism	45
7.11.1 Annotating SBOL objects	46
7.11.2 GenericTopLevel	47
8 Data Model Examples	49
9 SBOL RDF Serialization	55
10 Recommended Best Practices	57
10.1 Use of the Version Property	57
10.2 Compliant SBOL Objects	57
10.3 Annotations: Embedded Objects vs. External References	58
10.4 Completeness and Validation	58
10.5 Recommended Ontologies for External Terms	59
References	60
A Validation Rules	61
B Examples of Serialization	71
B.1 PoPS Receiver	71
B.2 Toggle Switch	74

When editing, change the todos that you handle to "Rtodo" ("R" for "resolved"), rather than just deleting them, so that a second person can review

1

Final pass before release must include:
Ensure that all references are defined precisely once (grep 'undefined' and 'multiply' in log)
Make sure all requirements words are properly upper-cased
Spell check
Ensure that TOC links to correct locations

2

1 Purpose

Goksel:I find this paragraph too detailed. Up to the sentence starting with "When designing" is ok. If we really want to give an example, just a TF would be enough without mentioning about microRNAs or antagonists, and keeping it short.

Synthetic biology builds upon the techniques and successes of genetics, molecular biology, and metabolic engineering by applying engineering principles to the design of biological systems. These principles include standardization, modularity, and design abstraction. The field still faces substantial challenges, including long development times, high rates of failure, and poor reproducibility. A common factor of these challenges is the exchange of information about designed systems between laboratories. When designing a synthetic system, synthetic biologists need to exchange information about multiple types of molecules and their planned roles in the design. Often the functional role may be associated with another type of molecule entirely, such as a small chemical, a DNA, an RNA or a Protein molecule. An example is a DNA sequence that is transcribed into a messenger RNA that contains an encoded microRNA binding site, and the messenger RNA in turn being translated into a protein molecule which is a transcription factor binding protein. Functionally the representation of the products of the designed DNA sequence need to describe the role of microRNA binding to the messenger RNA leading to possible degradation, the functional consequence of the transcription factor protein being absent leading to repression of expression of another gene and the kinetic information associated with these different elements so they can be mathematically modeled. The DNA sequence itself is thus one or two steps removed from the functional role of the designed device or circuit.

The *Synthetic Biology Open Language* has been designed as a standard to support synthetic biology, filling a need not satisfied by other pre-existing standards. Previous file formats such as GenBank and Swiss-Prot represent sequence information based upon annotation of sequence features - they do not represent the functional roles or consequences of these sequences. Modelling languages, such as the Systems Biology Markup Language (SBML) [Finney et al. \(2006\)](#) can be used represent biological processes, but is not sufficient to represent the associated nucleotide or amino acid sequences. Synthetic biology needs a structured standard with defined ways on how to represent relevant molecules and their functional roles within the designed system, standardized rules on how such information is encoded in the file and the means to enable exchange of such data between participating laboratories as part of publications.

To help address these challenges, the SBOL introduces a standardized format for the electronic exchange of information describing the structural and functional aspects of biological designs. The standard is designed to support the development of explicit and unambiguous data models of biological designs through the use of a well defined data model on how to represent the biological molecules, and their structural and functional roles in a systematic fashion. The standard further describes rules and best practices on how to include, develop, and populate this format with relevant information of essential design details. SBOL uses existing Semantic Web resources such as biological ontologies and *Universal Resource Identifiers* (URIs) to unambiguously define genetic design elements. The definition of the data model and associated format, the rules on the addition of data within the format and the representation of this in electronic data files are intended to make the SBOL standard a useful means of promoting global data exchange between laboratories and between software programs.

This document presents the second version of SBOL. The previous version 1.1 of the SBOL standard focused on representing the structural aspects of genetic designs. Users of the standard were able to use it to exchange information on DNA designs but could not represent molecules other than DNA or represent the functional aspects of their designs beyond the DNA sequences. To serve as an effective medium for the computational exchange of genetic designs, SBOL must be extended to capture more aspects of a designed system, including both structural and functional information, and the composition of complex structural and functional designs by combining simpler parts. The SBOL 2.0 data model defined in this specification thus extends the prior model to provide for addressing the most pressing needs for expanding SBOL version 1.1, in particular it can:

- represent structural components of a biological design, including DNA, RNA, proteins, small molecules and

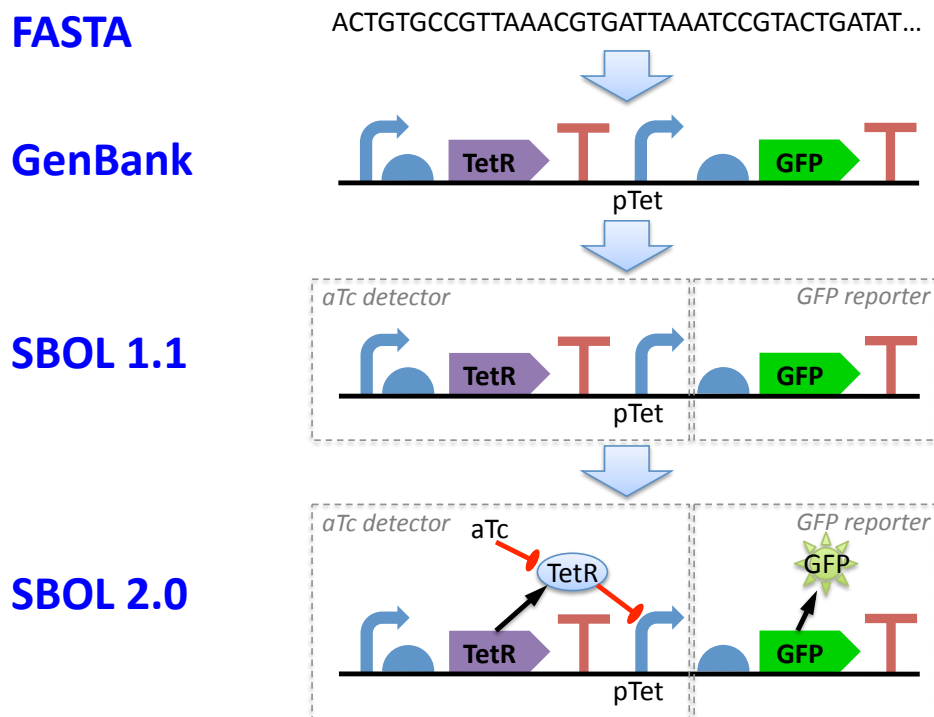


Figure 1: SBOL 2.0 extends prior formats to represent both structure and function of a genetic design in a single model.

other physical components

- describe behavioral aspects of a biological design, the intended or expected interactions and dynamic behavior
- associate structure and function together, so that a single design can be understood both in terms of its structure and its behavior
- support rich annotations of all components, so that data required to describe a design, but not formalized in this specification can be safely exchanged

Taken together, these capabilities allow SBOL sufficient expressiveness to support the description and exchange of hierarchical, modular representations of both the intended structure and function of designed biological systems.

Figure is not referenced in the text and moved to where it is referenced. -CJM. Goksel: We need to redraw the figure with SBOL-Visual icons.

To address the need for functional descriptions in SBOL, the proposed data model adds new classes to provide a firm basis for functional representation in SBOL without going so far as to create a new standard for mathematically modeling biology, as there already exist several established languages for doing so, from the SBML to CellML [Garny et al. \(2008\)](#) and even MatLab [MathWorks \(2015\)](#). Rather, these classes enable users of SBOL to group components that function together, describe the basic qualitative interactions between these components, and document references to standard mathematical models that are external to SBOL and that provide more detailed descriptions of component function. In other words, a module definition gathers together a set of component instantiations, a set of interactions between these component instantiations, and a set of references to external models that are expected to be consistent with the module's interactions.

The SBOL 2.0 specification also adds a number of measures to simplify adoption and validation of compatibility with the standard. First, the specification explicitly incorporates its serialization format into the specification, whereas

SBOL 1.1 used an implicit standard tied to a reference implementation. Second, the specification includes a set of validation rules for determining compatibility of a document with SBOL 2.0, most of which are machine-verifiable. Finally, the specification includes a set of recommended best-practices likely to allow a tool to take best advantage of the standard and other compatible tools.

Care has been taken to ensure that SBOL 2.0 is backward-compatible with SBOL 1.x. The generalization of the data model does mean that many names have changed and thus an SBOL 1.x file is not a valid SBOL 2.0 file. There is, however, a direct mapping from the SBOL 1.x data model into the SBOL 2.0 data model, making it simple to automatically “upgrade” any SBOL 1.x file into an SBOL 2.0 file. Since SBOL 2.0 can encode all data previously encoded in SBOL 1.1, developers are also encouraged to upgrade their SBOL 1.1 compliant software tools to use SBOL 2.0 data objects.

As discussed previously, SBOL 2.0 allows designs to be described beyond the simple annotated DNA sequence offered in SBOL 1.1. Of equal importance in SBOL 2.0 is the explicit provision of mechanisms that allows SBOL to be easily extended. The intent of SBOL is to allow designs of synthetic biological systems to be fully described so that such designs can be reproduced. However SBOL does not currently offer a full catalog of data to allow one to achieve complete reproducibility. For example the proposed standard does not yet include environmental and host context information or details on how the performance of the design is measured. Such details can now be included in SBOL through an explicit extension mechanism. Three scenarios are envisaged for extending SBOL:

- Use of the extension mechanism to include critical information related to the reproducibility of designs. For example, what growth media was used, what temperature were the organisms grown at, when were they harvested, was the DNA integrated into the host genome (if so here), or in a plasmid (what plasmid).
- For tool makers, the extension mechanism allows tool specific information to be added to SBOL. Such information could include tool settings specific to the design that is being loaded, for example what windows should be opened or settings initialized. Tool makers could also include encrypted proprietary information related to the company or client in an extension.
- Non-essential information to reproducibility but nevertheless useful information for many users. There are many cases where a community of users require specific information not available in core of SBOL. Examples include visualization information, how the DNA was assembled, information on evolutionary stability, or specialist modeling information.

The extension mechanism is therefore a critical part of SBOL 2.0 and will allow others in the community to incorporate either their own requirements for data into SBOL or contribute to community efforts to expand the scope of SBOL.

The SBOL standard has been developed in collaboration between both “wet” bench scientists and “dry” scientific modelers and tool designers active within the synthetic biology community. As with the earlier SBOL 1.1 standard, this community (open for any practitioner to join) has met to discuss, argue and agree upon needs that the SBOL standard should address. This information has then been used by developers within our community to design, develop, and test a specification of the standard. The specification has been tested by the community through several iterations for the ability to represent a wide range of synthetic biology design projects, as well as, the ability to share designs between different laboratories. The standard has also been used to develop software tools that employ the standard for developing and sharing synthetic design projects. The publication of this specification is intended to make these capabilities more widely accessible to the community of potential developers and users.

2 Relation to other BBF RFCs

Just added this section; needs review -JSB Looks fine, just need the new number -CJM

BBF RFC ??? replaces BBF RFC 87 (the SBOL 1.1 standard).

BBF RFC ??? updates BBF RFC 30 (RDF-based framework for synthetic biology data), as it proposes a standard conforming to BBF RFC 30.

BBF RFC ??? also implicitly supersedes the previously replaced BBF RFC 84 (SBOL 1.0, replaced by BBF RFC 87) and BBF RFC 31 (PoBoL, replaced by BBF RFC 84).

3 Copyright and License Statement

Copyright (C) The BioBricks Foundation and all authors listed on this BBF RFC. This work is made available under the Creative Commons Attribution 4.0 International Public License. To view a copy of this license visit <https://creativecommons.org/licenses/by/4.0/>.

In addition to the listed authors, the following people are specifically recognized as additional contributors sharing in the copyright (alphabetically by institution): Douglas Densmore (Boston University, USA), Jacqueline Quinn (Google, USA), Guy-Bart Stan (Imperial College London, UK)

4 A Brief History of SBOL

The text below needs thorough review by the community. We are certain that there are many errors, including missing people. Please send input to fix these errors. Reviewed: JSB

In early 2006, Microsoft issued a call for proposals in the field of computational synthetic biology. A proposal was submitted from UW with the aim to kickstart an effort to develop exchange standards for designs in the new field of synthetic biology. Along with five other groups, the UW group was successful in securing a modest grant. Part of the funds were used to fund the initial standards meeting held in Seattle on April 26-27, 2008. The organizers of the initial meeting were Herbert Sauro, Sean Sleight and Deepak Chandran. The meeting included talks by Raik Gruenberg, Kim de Mora from the Jason Kelly lab, John Cumbers, Christopher Anderson, Mac Cowell, Jason Morrison, Jean Peccoud, Ralph Santos, Andrew Milar, Vincent Rouilly, Mike Hucka, Michael Blinov, Lucian Smith, Sarah Richardson, Guillermo Rodrigo, Jonathan Goler, and last but not least Michal Galdzicki. Michal was to go on and lead the development of PoBol, as it was then called. Mike's early efforts were instrumental in making SBOL the success it is today. He organized annual workshops from 2008 to 2011 and kept the idea of developing a standard alive. These were held at the Synthetic Biology Data Exchange Working Group meeting at Stanford on July 26, 2009 and Anaheim, CA on June 13, 2010. Included at the Anaheim meeting were Chandran, Densmore, Dmytriv, Galdzicki, Ham, Rodriguez, Peccoud, Sauro, and Stan. The original SBOL 1.0 was developed at these early meetings through Michal's efforts, together with the small group of other dedicated researchers. It was also at the Anaheim meeting that a decision was made to write a letter to Nature Biotechnology highlighting the issue of reproducibility in synthetic biology. This letter was initiated by Jean Peccoud and submitted by participants of the Anaheim meeting.

An important meeting was held in 2011 at Blacksburg, Virginia on January 7-10, 2011 where new members joined the group resulting in 17 individuals at the meeting. New members included Cesar Rodriguez, Mandy Wilson, Guy-Bart Stan, Chris Myers, and Nicholas Roehner, and the overall pace of development quickened.

At a meeting in San Diego in June 2011, the SBOL Developers Group was officially established, rules of governance were established, and the first SBOL editors were elected: Mike Galdzicki, Cesar Rodriguez, and Mandy Wilson. At this time, Allan Kuchinsky, a research scientist at Agilent, joined the effort, and he was able to obtain some resources to begin development on what was to become libSBOLj. Kevin Clancy from LifeTechnologies also joined at this time, as well as Anil Wipat, Matthew Pocock, and Goksel Misirli from Newcastle University, and Jacob Beal, Aaron Adler, and Fusun Yaman Sirin from Raytheon BBN Technologies. In October 2011, SBOL 1.0 was officially released. At our next meeting in Seattle in January 2012, Herbert Sauro was elected the SBOL Chair, and two new editors were added: Matthew Pocock and Ernst Oberortner. At this meeting, the first data exchange between software tools using SBOL was conducted when a design was passed from Newcastle University's VirtualParts Repository to Boston University's Eugene tool, and finally to University of Utah's iBioSim tool.

In March 2012, SBOL 1.1 was released, the version that this document replaces. SBOL 1.1 did not make any major change, but provided a number of small adjustments and clarifications on the particulars of SBOL's intended use and implementation, particularly around the annotation of sequences and their locational relations. The 8th SBOL workshop was held in November 2012 at Boston University, and the major topic of discussion was the next version of SBOL. SBOL 1.1 is limited to describing hierarchical DNA sequences. Several extensions were discussed at this meeting, such as a means to describe genetic regulation, which later became the [Interaction](#) class, and a means to group components, which later became the [Module](#) class. In April 2013, at the 9th SBOL workshop at Newcastle University, the framework for SBOL 2.0 was agreed upon. Nicholas Roehner, Matthew Pocock, and Ernst Oberortner then began work on a draft proposal for SBOL 2.0. In January 2014 at the 10th SBOL workshop, this draft was discussed and many refinements were debated and approved. Another important decision at this meeting was that SBOL should be part of the COMBINE community of standards (www.co.mbine.org).

In the Spring and Summer of 2014, several important events occurred. In April, several SBOL representatives attended Harmony in Manchester UK to discuss joining the COMBINE community, which was approved by both sides shortly thereafter. In May, Herbert Sauro, John Gennari, and Chris Myers received a grant from the National

Science Foundation to support SBOL (this document and the supporting software are due in no small part to this support).

In June, a description and our initial, multi-institutional demonstration of the use of SBOL 1.1 was published in Nature Biotechnology [Galdzicki et al. \(2014\)](#). In July, Nicholas Roehner presented a proposal for the next version of SBOL at the SEED Conference in Los Angeles [Roehner et al. \(2015\)](#). Finally, in August 2014, the SBOL community attended their first COMBINE workshop as members of the COMBINE community. At this meeting, many of the final details of SBOL 2.0 were discussed, and the data model presented here is essentially the result of this meeting.

At the Harmony meeting in April 2015 in Wittenberg, Germany, the work on this specification began in earnest. The key contributors at this meeting and the previous one were: Bryan Bartley (University of Washington), Jacob Beal (Raytheon BBN Technologies), Kevin Clancy (ThermoFischer), Bryan Der (MIT), John Gennari (University of Washington), Curtis Madsen (Newcastle University), Goksel Misirli (Newcastle University), Chris J. Myers (University of Utah), Tramy Nguyen (University of Utah), Matthew Pocock (Newcastle University and Turing Ate My Hamster LTD), Jackie Quinn (Google), Nicholas Roehner (Boston University), Herbert M. Sauro (University of Washington), Anil Wipat (Newcastle University), and Zhen Zhang (University of Utah).

5 SBOL Specification Vocabulary

5.1 Term Conventions

This document indicates requirement levels using the controlled vocabulary specified in IETF RFC 2119 and reiterated in BBF RFC 0. In particular, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

- The words "MUST", "REQUIRED", or "SHALL" mean that the item is an absolute requirement.
- The phrases "MUST NOT" or "SHALL NOT" mean that the item is an absolute prohibition.
- The word "SHOULD" or the adjective "RECOMMENDED" mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- The phrases "SHOULD NOT" or "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- The word "MAY" or the adjective "OPTIONAL" mean that an item is truly optional.

5.2 SBOL Class Names

SBOL defines the following “top-level” and dependent classes:

Collection: Represents a user-defined container for organizing a group of SBOL objects.

ComponentDefinition: Describes the structure of designed entities, such as DNA, RNA, and proteins, as well as other entities they interact with, such as small molecules or environmental properties.

- **Component:** Represents a specific occurrence or instance of a single entity within the design of a more complex component. Each **Component** is associated with a **ComponentDefinition**, and there may be many different instances at different locations in a design that share the same definition.
- **Location:** Specifies the base coordinates and orientation of a genetic feature on a DNA or RNA molecule or a residue or site on another sequential macromolecule such as a protein.
- **SequenceAnnotation:** Describes the **Location** of a notable sub-sequence found within the **Sequence** linked to a **ComponentDefinition**, with an optional link to a **Component**.
- **SequenceConstraint:** Describes the relative spatial position and orientation of two **Component** objects that are contained within the same **ComponentDefinition**.

GenericTopLevel: Represents a data container that can contain custom data added by user applications.

Model: Links an SBOL representation of biological components and their interactions to quantitative or qualitative computational models that may be used to predict the functional behavior of a biological design.

ModuleDefinition: Describes a “system” design as a collection of biological components and their functional relationships.

- **FunctionalComponent:** Represents a specific occurrence or instance of an **ComponentDefinition** within a **ModuleDefinition**. Exactly like a **Component**, except that it can be associated with information about its context of use in the **Module**, rather than in the context of a containing **ComponentDefinition**.
- **Interaction:** Describes a functional relationship between biological entities, such as regulatory activation or repression, or a biological process such as transcription or translation.

- **MapsTo**: When a design (**ComponentDefinition** or **ModuleDefinition**) includes another design as a substructure, the larger design may need to refer to a **ComponentInstance** from the sub-design. In this case, a referencing **ComponentInstance** needs to be created in the design and a **MapsTo** is added to the instance for the sub-design, which associates the original and the referencing instance.
- **Module**: Represents a specific occurrence or instance of a sub-system within a larger design. Each **Module** is associated with a **ModuleDefinition**, and there may be many different instances at different locations in a design that share the same definition.
- **Participation**: Describes the role that a **Component** plays in an **Interaction**. For example, a transcription factor might participate in an **Interaction** as a repressor or as an activator.

Sequence: Represents a contiguous series of monomers in a macromolecular polymer such as DNA, RNA, or protein.

6 Overview of SBOL

Synthetic biology designs can be described using:

- Structural terms, e.g., a set of annotated sequences or information about the chemical makeup of components.
- Functional terms, e.g., the way that components might interact with each other and the overall behavior of a design.

In broad strokes, the prior SBOL 1.1 standard focused on physical, structural information, whereas SBOL 2.0 extends to also include functional aspects. The physical information about a designed genetic construct includes the order of its constituents and their descriptions. The exact locations of these constituents and their sequences allow genetic constructs to be defined unambiguously, and reused in other designs. SBOL 2.0 extends SBOL 1.1 in several ways: it extends physical descriptions to include entities beyond DNA sequences, and it allows for functional descriptions of the design.

As an example, consider the design of an expression cassette, such as the one found in the plasmid pUC18 [Norlander et al. \(1983\)](#). This device is designed to detect successful versus unsuccessful molecular cloning. As an overall system, the device is designed to grow either blue-colored (unsuccessful) or white-colored (successful) colonies in the presence of IPTG and the chemical X-gal. Internally, the device has a number of parts, including a promoter, the lac repressor binding site, and the lacZ coding sequence. These parts have specific component-level interactions with IPTG and X-gal, as well as native host gene products, transcriptional machinery and translational machinery that collectively cause the desired system-level behavior.

Knowledge of how such a device functions within the context of a host and how it might be adapted to new experimental applications has generally been passed on through working with fellow scientists or reading articles in papers and books. But there was no systematic way to communicate the integration of sequences with functional designs, so users typically had to look in many different places to develop an understanding of a system. The SBOL 2.0 standard allows designers to describe these functional characteristics and connect them to the physical parts and sequences that make up the design.

SBOL 2.0 includes two main classes that match the structural/functional distinction above:

- The [ComponentDefinition](#) object describes the physical aspects of the designed system, such as the DNA or RNA sequences and the physical relationships among sub-components, such as one sequence containing another as a sub-sequence.
- The [ModuleDefinition](#) object describes interactions of the designed system, such as specific binding relationships and repression and activation relationships.

[Figure 2](#) shows a simplified view of these classes, as well as other helper classes in SBOL. To continue with the pUC18 example, the description would begin with a top-level [ModuleDefinition](#). The [ModuleDefinition](#) specifies the structural elements that make up the cassette by referencing a number of [ComponentDefinition](#) objects. These would include the DNA component for the promoter and the small molecule component for IPTG, for example. The [ComponentDefinition](#) objects can be organized hierarchically. For example, the plasmid [ComponentDefinition](#) may reference [ComponentDefinitions](#) for the promoter, coding sequence, etc. Each [ComponentDefinition](#) object can also include the actual [Sequence](#) information (if available), as well as [SequenceAnnotation](#) objects that identify the locations of the promoters, coding sequences, etc., on the [Sequence](#). In order to specify functional information, the [ModuleDefinition](#) can specify [Interaction](#) objects that describe any qualitative relationships among components, such as how IPTG and X-gal interact with the gene products. Finally, a [ModuleDefinition](#) object can point to a [Model](#) object that provides a reference to a complete quantitative model using a language such as SBML, CellML, Matlab, etc. Finally, all the elements of the genetic design can be grouped together within a [Collection](#).

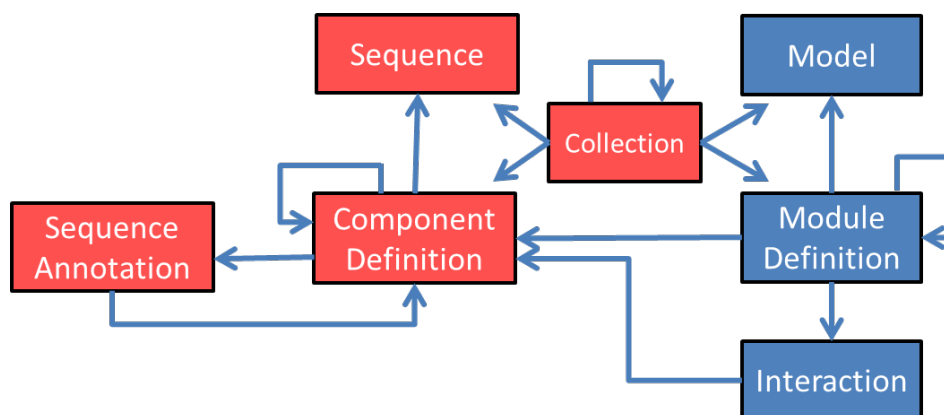


Figure 2: Main classes of information represented by the SBOL standard, and their relationships. Red boxes are classes from the SBOL 1.1 that focused on structure, whereas blue classes are some of the new classes that support the functional aspects of designs.

Whereas [Figure 2](#) provides a broad overview of SBOL, [Figure 3](#) provides a detailed, implementation-level overview of the class structure for the SBOL 2.0 data model. This figure relies on the semantics of the *Unified Modeling Language* (UML), which will be presented in more detail in the next section. [Figure 3](#) distinguishes between *top level* classes, in green, and other supporting classes (note that [Figure 2](#) also includes all of the top level classes). In [Figure 3](#), dashed arcs represent "refersTo", whereas a solid arrow represents ownership. In UML, the meaning of ownership is that if a parent class is deleted, so are all of its owned children. Thus, a [Collection](#) does not own its [ComponentDefinition](#) objects, because these can stand on their own. All of the supporting classes (in orange) must be owned by some top-level class, directly or indirectly.

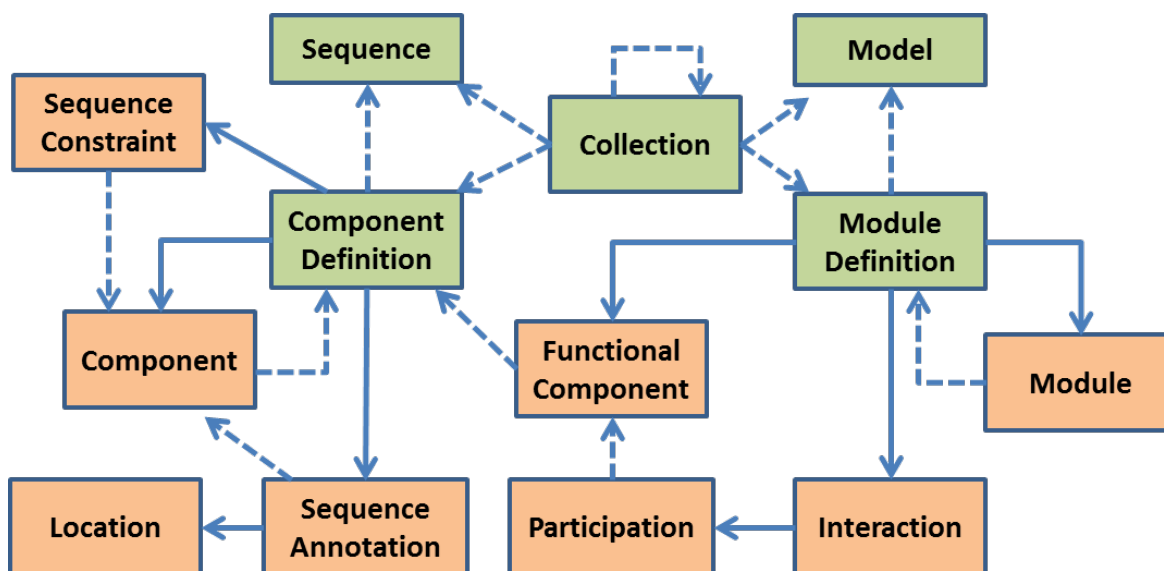


Figure 3: Main classes of information represented by the SBOL 2.0 standard, and their relationships. Green boxes are “top level” classes, while the other classes are in support of these classes. Solid arrows indicates ownership, whereas a dashed arrow indicates that one class refers to an object of another class.

The second figure additionally depicts the connection of the functional side (modules) to the physical side (components). This is accomplished via the class `FunctionalComponent`. This class allows modules to own their components instances, and yet also allows the physical descriptions (in `ComponentDefinitions`) to stand on their

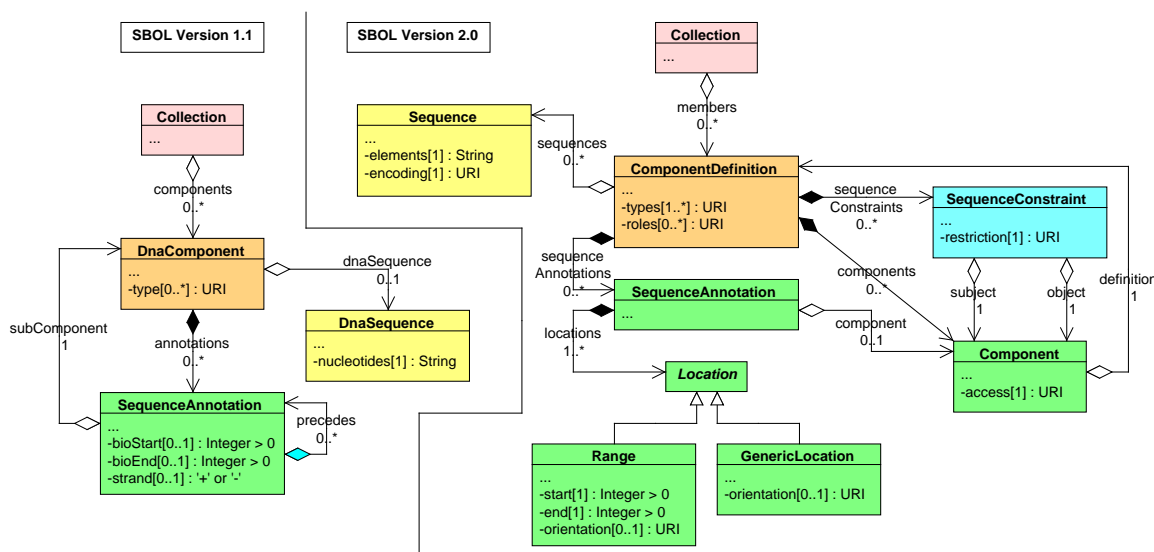


Figure 4: The mapping from the SBOL 1.1 data model to the SBOL 2.0 data model.

own. In a similar manner, the ability to have hierarchies of either functional or physical components shown in Figure 2 must be broken apart, so that sub-components can be used in multiple functional modules or multiple physical components. Thus, instead of the arc from `ModuleDefinition` to itself as in Figure 2, our implementation actually divides this notion into two classes, `ModuleDefinition` and `Module`. Therefore, a `ModuleDefinition` does not own the `ModuleDefinitions` that it uses, but instead it refers to them using the `Module` objects that it does own. The identical relationship occurs on the physical side with `ComponentDefinition` and `Component`. Finally, SBOL 2.0 provides a few other additional helper classes such as `Location`, which generalizes the positioning information from SBOL 1.1 to allow discontinuous ranges and cuts to be annotated, and `SequenceConstraint`, which generalizes the relative positioning information among `Components`. There is also `Participations`, which allow `Interaction` objects to specify the roles of their participants while referencing the `FunctionalComponents`, so that these can stand on their own. Finally, there is the `MapsTo` class (not shown) that enables connections to be made between `Components` and `FunctionalComponents` at various levels of the design hierarchy. The next section provides complete definitions and details for all of these classes.

Added mapping of objects from 1.1 to 2.0 using the diagram from the libSBOLj paper. Needs review. Also not entirely sure this is the right place. Should we move it to its own section later? -CJM
Yes please. Since this mapping figure really is drawn as UML, it needs to appear after section 7.1 "Understanding UML Diagrams". -jhg

Figure Figure 4 depicts the mapping between SBOL 1.1 data objects and SBOL 2.0 data objects. Collections of DNA components map to Collections of ComponentDefinitions, among other top level SBOL objects. DnaComponents map to ComponentDefinitions of type DNA. DnaSequences map to Sequences using the IUPAC encoding for nucleotide sequences. SequenceAnnotations with precise start and end positions are mapped to SequenceAnnotations with Range Locations, while SequenceAnnotations with imprecise positions are mapped to SequenceAnnotations with GenericLocations. Each Sequence Annotation also maps to a Component, which in SBOL 2.0 represents the instantiation or usage of a given ComponentDefinition. Finally, precedes relationships map to SequenceConstraints that specify precedes restrictions.

There is one final, critical element of SBOL 2.0: its extension mechanism. This extension mechanism enables the storage of application specific information within an SBOL document. It is also intended to support the prototyping

of data representations whose format is not yet a matter of consensus within the community. In particular, each SBOL entity can be annotated using the *Resource Description Framework* (RDF). Moreover, application specific entities in the form of RDF documents can be included as [GenericTopLevel](#) entities. SBOL libraries make these annotations and entities available to tools as generic properties and objects that are preserved during subsequent read and write operations.

1
2
3
4
5

7 SBOL Data Model

Ensure that no UML figures have labels conflicting with their arrows

In this section, we describe the types of biological design data that can belong to an SBOL document and the relationships between these data types. The SBOL data model is specified using Unified Modeling Language (UML) 2.0 diagrams (OMG 2005). Subsections [Section 7.1](#), [Section 7.2](#), [Section 7.3](#) review the basics of UML diagrams and explain the naming conventions and generic data types used in this specification. The remaining sections then describe the SBOL data model in detail. Complete SBOL examples and best practices when using the standard can be found in [Section 8](#) and [Section 10](#), respectively.

7.1 Understanding the UML Diagrams

The types of biological design data modeled by SBOL are commonly referred to as *classes*, especially when discussing the details of software implementation. Each SBOL class can be instantiated by many SBOL objects. These objects may contain data that differ in content, but they **MUST** agree on the type and form of their data as dictated by their common class. Classes are represented in UML diagrams as rectangles labeled at the top with class names.

Classes may be connected to other classes by association properties, which are represented in UML diagrams as arrows. These arrows are labeled with data cardinalities in order to indicate how many values a given association property may possess (see below). The remaining (non-association) properties of a class are listed below its name. Each of the latter properties is labeled with its data type and cardinality.

In the case of an association property, the class from which the arrow originates is the owner of the association property. A diamond at the origin of the arrow indicates the type of association. Open-faced diamonds indicate shared aggregation, in which the owner of the association property exists independently of its value. In the SBOL data model, the value of an association property **MUST** be a URI or set of URIs that refer to SBOL objects belonging to the class at the tip of the arrow.

By contrast, filled diamonds indicate composite aggregation, also known as a part-whole relationship, in which the value of the association property **MUST NOT** exist independently of its owner. In addition, in the SBOL data model, it is **REQUIRED** that the value of each composite aggregation property is a unique SBOL object (that is, not the value for more than one such property). Note that in all cases, composite aggregation is used in such a way that there should be no duplication of such objects.

All SBOL properties are labeled with one of several restrictions on data cardinality. These are:

- 1 - required, one: there must be exactly one value for this property.
- 0...1 - optional: there may be a single value for this property, or it may be absent.
- 0...* - unbounded: there may be any number of values for this property, including none.
- 1...* - required, unbounded: there may be any number of values for this property, as long as there is at least one.
- $n \dots *$ - at least: there must be at least n values for this property.

Finally, classes can inherit the properties of other classes. Inheritance relationships are represented in UML diagrams as open-faced, triangular arrows that point from the inheriting class to the inherited class. Some classes in the SBOL data model cannot be instantiated as objects and exist only to group common properties for inheritance. These classes have italicized names and are known as abstract classes.

7.2 Naming and Font Conventions

SBOL classes are named using upper "camel case," meaning that each word is capitalized and all words are run together without spaces, e.g. **Identified**, **SequenceAnnotation**. Properties, on the other hand, are named using lower camel case, meaning that they begin lowercase (e.g., **identity**) but if they consist of multiple words, all words after the first begin with an uppercase letter (e.g., **persistentIdentity**).

Within the SBOL data model, each property is given a singular or plural name in accordance with its data cardinalities. The forms of these names follow the usual rules of English grammar. For example, **SequenceAnnotation** is the singular form of **SequenceAnnotations**.

SBOL properties are always given singular names, however, when SBOL objects are serialized (using *Resource Description Framework* (RDF) as described in [Section 9](#)). This is because the SBOL data model does not contain classes that correspond directly to the RDF elements that group other elements into ordered or unordered sets. Consequently, if an SBOL property has multiple values, then it is serialized as multiple property entries, each with a singular name and a single value. For example, if an SBOL property has five values, then its serialization contains five RDF triples, each with a singular predicate name and one of the five values as its object.

Lastly, font color is used in the body text of this specification to indicate whether a class or property is defined externally or within the SBOL data model. In particular, if a class or property name is written in a blue font, then it is defined by SBOL. If it is written in a bold font, then it is defined externally.

7.3 Data Types

When SBOL use simple "primitive" data types such as strings or integers, these are defined as the following specific formal types:

- String: <http://www.w3.org/TR/xmlschema11-2/#string>
Example: "LacI coding sequence"
- Integer: <http://www.w3.org/TR/xmlschema11-2/#integer>
Example: 3
- Double: <http://www.w3.org/TR/xmlschema11-2/#double>
Example: 3.14159
- Boolean: <http://www.w3.org/TR/xmlschema11-2/#boolean>
Example: true

The term **literal** is used to denote an object that can be any of the four types listed above. In addition to the simple types listed above, SBOL also uses objects with types *uniform resource identifier* (URI) and *XML qualified name* (QName):

- URI: <http://www.w3.org/TR/xmlschema11-2/#anyURI>
Example: http://www.partsregistry.org/Part:BBa_J23119
- QName: <http://www.w3.org/TR/xmlschema11-2/#QName>
Example: **myapp:Datasheet** where **myapp**="http://www.myapp.org/" namespace.

Note that, in compliance with RDF standards, URIs are generally serialized using an **rdf:resource** property, e.g.: **rdf:resource="http://www.partsregistry.org/Part:BBa_J23119"**

It is important to realize that in RDF, a URI may or may not be a resolvable URL (web address). A URI is always a globally unique identifier within a structured namespace. In some cases, that name is also a reference to (or within) a document, and in some cases that document can also be retrieved (e.g., using a web browser).

7.4 Identified

Put a small concrete example for each toplevel, in the style of the mapsTo diagram

All SBOL-defined classes are directly or indirectly derived from the **Identified** abstract class. This inheritance means that all SBOL objects are uniquely identified using **URIs** that uniquely refer to these objects within an SBOL document or at locations on the World Wide Web.

As shown in Figure 5, the **Identified** class includes the following properties: **identity**, **persistentIdentity**, **version**, **wasDerivedFrom**, **name**, **description**, and **annotations**. The latter property is described separately in Section 7.11.

When an SBOL resource reference takes the form of a URI, that URI may either be the value of an **identity** property or the value of a **persistentIdentity** property. If the URI is equal to the value of an **identity** property, then it is guaranteed to be unique, and it refers to precisely one SBOL object with that identity value. If the URI is equal to the value of a **persistentIdentity** property, then it may refer to multiple SBOL objects that are different “versions” of each other. These objects SHOULD be compared to one another to determine which single object the URI should resolve to (normally the most recent version - see Section 7.4). Throughout this document, when a URI is used to refer to an SBOL object, it could fall into either of these cases.

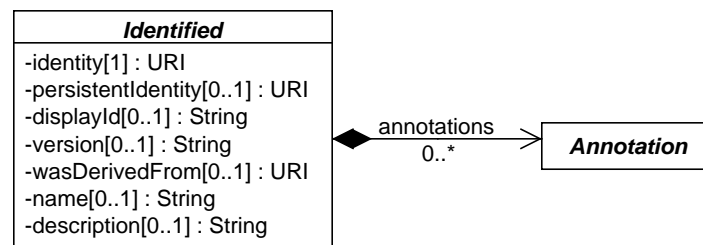


Figure 5: Diagram of the **Identified** abstract class and its associated properties

The **identity** property

The **identity** property is REQUIRED by all **Identified** objects and has a data type of **URI**. A given **Identified** object's **identity** **URI** MUST be globally unique among all other **identity** **URIs**. It is also highly RECOMMENDED that the URI structure follows the recommended best practices for compliant **URIs** specified in Section 10.2.

Although most SBOL properties are defined by SBOL and serialized with its namespace, the **identity** property is defined by the analogous RDF **about** property and is serialized with the RDF namespace as follows:

<http://www.w3.org/1999/02/22-rdf-syntax-ns#about>.

The use of **about** is expressly for the purpose of making SBOL compliant with pre-existing standards: when you see **about** in an SBOL document, you should interpret it as meaning **identity**.

The **persistentIdentity** property

The **persistentIdentity** property is OPTIONAL and has a data type of **URI**. This **URI** serves to uniquely refer to a set of SBOL objects that are different versions of each other.

An **Identified** object MUST be referred to using either its **identity** **URI** or its **persistentIdentity** **URI**.

The `displayId` property

The `displayId` property is an OPTIONAL identifier with a data type of `String`. This property is intended to be an intermediate between `name` and `identity` that is machine-readable, but more human-readable than the full `URI` of an `identity`.

If the `displayId` property is used, then its `String` value SHOULD be locally unique (global uniqueness is not required) and MUST be composed of only alphanumeric or underscore characters and MUST NOT begin with a digit.

The `version` property

The `version` property is OPTIONAL and has a data type of `String`. This property can be used to compare two SBOL objects with the same `persistentIdentity`.

If the `version` property is used, then it is RECOMMENDED that version numbering should follow the conventions of semantic versioning (<http://semver.org/>), particularly as implemented by Maven (<http://maven.apache.org/>). This convention represents versions as sequences of numbers and qualifiers that are separated by the characters “.” and “-” and are compared in lexicographical order (for example, 1 < 1.3.1 < 2.0-beta). For a full explanation, see the linked resources.

The `wasDerivedFrom` property

The `wasDerivedFrom` property is OPTIONAL and has a data type of `URI`. An SBOL object with this property refers to another SBOL object or non-SBOL resource from which this object was derived.

If the `wasDerivedFrom` property of an SBOL object *A* that refers to an SBOL object *B* has an identical `persistentIdentity`, and both *A* and *B* have a `version`, then the `version` of *B* MUST precede that of *A*. In addition, an SBOL object MUST NOT refer to itself via its own `wasDerivedFrom` property or form a cyclical chain of references via its `wasDerivedFrom` property and those of other SBOL objects. For example, the reference chain “*A* was derived from *B* and *B* was derived from *A*” is cyclical.

The `name` property

The `name` property is OPTIONAL and has a data type of `String`. This property is intended to be displayed to a human when visualizing an `Identified` object.

If an `Identified` object lacks a name, then software tools SHOULD instead display the object’s `displayId` or `identity`. It is RECOMMENDED that software tools give users the ability to switch perspectives between `name` properties that are human-readable and `displayId` properties that are less human-readable, but are more likely to be unique.

The `description` property

The `description` property is OPTIONAL and has a data type of `String`. This property is intended to contain a more thorough text description of an `Identified` object.

The `annotations` property

The `annotations` property is OPTIONAL and MAY specify a set of `Annotation` objects that are contained by the `Identified` object. `Annotation` objects are described in more detail in Section [Section 7.11.1](#).

Serialization

No complete serialization is defined for `Identified`, since this class is only used indirectly through its child classes. Any such child class, however, has the following form for serializing properties inherited from `Identified`, where `CLASS_NAME` is replaced by the name of the class:

```

<?xml version="1.0" ?>
<rdf:RDF xmlns:pr="http://partsregistry.org" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.
  org/dc/terms/" xmlns:prov="http://www.w3.org/ns/prov#" xmlns:sbol="http://sbols.org/v2#">
  <sbol:CLASS_NAME rdf:about="...">
    zero or one <sbol:persistentIdentity rdf:resource="..."> element
    zero or one <sbol:displayId>...</sbol:displayId> element
    zero or one <sbol:version>...</sbol:version> element
    zero or one <prov:wasDerivedFrom rdf:resource="..."> element
    zero or one <dcterms:title>...</dcterms:title> element
    zero or one <dcterms:description>...</dcterms:description> element
    ...
  </sbol:CLASS_NAME>
  ...
</rdf:RDF>

```

Note that several of the properties are not in the `sbol` namespace, but are mapped to standardized terms defined elsewhere:

- `identity` is serialized as `rdf:about`
- `wasDerivedFrom` is serialized as `prov:wasDerivedFrom`
- `name` is serialized as `dcterms:title`
- `description` is serialized as `dcterms:description`

7.5 TopLevel

`TopLevel` is an abstract class that is extended by any `Identified` class that can be found at the top level of an SBOL document or file. In other words, `TopLevel` objects are not nested inside any other object via a composite aggregation or black diamond arrow association property. Instead of nesting, composite `TopLevel` objects refer to subordinate `TopLevel` objects by their URIs using shared aggregation or white diamond arrow association properties. The `TopLevel` classes defined in this specification are `Sequence`, `ComponentDefinition`, `Model`, `ModuleDefinition`, `Collection`, and `GenericTopLevel` (Figure 6).

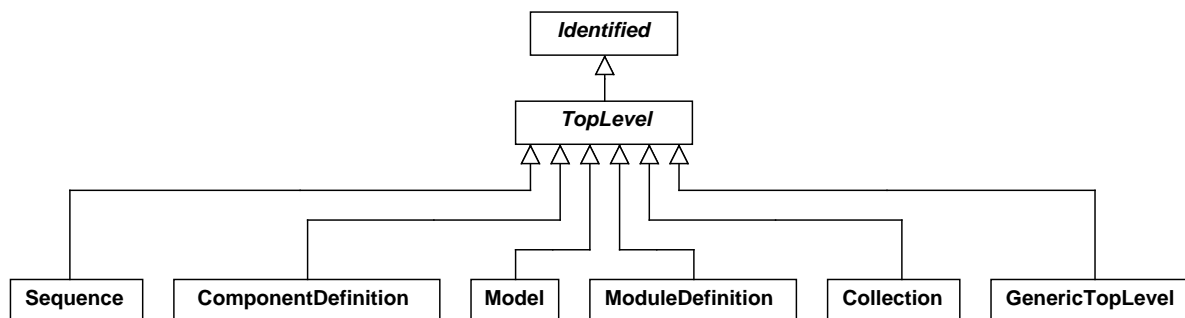


Figure 6: Classes that inherit from the `TopLevel` abstract class.

Serialization

No serialization is defined for `TopLevel`, since this class has no properties of its own and is only used indirectly through its child classes. All `TopLevel` classes are serialized one level beneath the RDF document root.

7.6 Sequence

The purpose of the [Sequence](#) class is to represent the primary structure of a [ComponentDefinition](#) object and the manner in which it is encoded. This representation is accomplished by means of the [elements](#) property and [encoding](#) property (Figure 7).

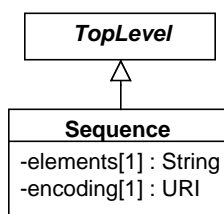


Figure 7: Diagram of the [Sequence](#) class and its associated properties.

The [elements](#) property

The [elements](#) property is a REQUIRED [String](#) of characters that represents the constituents of a biological or chemical molecule. For example, these characters could represent the nucleotide bases of a molecule of DNA, the amino acid residues of a protein, or the atoms and chemical bonds of a small molecule.

The [encoding](#) property

The [encoding](#) property is REQUIRED and has a data type of [URI](#). This property is used to indicate how the [elements](#) property of a [Sequence](#) MUST be formed and interpreted.

For example, the [elements](#) property of a [Sequence](#) with an IUPAC DNA encoding property MUST contain characters that represent nucleotide bases, such as a, t, c, and g. The [elements](#) property of a [Sequence](#) with a Simplified Molecular-Input Line-Entry System (SMILES) encoding, on the other hand, MUST contain characters that represent atoms and chemical bonds, such as C, N, O, and =.

Table 1 provides a list of RECOMMENDED URIs for the [encoding](#) property. The terms in Table 1 are organized by the type of [ComponentDefinition](#) (see Table 2) that typically refer to a [Sequence](#) with such an [encoding](#). When the [encoding](#) of a [Sequence](#) is well described by one of the URIs in Table 1, it MUST use that URI for this property.

revisit for next version of the spec. We looked at EDAM on NCBO as a nice example that integrates many ontologies in biology. This could be a possible way to simplify to one ontology source rather than many as a best practice – KC

Encoding	URI	ComponentDefinition Type
IUPAC DNA, RNA	http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html	DNA, RNA
IUPAC Protein	http://www.chem.qmul.ac.uk/iupac/AminoAcid/	Protein
SMILES	http://www.opensmiles.org/opensmiles.html	SmallMolecule

Table 1: RECOMMENDED URIs for specifying the [encoding](#) property of a [Sequence](#), organized by the type of [ComponentDefinition](#) (see Table 2) that typically refer to a [Sequence](#) with such an [encoding](#).

Serialization

The serialization of a [Sequence](#) MUST have the following form:

```

<sbol:Sequence rdf:about="...">
  ... properties inherited from identified ...
  one <sbol:elements>...</sbol:elements> element
  one <sbol:encoding rdf:resource="..."> element
</sbol:Sequence>

```

The example below shows the serialization of the [Sequence](#) for a promoter. The nucleotide bases of the [Sequence](#) are serialized as the [String](#) value of its [elements](#) property, while its IUPAC DNA encoding is serialized as the [URI](#) value of its [encoding](#) property.

```

<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:prov="http://www.
w3.org/ns/prov#" xmlns:sbol="http://sbols.org/v2#">
  <sbol:Sequence rdf:about="http://partsregistry.org/seq/BBa_J23119">
    <sbol:persistentIdentity rdf:resource="http://partsregistry.org/seq/BBa_J23119"/>
    <sbol:displayId>BBa_J23119</sbol:displayId>
    <prov:wasDerivedFrom rdf:resource="http://parts.igem.org/Part:BBa_J23119:Design"/>
    <sbol:elements>ttgacagctagctcagtcctaggtataatgctagc</sbol:elements>
    <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
  </sbol:Sequence>
</rdf:RDF>

```

7.7 ComponentDefinition

The [ComponentDefinition](#) class represents the structural entities of a biological design. The primary usage of this class is to represent structural entities with designed sequences, such as DNA, RNA, and proteins, but it can also be used to represent any other entity that is part of a design, such as small molecules, molecular complexes, and light.

As shown in [Figure 8](#), the [ComponentDefinition](#) class describes a structural design entity using the following properties: [types](#), [roles](#), and [sequences](#). In addition, this class has properties for describing and organizing the substructure of said design entity, including [components](#), [sequenceAnnotations](#), and [sequenceConstraints](#).

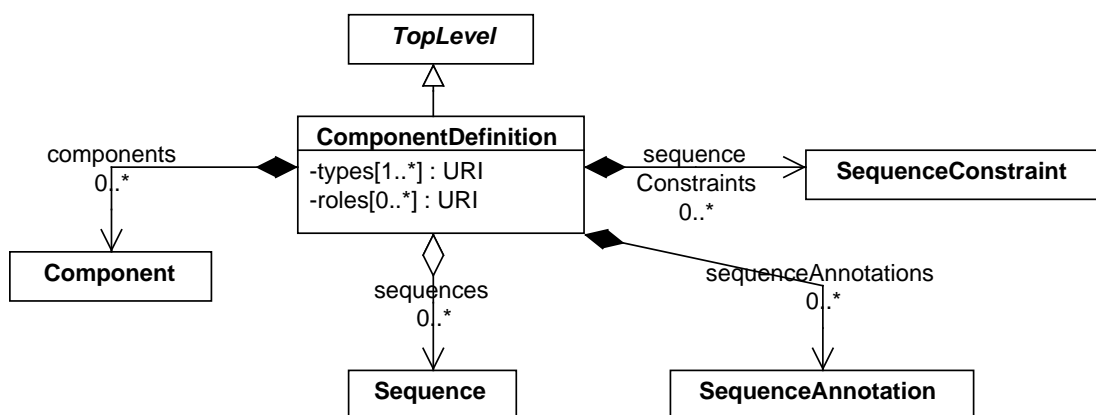


Figure 8: Diagram of the [ComponentDefinition](#) class and its associated properties.

The types property

The **types** property is a REQUIRED set of URIs that specifies the category of biochemical or physical entity (for example DNA, protein, or small molecule) that a [ComponentDefinition](#) object abstracts for the purpose of engineering design.

The **types** property of every **ComponentDefinition** MUST contain one or more **URIs** that MUST identify terms from appropriate ontologies, such as the BioPAX ontology or the ontology of Chemical Entities of Biological Interest (ChEBI). **Table 2** provides a list of RECOMMENDED ontology terms for the **types** property and their **URIs**. In order to maximize the compatibility of designs, any **ComponentDefinition** that can be well-described by one of the terms in **Table 2** MUST use the **URI** for that term as one of its **types**. Finally, if the **types** property contains multiple **URIs**, then they MUST identify non-conflicting terms (otherwise, it may not be clear how to interpret them). For example, the BioPAX terms provided by **Table 2** would conflict because they specify classes of biochemical entities with different molecular structures.

ComponentDefinition Type	URI for BioPAX Term
DNA	http://www.biopax.org/release/biopax-level3.owl#DnaRegion
RNA	http://www.biopax.org/release/biopax-level3.owl#RnaRegion
Protein	http://www.biopax.org/release/biopax-level3.owl#Protein
Small Molecule	http://www.biopax.org/release/biopax-level3.owl#SmallMolecule
Complex	http://www.biopax.org/release/biopax-level3.owl#Complex

Table 2: RECOMMENDED BioPAX terms to specify the **types** property of a **ComponentDefinition**.

The roles property

The **roles** property is an OPTIONAL set of **URIs** that clarifies the potential function of an entity in a biochemical or physical context.

The **roles** property of a **ComponentDefinition** MAY contain one or more **URIs** that MUST identify terms from ontologies that are consistent with the **types** property of the **ComponentDefinition**. For example, the **roles** property of a DNA or RNA **ComponentDefinition** could contain **URIs** identifying terms from the Sequence Ontology (SO). **Table 3** contains a list of RECOMMENDED ontology terms for the **roles** property and their **URIs**. These terms are organized by the type of **ComponentDefinition** to which they SHOULD apply (see **Table 2**). Any **ComponentDefinition** that can be well-described by one of the terms in **Table 3** MUST use the **URI** for that term as one of its **roles**.

ComponentDefinition Role	URI for Ontology Term	ComponentDefinition Type
Promoter	http://identifiers.org/so/SO:0000167	DNA
RBS	http://identifiers.org/so/SO:0000139	DNA
CDS	http://identifiers.org/so/SO:0000316	DNA
Terminator	http://identifiers.org/so/SO:0000141	DNA
Gene	http://identifiers.org/so/SO:0000704	DNA
Operator	http://identifiers.org/so/SO:0000057	DNA
Engineered Gene	http://identifiers.org/so/SO:0000280	DNA
mRNA	http://identifiers.org/so/SO:0000234	RNA
Effector	http://identifiers.org/chebi/CHEBI:35224	Small Molecule

Table 3: RECOMMENDED ontology terms to specify the **roles** property of a **ComponentDefinition**, organized by the type of **ComponentDefinition** to which they should apply (see **Table 2**).

GoKsel to replace URI for Transcription Factor role with something more appropriate (currently is URI for the term "transcription activity")

The sequences property

The **sequences** property is OPTIONAL and MAY include a set of **URI**s that refer to **Sequence** objects. These objects define the primary structure of the **ComponentDefinition**.

Many **ComponentDefinition** objects will refer to precisely one **Sequence** object. For certain use cases, however, it may be appropriate to refer to multiple **Sequence** objects. For example, a user may wish to provide two different representations of the structure of a DNA **ComponentDefinition**, one that represents its structure at the level of nucleotide bases and one that represents its structure at the level of atoms and bonds.

If a **ComponentDefinition** refers to more than one **Sequence** object, then these objects SHOULD be consistent with each other, such that well-defined mappings exist between their **elements** properties in accordance with their **encoding** properties. In addition, if a **ComponentDefinition** refers to more than one **Sequence** with the same **encoding**, then the **elements** of these **Sequence** objects SHOULD have equal lengths. These best practices are intended to make it easier for software tools to locate any regions specified by the **SequenceAnnotation** objects of a **ComponentDefinition** on its associated **Sequence** objects, as well as validate whether its **Sequence** objects are consistent with those associated with any **ComponentDefinition** objects that it may compose via its **Component** objects.

Finally, a **ComponentDefinition** MUST NOT refer to **Sequence** objects with conflicting **encoding** properties. For example, the **IUPAC encoding** properties provided by **Table 1** conflict with each other because they do not specify how to encode the same class of biochemical entity. The **SMILES encoding**, however, does not conflict with them because it specifies how to encode biochemical entities in general, which includes DNA, RNA, and proteins. In addition, if a **ComponentDefinition** refers to one or more **Sequence** objects and its **types** property refers to a term from **Table 2**, then one of these **Sequence** objects MUST have the **encoding** that is cross-listed with this term in **Table 1**. Conversely, if a **ComponentDefinition** refers to a **Sequence** with an **encoding** from **Table 1**, then its **types** property MUST refer to the term from **Table 2** that is cross-listed with this **encoding** in **Table 1**. For example, if the **types** property of a **ComponentDefinition** refers to the BioPAX term for DNA, then one of the **Sequence** objects to which it refers (if any) MUST have an **IUPAC DNA encoding**, and if a **ComponentDefinition** refers to a **Sequence** with an **IUPAC DNA encoding**, then its **types** property must refer to the BioPAX term for DNA.

The components property

The **components** property is OPTIONAL and MAY specify a set of **Component** objects that are contained by the **ComponentDefinition**. Note that the set of relations between **Component** and **ComponentDefinition** objects is strictly acyclic.

While the **ComponentDefinition** class is analogous to a blueprint or specification sheet for a biological part, the **Component** class represents the specific occurrence of a part within a design. Hence, this class allows a biological design to include multiple instances of a particular part (defined by reference to the same **ComponentDefinition**). For example, the **ComponentDefinition** of a polycistronic gene could contain two **Component** objects that refer to the same **ComponentDefinition** of a CDS.

In this way, the **components** property can be used to construct a hierarchy of **ComponentDefinition** objects. If a **ComponentDefinition** in the hierarchy refers to one or more **Sequence** objects, and there exist **ComponentDefinition** objects lower in the hierarchy that refer to **Sequence** objects with the same **encoding**, then the **elements** properties of these **Sequence** objects SHOULD be consistent with each other, such that well-defined mappings exist from the “lower level” **elements** to the “higher level” **elements** in accordance with their shared **encoding** (subject to any restrictions on the positions of **Component** objects in the hierarchy that are imposed by **SequenceAnnotation** or **SequenceConstraint** objects).

A DNA **ComponentDefinition**, for example, could refer to a **Sequence** with an **IUPAC DNA encoding** and an **elements String** of “gattaca.” In turn, this **ComponentDefinition** could contain a **Component** that refers to a “lower level” **ComponentDefinition** that also refers to a **Sequence** with an **IUPAC DNA encoding**. Consequently, the **elements String** of this “lower level” **Sequence** could be “gatta,” or perhaps “tgta” if the **Component** is positioned by a **SequenceAnnotation** that contains a **Location** with an **orientation** of “reverse complement” (see

Section 7.7.5).

The *sequenceAnnotations* property

The *sequenceAnnotations* property is OPTIONAL and MAY contain a set of *SequenceAnnotation* objects. Each *SequenceAnnotation* specifies and describes a potentially discontinuous region on the *Sequence* objects referred to by the *ComponentDefinition*.

In addition, each *SequenceAnnotation* can position a *Component* of the *ComponentDefinition* at the region specified by its *Location* objects (see Section 7.7.5). If more than one *SequenceAnnotation* refers to a *Component* in this manner, then they MUST NOT specify conflicting regions. That is, their *Location* objects MUST NOT specify regions that have conflicting *orientation* properties or occupy non-overlapping positions.

Finally, as a best practice, if a *ComponentDefinition* refers to a *Sequence* with an IUPAC *encoding* from Table 1, then each of its *SequenceAnnotation* objects that contains a *Range* or *Cut* SHOULD specify a region on the *elements* of this *Sequence*. For example, the *ComponentDefinition* of a eukaryotic gene could refer to a *Sequence* with an IUPAC *DNA encoding*. In order to specify the discontinuous region occupied by its CDS, this gene *ComponentDefinition* would need a *SequenceAnnotation* that contains one or more *Range* objects, each one specifying *start* and *end* positions that correspond to indices of the *elements* of its *DNA Sequence*.

The *sequenceConstraints* property

The *sequenceConstraints* property is OPTIONAL and MAY contain a set of *SequenceConstraint* objects. These objects describe any restrictions on the relative, sequence-based positions and/or orientations of the *Component* objects contained by the *ComponentDefinition*. For example, the *ComponentDefinition* of a gene may specify that the position of its promoter *Component* precedes that of its CDS *Component*. This is particularly useful when a *ComponentDefinition* lacks a *Sequence* and therefore cannot specify the precise, sequence-based positions of its *Component* objects using *SequenceAnnotation* objects.

Serialization

The serialization of a *ComponentDefinition* MUST have the form below. The *components*, *sequenceConstraints*, *sequenceAnnotations*, and *sequences* properties of a *ComponentDefinition* contain objects belonging to the appropriate SBOL classes as their values, while the *types* and *roles* properties contain URIs that identify ontology terms as their values. As shown below, each of these objects and URIs is serialized as part of an implicit set of SBOL properties with singular rather than plural names. In particular, each object is serialized as a RDF/XML node nested within a singular property, while each URI (except the *identity*) is serialized as a *rdf:resource* on a singular property.

```
<sbol:ComponentDefinition rdf:about="...">
  ... properties inherited from identified ...
  zero or more <sbol:sequence rdf:resource="..."> element
  one or more <sbol:type rdf:resource="..."> elements
  zero or more <sbol:role rdf:resource="..."> elements
  zero or more <sbol:component>
    <sbol:Component rdf:about="...">...</sbol:Component>
  </sbol:component> elements
  zero or more <sbol:sequenceAnnotation>
    <sbol:SequenceAnnotation rdf:about="...">...</sbol:SequenceAnnotation>
  </sbol:sequenceAnnotation> elements
  zero or more <sbol:sequenceConstraint>
    <sbol:SequenceConstraint rdf:about="...">...</sbol:SequenceConstraint>
  </sbol:sequenceConstraint> elements
</sbol:ComponentDefinition>
```

The example below shows the serialization for the *ComponentDefinition* of a promoter. The BioPAX term *DnaRegion* and the ChEBI term *CHEBI:4705* (double-stranded DNA) are used to indicate that the type of biological entity represented by this *ComponentDefinition* is DNA. Its role is specified using the SO terms *SO:0000167*

(promoter) and the more specific SO:0000613 (bacterial_RNApol_promoter).

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:prov="http://www.
w3.org/ns/prov#" xmlns:sbol="http://sbols.org/v2#">
  <sbol:ComponentDefinition rdf:about="http://partsregistry.org/cd/BBa_J23119">
    <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_J23119"/>
    <sbol:displayId>BBa_J23119</sbol:displayId>
    <prov:wasDerivedFrom rdf:resource="http://partsregistry.org/Part:BBa_J23119"/>
    <dcterms:title>J23119 promoter</dcterms:title>
    <dcterms:description>Constitutive promoter</dcterms:description>
    <sbol:type rdf:resource="http://identifiers.org/chebi/CHEBI:4705"/>
    <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
    <sbol:role rdf:resource="http://identifiers.org/so/SO:0000167"/>
    <sbol:role rdf:resource="http://identifiers.org/so/SO:0000613"/>
    <sbol:sequence rdf:resource="http://partsregistry.org/seq/BBa_J23119"/>
  </sbol:ComponentDefinition>
</rdf:RDF>
```

7.7.1 ComponentInstance

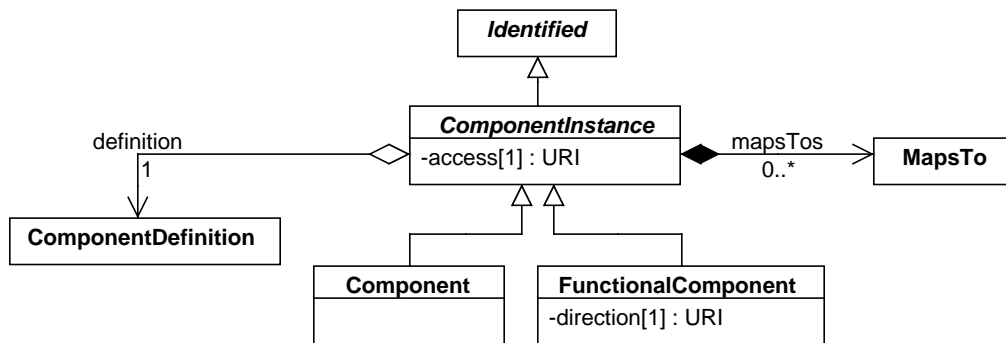


Figure 9: Diagram of the [ComponentInstance](#) class and its associated properties.

The [ComponentInstance](#) abstract class is inherited by SBOL classes that represent the usage or occurrence of a [ComponentDefinition](#) within a larger design (that is, another [ComponentDefinition](#) or [ModuleDefinition](#)). Currently, there are two subclasses of [ComponentInstance](#):

- The [Component](#) class is used to specify the structural usage of a [ComponentDefinition](#) inside another [ComponentDefinition](#) via the [components](#) property.
- The [FunctionalComponent](#) class is used to specify the functional usage of a [ComponentDefinition](#) inside a [ModuleDefinition](#) via the [functionalComponents](#) property. This class is described in [Section 7.9.1](#).

The definition property

The [definition](#) property is a REQUIRED [URI](#) that refers to the [ComponentDefinition](#) of the [ComponentInstance](#). As described in the previous section, this [ComponentDefinition](#) effectively provides information about the [types](#) and [roles](#) of the [ComponentInstance](#).

The [definition](#) property MUST NOT refer to the same [ComponentDefinition](#) as the one that contains the [ComponentInstance](#). Furthermore, [ComponentInstance](#) objects MUST NOT form a cyclical chain of references via their [definition](#) properties and the [ComponentDefinition](#) objects that contain them. For example, consider

the [ComponentInstance](#) objects *A* and *B* and the [ComponentDefinition](#) objects *X* and *Y*. The reference chain “*X* contains *A*, *A* is defined by *Y*, *Y* contains *B*, and *B* is defined by *X*” is cyclical.

The *mapsTo* property

The *mapsTo* property is OPTIONAL and MAY contain a set of [MapsTo](#) objects that refer to and link together [ComponentInstance](#) objects (both [Component](#) objects and [FunctionalComponent](#) objects) within a larger design.

Section 7.7.3 contains a more detailed description of the [MapsTo](#) class.

The *access* property

The *access* property is a REQUIRED [URI](#) that indicates whether the [ComponentInstance](#) can be referred to by a [MapsTo](#).

Table 4 provides a list of REQUIRED *access* URIs. The value of the *access* property MUST be one of these URIs.

Access URI	Description
http://sbols.org/v2#public	The ComponentInstance MAY be referred to by MapsTo objects.
http://sbols.org/v2#private	The ComponentInstance MUST NOT be referred to by any MapsTo object.

Table 4: REQUIRED URIs for the *access* property.

Unless a designer has a reason to prevent others from accessing the [ComponentInstance](#) when their design is reused, it is RECOMMENDED that the *access* property of the [ComponentInstance](#) be set to “public.” For example, a designer who is concerned about retroactivity might set the *access* of the [ComponentInstance](#) to “private” in order to prevent others from specifying its [Participation](#) in a new [Interaction](#) as part of a composite design.

Serialization

No serialization is defined for the [ComponentInstance](#) class, since this class is only used indirectly through the [Component](#) and [FunctionalComponent](#) subclasses.

7.7.2 Component

The [Component](#) class is used to compose [ComponentDefinition](#) objects into a structural hierarchy. For example, the [ComponentDefinition](#) of a gene could contain four [Component](#) objects, including a promoter, RBS, CDS, and terminator. In turn, the [ComponentDefinition](#) of the promoter [Component](#) could contain [Component](#) objects defined as various operator sites.

Serialization

The serialization of a [Component](#) MUST have the following form:

```
<sbol:Component rdf:about="...">
  ... properties inherited from identified ...
  one      <sbol:access rdf:resource="..."> element
  one      <sbol:definition rdf:resource="..."> element
  zero or more <sbol:mapsTo rdf:resource="..."> elements
</sbol:Component>
```

The example below shows the serialization of a [Component](#) that represents an instance of a promoter:

```
<sbol:Component rdf:about="http://partsregistry.org/cd/BBa_F2620/pLuxR">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/pLuxR"/>
  <sbol:displayName>pLuxR</sbol:displayName>
  <sbol:access rdf:resource="http://sbols.org/v2#public"/>
```

```
<sbol:definition rdf:resource="http://partsregistry.org/cd/BBa_R0062"/>
</sbol:Component>
```

7.7.3 MapsTo

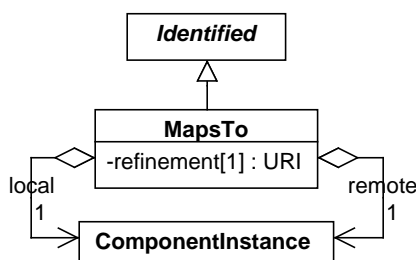


Figure 10: Diagram of the *MapsTo* class and its associated properties.

When *ComponentDefinition* and *ModuleDefinition* objects are composed into structural and functional hierarchies using *ComponentInstance* and *Module* objects, it is often the case that some *ComponentInstance* objects are intended to represent the same entity in the overall design. The purpose of the *MapsTo* class is to make these identity relationships clear and explicit. For example, consider a *ModuleDefinition* for a generic genetic inverter that includes a *FunctionalComponent* for a repressor protein. When this *ModuleDefinition* is instantiated within a higher level *ModuleDefinition* that includes a *FunctionalComponent* for a LacI protein, the *MapsTo* object can be used to indicate that the specific repressor protein in this instance is LacI.

In particular, a *MapsTo* object provides two pieces of information:

- An identity relationship between two *ComponentInstance* objects, the first contained by the “lower level” definition of the *ComponentInstance* or *Module* that owns the *MapsTo*, and the second contained by the “higher level” definition that contains this *ComponentInstance* or *Module*. The *remote* property of a *MapsTo* refers to the first “lower level” *ComponentInstance*, while the *local* property refers to the second “higher level” *ComponentInstance*.
- Instructions on how to interpret *local* and *remote* *ComponentInstance* objects that refer to different *ComponentDefinition* objects (that is, they are not identical). These are specified using the *refinement* property of the *MapsTo* class.

To illustrate this concept, two examples are provided in Figure 11, in which the *ComponentDefinition* of a transcriptional unit is specified by composing two “lower level” *ComponentDefinition* objects. In both examples, the two “lower level” *ComponentDefinition* objects each contain a RBS *Component* that is intended to represent the same design entity in the “higher level” *ComponentDefinition* of the transcriptional unit.

In order to explicitly represent the identity relationships in this example, a new RBS *Component* needs to be created inside the “higher level” *ComponentDefinition*. This “higher level” *Component* then needs to be linked to the equivalent “lower level” *Component* objects by means of the *MapsTo* class, using one *MapsTo* object per link. For example, in order to link the “higher level” RBS *Component* to the “lower level” RBS *Component* of the promoter-RBS *ComponentDefinition*, a *MapsTo* must be created on the “higher level” promoter-RBS *Component*. The *local* property of this *MapsTo* must then refer to the “higher level” RBS *Component*, while its *remote* property must refer to the “lower level” RBS *Component*. In this way, many “lower level” *Component* objects can be linked together at the “higher level” using an equal number of *MapsTo* objects, each one referring to a different *remote* *Component*, but all referring to the same *local* *Component*.

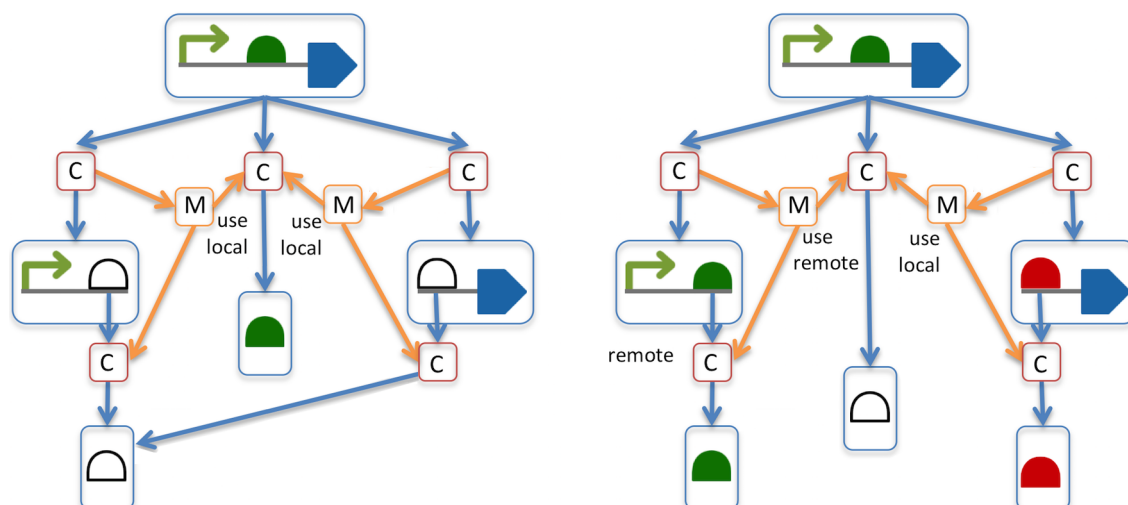


Figure 11: Linking [Component](#) objects using [MapsTo](#) entities. Boxes with diagrams represent [ComponentDefinition](#) objects, boxes with the C label represent [Component](#) objects, and boxes with the M label represent [MapsTo](#) objects. In both diagrams, a promoter-RBS [ComponentDefinition](#) and a RBS-CDS [ComponentDefinition](#) are being composed to form the [ComponentDefinition](#) of a complete transcriptional unit. In the lefthand diagram, the two [Component](#) objects inside the promoter-RBS [ComponentDefinition](#) and RBS-CDS [ComponentDefinition](#) objects both refer to an abstract RBS [ComponentDefinition](#) that lacks a sequence (white semicircle). Through the use of [MapsTo](#) objects with [refinement](#) set to [useLocal](#), these “lower level” [ComponentDefinition](#) objects are effectively overridden by that of the green RBS in the [ComponentDefinition](#) of the complete transcriptional unit. In the righthand diagram, however, the two “lower level” RBS [ComponentDefinition](#) objects do not lack sequences and it is the “higher level” RBS [ComponentDefinition](#) that is abstract. In this case, one of the [MapsTo](#) objects has a [useRemote](#) [refinement](#), resulting in the green RBS [ComponentDefinition](#) overriding that of the abstract RBS in the “higher level” [ComponentDefinition](#).

The same types of identity relationships can also be declared between [FunctionalComponent](#) objects contained by [ModuleDefinition](#) objects, or between [Component](#) objects and [FunctionalComponent](#) objects contained by [ComponentDefinition](#) objects and [ModuleDefinition](#) objects, respectively. See [Section 8](#) and [Section B](#) for additional examples using the [MapsTo](#) class.

The local property

This REQUIRED property has a data type of [URI](#) and is used to refer to the [ComponentInstance](#) contained by the “higher level” [ComponentDefinition](#) or [ModuleDefinition](#). This [local](#) [ComponentInstance](#) MUST be contained by the [ComponentDefinition](#) or [ModuleDefinition](#) that contains the [ComponentInstance](#) or [Module](#) that owns the [MapsTo](#). Finally, the [access](#) property of the [local](#) [ComponentInstance](#) MUST be set to “public.”

The remote property

This REQUIRED property has a data type of [URI](#) and is used to refer to the [ComponentInstance](#) contained by the “lower level” [ComponentDefinition](#) or [ModuleDefinition](#). This [remote](#) [ComponentInstance](#) MUST be contained by the [ComponentDefinition](#) or [ModuleDefinition](#) that is the [definition](#) of the [ComponentInstance](#) or [Module](#) that owns the [MapsTo](#). Lastly, the [access](#) property of the [remote](#) [ComponentInstance](#) MUST be set to “public.”

The refinement property

The [refinement](#) property is REQUIRED and has a data type of [URI](#). Each [MapsTo](#) object MUST specify the relationship between its [local](#) and [remote](#) [ComponentInstance](#) objects using one of the REQUIRED [refinement](#) URIs provided in [Table 5](#).

Serialization

The serialization of [MapsTo](#) MUST have the following form.

Refinement URI	Description
http://sbols.org/v2#useRemote	All references to the definition property of the local ComponentInstance MUST dereference to that of the remote ComponentInstance instead.
http://sbols.org/v2#useLocal	In the context of the ComponentDefinition or ModuleDefinition that contains the owner of the MapsTo , all references to the definition property of the remote ComponentInstance MUST dereference to that of the local ComponentInstance instead.
http://sbols.org/v2#verifyIdentical	The definition properties of the local and remote ComponentInstance objects MUST refer to the same ComponentDefinition .
http://sbols.org/v2#merge	In the context of the ComponentDefinition or ModuleDefinition that contains the owner of the MapsTo , all references to the definition property of the local ComponentInstance or that of the remote ComponentInstance MUST dereference to both properties.

Table 5: REQUIRED URIs for the [refinement](#) property.

```

<sbol:MapsTo rdf:about="...">
  ... properties inherited from identified ...
  one <sbol:refinement rdf:resource="..."> element
  one <sbol:remote rdf:resource="..."> element
  one <sbol:local rdf:resource="..."> element
</sbol:MapsTo>

```

In the example below, a [FunctionalComponent](#) in a “lower level” [LacI inverter ModuleDefinition](#) is linked to a [FunctionalComponent](#) in a “higher level” [ModuleDefinition](#) of a genetic toggle switch. The full example can be found in [Section B.2](#).

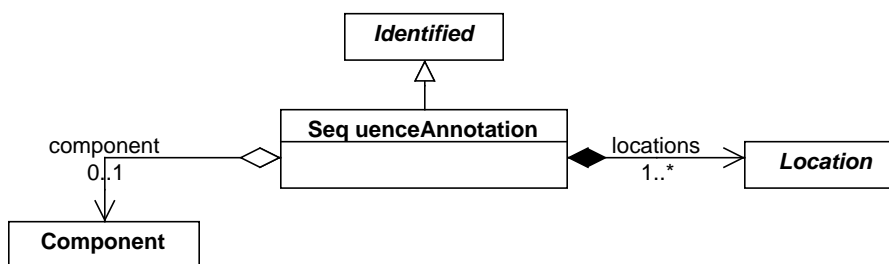
```

<sbol:MapsTo rdf:about="http://sbolstandard.org/example/toggle_switch/laci_inverter/LacI_mapping">
  <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/toggle_switch/laci_inverter/LacI_mapping"/>
  <sbol:displayId>LacI_mapping</sbol:displayId>
  <sbol:refinement rdf:resource="http://sbols.org/v2#useRemote"/>
  <sbol:remote rdf:resource="http://sbolstandard.org/example/toggle_switch/LacI"/>
  <sbol:local rdf:resource="http://sbolstandard.org/example/laci_inverter/TF"/>
</sbol:MapsTo>

```

7.7.4 SequenceAnnotation

The [SequenceAnnotation](#) class describes one or more regions of interest on the [Sequence](#) objects referred to by its parent [ComponentDefinition](#). In addition, [SequenceAnnotation](#) objects can describe the substructure of their parent [ComponentDefinition](#) through association with the [Component](#) objects contained by this [ComponentDefinition](#).

Figure 12: Diagram of the [SequenceAnnotation](#) class and its associated properties.

The *locations* property

The **locations** property is a REQUIRED set of one or more **Location** objects that indicate which **elements** of a **Sequence** are described by the **SequenceAnnotation**.

Allowing multiple **Location** objects on a single **SequenceAnnotation** is intended to allow representation of discontinuous ranges (for example, a **Component** encoded across a set of exons with interspersed introns). As such, the **Location** objects of a single **SequenceAnnotation** SHOULD NOT specify overlapping regions, since it is not clear what this would mean. There is no such concern with different **SequenceAnnotation** objects, however, which can freely overlap in **Location** (for example, specifying overlapping linkers for sequence assembly).

The *component* property

The **component** property is OPTIONAL and has a data type of **URI**. This **URI** MUST refer to a **Component** that is contained by the same parent **ComponentDefinition** that contains the **SequenceAnnotation**. In this way, the properties of the **SequenceAnnotation**, such as its **description** and **locations**, are associated with part of the substructure of its **ComponentDefinition**.

Serialization

The serialization of a **SequenceAnnotation** MUST have the form below. In this template, A_LOCATION_SUBCLASS represents one of the **Location** subclasses.

```
<sbol:SequenceAnnotation rdf:about="...">
  ... properties inherited from identified ...
  zero or one <sbol:component rdf:resource="..."> element
  one or more
    <sbol:location>
      <sbol:A_LOCATION_SUBCLASS rdf:about="...">...</sbol:A_LOCATION_SUBCLASS>
    </sbol:location> elements
</sbol:SequenceAnnotation>
```

The example below shows the serialization of a **SequenceAnnotation** object. It specifies the region occupied by a **Component** named BBa_F2620.

```
<sbol:SequenceAnnotation rdf:about="http://partsregistry.org/cd/BBa_F2620/anno2">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno2"/>
  <sbol:displayId>anno2</sbol:displayId>
  <sbol:location>
    <sbol:Range rdf:about="http://partsregistry.org/cd/BBa_F2620/anno2/range">
      <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno2/range"/>
      <sbol:displayId>range</sbol:displayId>
      <sbol:start>56</sbol:start>
      <sbol:end>68</sbol:end>
      <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
    </sbol:Range>
  </sbol:location>
  <sbol:component rdf:resource="http://partsregistry.org/cd/BBa_F2620/rbs"/>
</sbol:SequenceAnnotation>
```

7.7.5 Location

The **Location** class is extended by the **Range**, **Cut**, and **GenericLocation** classes.

The *orientation* property

The **orientation** property is OPTIONAL and has a data type of **URI**. All subclasses of **Location** share this property, which can be used to indicate how the region specified by the **SequenceAnnotation** and any associated double-stranded **Component** is oriented on the **elements** of a **Sequence** from their parent **ComponentDefinition**. Table 6 provides a list of REQUIRED **orientation** URIs. If a **Location** object has an **orientation**, then it MUST come from Table 6.

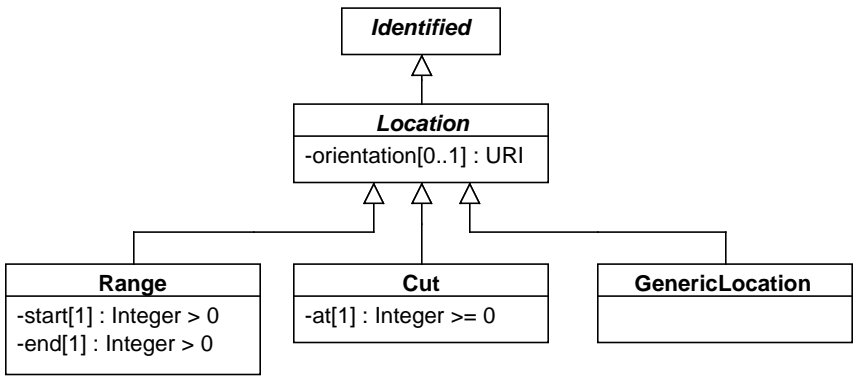


Figure 13: Diagram of the [Location](#) class and its associated properties.

Orientation URI	Description
http://sbols.org/v2#inline	The region specified by this Location MUST be on the elements of a Sequence .
http://sbols.org/v2#reverseComplement	The region specified by this Location MUST be on the reverse-complement translation of the elements of a Sequence . The exact nature of this translation depends on the encoding of the Sequence .

Table 6: REQUIRED URIs for the [orientation](#) property

Range

A [Range](#) object specifies a region via discrete, inclusive [start](#) and [end](#) positions that correspond to indices for characters in the [elements String](#) of a [Sequence](#).

Note that the index of the first location is 1, as is typical practice in biology, rather than 0, as is typical practice in computer science.

The start property

The [start](#) property specifies the inclusive starting position of the [Range](#). This property is REQUIRED and MUST contain an [Integer](#) value greater than zero.

The end property

The [end](#) property specifies the inclusive ending position of the [Range](#). This property is REQUIRED and MUST contain an [Integer](#) value greater than zero. In addition, this [Integer](#) value MUST be greater than or equal to that of the [start](#) property.

Serialization

The serialization of a [Range](#) MUST have the following form:

```
<sbol:Range rdf:about="...">
  ... properties inherited from identified ...
  one      <sbol:start>...</sbol:start> element
  one      <sbol:end>...</sbol:end> element
  zero or one <sbol:orientation rdf:resource="..."> element
</sbol:Range>
```

The example below shows the serialization of a [Range](#) object. It specifies the region between the inclusive positions 56 and 68, with an [orientation](#) of “inline.”

```
<sbol:Range rdf:about="http://partsregistry.org/cd/BBa_F2620/anno2/range">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno2/range"/>
  <sbol:displayId>range</sbol:displayId>
  <sbol:start>56</sbol:start>
  <sbol:end>68</sbol:end>
  <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
</sbol:Range>
```

Cut

The [Cut](#) class has been introduced to enable the specification of a region between two discrete positions. This specification is accomplished using the [at](#) property, which specifies a discrete position that corresponds to the index of a character in the [elements String](#) of a [Sequence](#) (except in the case when [at](#) is equal to zero—see below).

The at property

The [at](#) property is REQUIRED and MUST contain an [Integer](#) value greater than or equal to zero. The region specified by the [Cut](#) is between the position specified by this property and the position that immediately follows it. When the [at](#) property is equal to zero, the specified region is immediately before the first discrete position or character in the [elements String](#) of a [Sequence](#).

Serialization

The serialization of a [Cut](#) MUST have the following form:

```
<sbol:Cut rdf:about="...">
  ... properties inherited from identified ...
  one      <sbol:at>...</sbol:at> element
  zero or one <sbol:orientation rdf:resource="..."> element
</sbol:Cut>
```

The example below shows the serialization of a [Cut](#) object. It specifies a region in between positions 10 and 11, with an [orientation](#) of “inline.”

```
<sbol:Cut rdf:about="http://partsregistry.org/cd/BBa_J23119/cutat10/cut">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_J23119/cutat10/cut"/>
  <sbol:displayId>cut</sbol:displayId>
  <sbol:at>10</sbol:at>
  <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
</sbol:Cut>
```

GenericLocation

While the [Range](#) and [Cut](#) classes are best suited to specifying regions on [Sequence](#) objects with IUPAC encodings, the [GenericLocation](#) class is included as a starting point for specifying regions on [Sequence](#) objects with different [encoding](#) properties and potentially nonlinear structure. This class can also be used to set the [orientation](#) of a [SequenceAnnotation](#) and any associated [Component](#) when their parent [ComponentDefinition](#) is partial design that lacks a [Sequence](#).

Serialization

The serialization of a [GenericLocation](#) MUST have the following form:

```
<sbol:GenericLocation rdf:about="...">
  ... properties inherited from identified ...
  zero or one <sbol:orientation rdf:resource="..."> element
```

```
</sbol:GenericLocation>
```

The example below shows the serialization of a [GenericLocation](#) object with an [orientation](#) of “reverse complement”:

```
<sbol:GenericLocation rdf:about="http://www.partsregistry.org/Part:BBa_F2620/anno5/location">
  <sbol:orientation rdf:resource="http://sbols.org/v2#reverseComplement"/>
</sbol:GenericLocation>
```

7.7.6 SequenceConstraint

The [SequenceConstraint](#) class can be used to assert restrictions on the relative, sequence-based positions of pairs of [Component](#) objects contained by the same parent [ComponentDefinition](#). The primary purpose of this class is to enable the specification of partially designed [ComponentDefinition](#) objects, for which the precise positions or ordering of their contained [Component](#) objects are not yet fully determined. Each [SequenceConstraint](#) includes the [restriction](#), [subject](#), and [object](#) properties.

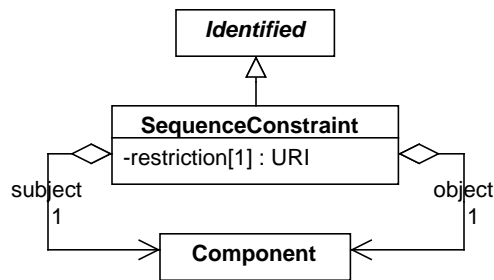


Figure 14: Diagram of the [SequenceConstraint](#) class and its associated properties.

The subject property

The [subject](#) property is REQUIRED and MUST contain a [URI](#) that refers to a [Component](#) contained by the same parent [ComponentDefinition](#) that contains the [SequenceConstraint](#).

The object property

The [object](#) property is REQUIRED and MUST contain a [URI](#) that refers to a [Component](#) contained by the same parent [ComponentDefinition](#) that contains the [SequenceConstraint](#). This [Component](#) MUST NOT be the same [Component](#) that the [SequenceConstraint](#) refers to via its [subject](#) property.

The restriction property

The [restriction](#) property is REQUIRED and has a data type of [URI](#). This property is used to indicate the type of structural restriction on the relative, sequence-based positions or orientations of the [subject](#) and [object](#) [Component](#) objects. The [URI](#) value of this property SHOULD come from the RECOMMENDED [URIs](#) in [Table 7](#).

Per Mike B comment: changed “MUST ... or an [ill-defined] appropriate ontology” to SHOULD, since that’s the same. -JSB I’m not sure about this. I think it is MUST, as these are the only ones we allow right now. Same for all enumerated types we have defined. -CJM I agree with CJM. The original wording was problematic because it said ‘restrictions MUST come from table 7 or anywhere else,’ which we’ve simplified to a SHOULD. But implementors need a solid definition of each option, and a complete enumeration of options, because a hard constraint MUST never be disobeyed – and that means that implementors MUST enforce all constraints. So this can’t be open-ended. -MB So the outstanding question is do we want to make this SHOULD or MUST? -CJM

Restriction URI	Description
http://sbols.org/v2#precedes	The position of the subject Component MUST precede that of the object Component . If each one is associated with a SequenceAnnotation , then the SequenceAnnotation associated with the subject Component MUST specify a region that starts before the region specified by the SequenceAnnotation associated with the object Component .
http://sbols.org/v2#sameOrientationAs	The subject and object Component objects MUST have the same orientation. If each one is associated with a SequenceAnnotation , then the orientation URIs of the Location objects of the first SequenceAnnotation MUST be represented among those of the second SequenceAnnotation , and vice versa.
http://sbols.org/v2#oppositeOrientationAs	The subject and object Component objects MUST have opposite orientations. If each one is associated with a SequenceAnnotation , then the orientation URIs of the Location objects of one SequenceAnnotation MUST NOT be represented among those of the other SequenceAnnotation .

Table 7: RECOMMENDED URIs for the *restriction* property.

We need things like nextTo and overlapping, and think we might be able to get it from region-connection-calculus ontology

Serialization

The serialization of a **SequenceConstraint** MUST have the following form:

```
<sbol:SequenceConstraint rdf:about="...">
  ... properties inherited from identified ...
  one <sbol:restriction rdf:resource="..."> element
  one <sbol:subject rdf:resource="..."> element
  one <sbol:object rdf:resource="..."> element
</sbol:SequenceConstraint>
```

The example below shows the serialization of a **SequenceConstraint** belonging to the **ComponentDefinition** of a LacI-repressible promoter. This **SequenceConstraint** has a “precedes” **restriction** that indicates that the **subject Component**, which represents the core of the promoter, is positioned before the **object Component**, which represents the LacI operator of the promoter.

```
<sbol:SequenceConstraint rdf:about="http://partsregistry.org/cd/BBa_K174004/r1">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_K174004/r1"/>
  <sbol:displayName>r1</sbol:displayName>
  <sbol:restriction rdf:resource="http://sbols.org/v2#precedes"/>
  <sbol:subject rdf:resource="http://partsregistry.org/cd/pspac"/>
  <sbol:object rdf:resource="http://partsregistry.org/cd/LacI_operator"/>
</sbol:SequenceConstraint>
```

7.8 Model

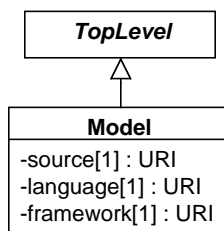


Figure 15: Diagram of the `Model` class and its associated properties.

The purpose of the `Model` class is to serve as a placeholder for an external model and provide additional meta-data to enable better reasoning about the contents of this model. In this way, there is minimal duplication of standardization efforts and users of SBOL can specify the quantitative function of a `ModuleDefinition` in the language of their choice.

The meta-data provided by the `Model` class include the following properties: the `source` or location of the actual content of the model, the `language` in which the model is implemented, and the model's mathematical `framework`.

The `source` property

The `source` property is REQUIRED and MUST contain a URI reference to a qualitative or quantitative model.

The `language` property

The `language` property is REQUIRED and MUST contain a URI that specifies the language in which the model is implemented. It is RECOMMENDED that this URI refer to a term from the EMBRACE Data and Methods (EDAM) ontology. Table 8 provides a list of RECOMMENDED terms from this ontology and their URIs. If the `language` property of a `Model` is well-described by one these terms, then it MUST use the URI for this term as its value.

Model Language	URI
SBML	http://identifiers.org/edam/format_2585
CellML	http://identifiers.org/edam/format_3240
BioPAX	http://identifiers.org/edam/format_3156

Table 8: RECOMMENDED terms from the EDAM ontology to specify the `language` property of a `Model`.

The `framework` property

This REQUIRED property is a URI that specifies the modeling framework that a model is implemented within. Values for this URI are RECOMMENDED to be chosen from the SBO's modeling framework terms where possible. A few suggested model frameworks and corresponding URI values are shown in Table 9.

Serialization

The serialization of a `Model` MUST have the following form:

```

<sbol:Model rdf:about="http://www.sbolstandard.org/examples/toogleswitch">
  ... properties inherited from identified ...
  one <sbol:source rdf:resource="..."> element

```

Framework	URI
Continuous	http://identifiers.org/biomodels.sbo/SBO:0000062
Discrete	http://identifiers.org/biomodels.sbo/SBO:0000063

Table 9: Example modelling frameworks and corresponding SBO terms.

```

one <sbol:language rdf:resource="..."> element
one <sbol:framework rdf:resource="..."> element
</sbol:Model>

```

The example below shows the serialization of a **Model** object that refers to a quantitative model of a genetic toggle switch. The model is implemented in the SBML **language** and adheres to a continuous modeling **framework**. Lastly, the model can be retrieved from a model repository via its **source** URI, which is a URL.

```

<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:prov="http://www.
w3.org/ns/prov#" xmlns:sbol="http://sbols.org/v2#">
  <sbol:Model rdf:about="http://www.sbolstandard.org/examples/pIKE_Toggle_1">
    <sbol:persistentIdentity rdf:resource="http://www.sbolstandard.org/examples/pIKE_Toggle_1"/>
    <sbol:displayId>pIKE_Toggle_1</sbol:displayId>
    <dcterms:title>pIKE_Toggle_1 toggle switch</dcterms:title>
    <sbol:source rdf:resource="http://virtualparts.org/part/pIKE_Toggle_1"/>
    <sbol:language rdf:resource="http://identifiers.org/edam/format_2585"/>
    <sbol:framework rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000062"/>
  </sbol:Model>
</rdf:RDF>

```

7.9 ModuleDefinition

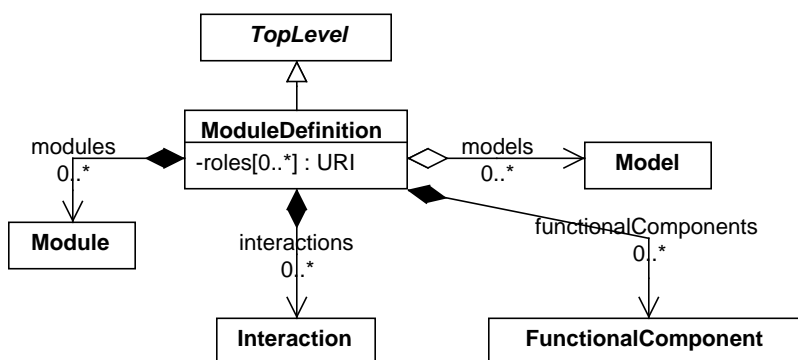


Figure 16: Diagram of the **ModuleDefinition** class and its associated properties.

The **ModuleDefinition** class is the hub where the structural and functional aspects of genetic designs are brought together to form a complete picture of the design. A **ModuleDefinition** object is composed from zero or more **FunctionalComponent**, **Module**, and **Interaction** objects, and links to zero or more **Model** objects.

As an engineering object, a **ModuleDefinition** will often have certain of its **FunctionalComponent** objects that are intended to carry signals in or out of it. This functionality of designated “inputs” and “outputs” is expressed by **direction** properties on its **FunctionalComponent** elements.

The roles property

The **roles** property is an OPTIONAL set of **URI**s that clarifies the intended function of a **ModuleDefinition** in a biological context.

These terms might identify “logical” roles, such as “inverter” or “AND gate”, or they might identify descriptive biological roles, such as “metabolic pathway” and “signaling cascade,” or might identify roles from some other manner of describing intended function. Interpretation of the meaning of such roles is currently implementation-dependent.

The modules property

The **modules** property is OPTIONAL and MAY specify a set of **Module** objects contained by the **ModuleDefinition**. Note that the set of relations between **Module** and **ModuleDefinition** objects is strictly acyclic.

While the **ModuleDefinition** class is analogous to a blueprint or specification sheet for a system of interaction biological elements, the **Module** class represents the specific occurrence of a particular sub-system within that design. Hence, this class allows a biological design to include multiple instances of a subsystem (defined by reference to the same **ModuleDefinition**). For example, the **ModuleDefinition** for a network of two-input repressor systems, where the particular repressors have not yet been chosen, contain multiple **Module** objects that refer to the same **URI** for the **ModuleDefinition** of an abstract two-input repressor device.

The functionalComponents property

The **functionalComponents** property is OPTIONAL and MAY specify a set of **FunctionalComponent** objects contained by the **ModuleDefinition**.

Just as a **Module** represents an instance of a subsystem in the **ModuleDefinition** "blueprint", a **FunctionalComponent** represents an instance of an individual element whose **ComponentDefinition** may be used multiple times in a **ModuleDefinition**. For example, a **ModuleDefinition** might contain several copies of a particular promoter.

The interactions property

The **interactions** property is OPTIONAL and MAY specify a set of **Interaction** objects within a **ModuleDefinition**.

The **Interaction** class provides an abstract, machine-friendly representation of the functional interactions of entities within a **ModuleDefinition** (whereas a **Model** of a system may not be readily susceptible to machine reasoning, depending on how it is implemented). Each **Interaction** includes **Participation** entities that indicate the roles of the **FunctionalComponent** objects involved in the **Interaction**

The models property

The **models** property is OPTIONAL and MAY specify a set of **URI**s identifying **Model** objects.

SBOL's **Model** objects are placeholders used to link specifications of biological parts and their interactions to computational models of arbitrary format. A **ModuleDefinition** object can link to more than one **Model** since each might encode the same system in a different way or at a different level of detail.

Serialization

The serialization of **ModuleDefinition** has the following form:

```
<sbol:ModuleDefinition rdf:about="...">
  ... properties inherited from identified ...
  zero or more <sbol:role rdf:resource="..."> elements
  zero or more <sbol:model rdf:resource="..."> elements
  zero or more <sbol:functionalComponent>
    <sbol:FunctionalComponent rdf:about="...">...</sbol:FunctionalComponent >
  </sbol:functionalComponent> elements
```

```

zero or more <sbol:module>
    <sbol:Module rdf:about="...">...</sbol:Module>
</sbol:module> elements
zero or more <sbol:interaction>
    <sbol:Interaction rdf:about="...">...</sbol:Interaction>
</sbol:interaction> elements
</sbol:ModuleDefinition>

```

The example below shows a simple **ModuleDefinition** containing two components, a **FunctionalComponent** for a DNA sequence encoding constitutive expression of GFP and another for the GFP protein expressed from this sequence, plus an interaction describing that relation.

```

<sbol:ModuleDefinition rdf:about="http://sbolstandard.org/example/md/GFP_expression">
  <sbol:functionalComponent>
    <sbol:FunctionalComponent rdf:about="http://sbolstandard.org/example/md/GFP_expression/GFP_protein">
      <sbol:definition rdf:resource="http://sbolstandard.org/example/GFP"/>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:direction rdf:resource="http://sbols.org/v2#output"/>
    </sbol:FunctionalComponent>
  </sbol:functionalComponent>
  <sbol:functionalComponent>
    <sbol:FunctionalComponent rdf:about="http://sbolstandard.org/example/md/GFP_expression/Constitutive_GFP">
      <sbol:definition rdf:resource="http://sbolstandard.org/example/GFP_generator"/>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:direction rdf:resource="http://sbols.org/v2#none"/>
    </sbol:FunctionalComponent>
  </sbol:functionalComponent>
  <sbol:interaction>
    <sbol:Interaction rdf:about="http://sbolstandard.org/example/md/GFP_expression/express_GFP">
      ...
    </sbol:Interaction>
  </sbol:interaction>
</sbol:ModuleDefinition>

```

7.9.1 FunctionalComponent

A **FunctionalComponent** is an instance of a **ComponentDefinition** being used as part of a **ModuleDefinition** object. Each **FunctionalComponent** object is owned by a **ModuleDefinition** object and represents an explicit usage of a **ComponentDefinition** object for the purpose of fulfilling some function.

FunctionalComponent derives from **ComponentInstance**, and therefore has the **definition**, **access**, and **mapsTo** properties. Additionally, it has a **direction** property that specifies whether it serves as an input, output, both, or neither with regards to the **ModuleDefinition** that contains it.

The direction property

Each **FunctionalComponent** MUST specify via the **direction** property whether it serves as an input, output, both, or neither for its parent **ModuleDefinition** object. The value for this property MUST be one of the values given in Table 10.

Direction URI	Description
http://sbols.org/v2#inout	To indicate a FunctionalComponent can be used as both input or output
http://sbols.org/v2#in	To indicate a FunctionalComponent can be used as input
http://sbols.org/v2#out	To indicate a FunctionalComponent can be used as output
http://sbols.org/v2#none	To indicate a FunctionalComponent is neither input nor output

Table 10: URIs for the **direction** property.

The **direction** property is a means to encode a common way in which designers think about the “purpose” of a connection in a system. In this case, the connection is the **FunctionalComponent**, and the system is the **ModuleDefinition**. For example, consider a system that is designed to indicate concentration of the cell-cell signalling molecule 3OC₆HSL by the concentration of the product of a particular CDS. In this system, the concentration of 3OC₆HSL is the signal being interpreted by the system, so the **FunctionalComponent** for 3OC₆HSL would have a **direction** of “input.” Complementarily, the concentration of the designated product is the signal intended for consumption by other biologicals systems, and so the **FunctionalComponent** for that product would have a **direction** of “output.” The CDS encoding the product, however, is not intended to interact directly, and so its **FunctionalComponent** would have a **direction** of “neither.” Finally, in some cases a **FunctionalComponent** may serve as both an input and output, and be marked as having a **direction** of “both.”

Serialization

The serialization of **FunctionalComponents** has the following form.

```
<sbol:FunctionalComponent rdf:about="...">
  ... properties inherited from identified ...
  one      <sbol:definition rdf:resource="..."> element
  one      <sbol:access rdf:resource="..."> element
  one      <sbol:direction rdf:resource="..."> element
  zero or more <sbol:mapsTo rdf:resource="..."> elements
</sbol:FunctionalComponent>
```

In the example below, the functional component is defined as public input or output. The component refers to the Part:BBa_R0010 promoter from the Parts Registry.

```
<sbol:FunctionalComponent rdf:about="http://sbolstandard.org/example/laci_inverter/promoter">
  <sbol:definition rdf:resource="http://www.partsregistry.org/BBa_R0010"/>
  <sbol:access rdf:resource="http://sbols.org/v2#public"/>
  <sbol:direction rdf:resource="http://sbols.org/v2#inout"/>
</sbol:FunctionalComponent>
```

7.9.2 Module

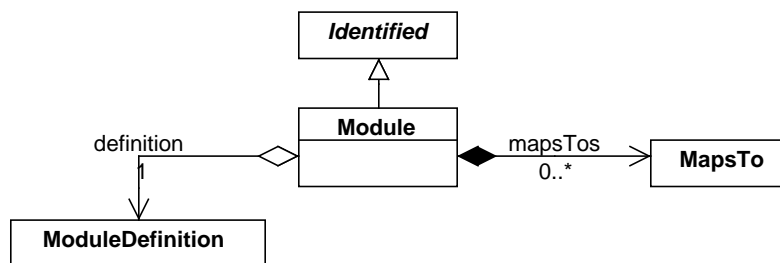


Figure 17: Diagram of the **Module** class and its associated properties.

The **Module** class represents the usage or occurrence of a **ModuleDefinition** within a larger design (that is, another **ModuleDefinition**).

The definition property

The **definition** property is a REQUIRED URI that refers to the **ModuleDefinition** for the **Module**.

The **definition** property MUST NOT refer to the same **ModuleDefinition** as that which contains the **Module**.

Furthermore, **Module** objects MUST NOT form a cyclical chain of references via their **definition** properties and the **ModuleDefinition** objects that contain them. For example, consider the **Module** objects *A* and *B* and the **ModuleDefinition** objects *X* and *Y*. The reference chain “*X* contains *A*, *A* is defined by *Y*, *Y* contains *B*, and *B* is defined by *X*” is cyclical.

The **mapsTo** property

The **mapsTo** property is an OPTIONAL set of **MapsTo** objects that refer to and link **Module** objects together within a larger design.

Section 7.7.3 contains a detailed description of the **MapsTo** class.

Serialization

The serialization of **Modules** has the following form.

```
<sbol:Module rdf:about="...">
  ... properties inherited from identified ...
  one      <sbol:definition rdf:resource="..."> element
  zero or more <sbol:mapsTo>
    <sbol:MapsTo rdf:about="...">...</sbol:MapsTo>
    </sbol:mapsTo> element
</sbol:Module>
```

The example below specifies a TetR inverter that is being used as a part of a toggle switch:

```
<sbol:Module rdf:about="http://sbolstandard.org/example/toggle_switch/tetr_inverter">
  <sbol:definition rdf:resource="http://sbolstandard.org/example/tetr_inverter"/>
  ...
</sbol:Module>
```

7.9.3 Interaction

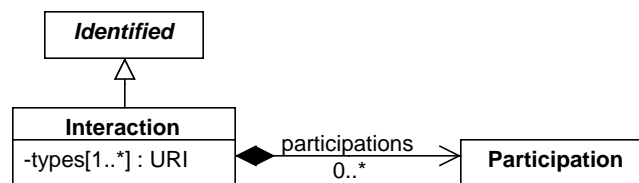


Figure 18: Diagram of the **Interaction** class and its associated properties.

The **Interaction** class provides a description of the functional interactions of entities within a **ModuleDefinition**. For example, it can be used to represent regulatory interactions, such as activation or repression, processes from the central dogma of biology, such as transcription and translation, or molecular interactions like non-covalent binding between a small molecule and transcription factor or phosphorylation of a transcription factor by an enzyme. Such an **Interaction** is represented in SBOL by referring to an ontology defining the type of interaction and declaring how various entities participate in the interaction.

The **types** property

The **types** property is a REQUIRED set of one or more **URI**s that identify an appropriate ontology term describing the behavior represented by this **Interaction**. If an **Interaction** object has multiple **types** **URI**s, then they must

identify synonymous terms.

Values for this URI are RECOMMENDED to be chosen from the occurring entity branch of the Systems Biology Ontology (SBO), where possible. (See <http://www.ebi.ac.uk/sbo/main/>)

The participations property

The `participations` property is an OPTIONAL set of `Participation` objects, each of which identifies the `roles` that the referenced `FunctionalComponent` plays in the interaction.

Note that even though an `Interaction` generally involves at least one `Participation`, the case of zero participations is allowed because it is plausible that a design may wish to specify that an `Interaction` will exist, even if its `participations` are not yet determined.

Serialization

The serialization of `Interaction` objects has the following form.

```
<sbol:Interaction rdf:about="...">
  ... properties inherited from identified ...
  one or more <sbol:type rdf:resource="..." /> elements
  zero or more <sbol:participation>
    <sbol:Participation rdf:about="...">...</sbol:Participation>
  </sbol:participation> elements
</sbol:Interaction>
```

The example below shows an `Interaction` representing an inhibition relationship (SBO:0000169) between a repressor (SBO:0000020, full `Participation` details shown) and a promoter:

```
<sbol:Interaction rdf:about="http://sbolstandard.org/example/laci_inverter/LacI_pLacI">
  <sbol:type rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000169"/>
  <sbol:participation>
    <sbol:Participation rdf:about="http://sbolstandard.org/example/laci_inverter/LacI_pLacI/P03023">
      <sbol:role rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000020"/>
      <sbol:participant rdf:resource="http://sbolstandard.org/example/laci_inverter/TF"/>
    </sbol:Participation>
  </sbol:participation>
  <sbol:participation>
    <sbol:Participation rdf:about="...">
      ...
    </sbol:Participation>
  </sbol:participation>
</sbol:Interaction>
```

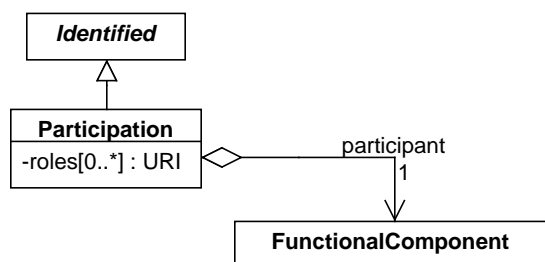


Figure 19: Diagram of the `Participation` class and its associated properties.

7.9.4 Participation

Each **Participation** object describes the role or roles that a particular **FunctionalComponent** plays in its parent **Interaction**.

The *roles* property

The **roles** property is an OPTIONAL set of **URI**s that identify an appropriate ontology term describing this elements relationship to its parent **Interaction**. If a **Participation** object has multiple **roles** **URI**s, then they must identify synonymous terms.

Values for this URI are RECOMMENDED to be chosen from the participant role branch of the Systems Biology Ontology (SBO) where possible. (See <http://www.ebi.ac.uk/sbo/main/>)

The *participant* property

The **participant** property MUST specify precisely one **FunctionalComponent** object that plays the designated **roles** in its parent **Interaction** object.

Serialization

The serialization of **Participation** objects has the following form.

```
<sbol:Participation rdf:about="...">
  ... properties inherited from identified ...
  zero or more <sbol:role rdf:resource="..."> elements
  one <sbol:participant rdf:resource="..."> element
</sbol:Participation>
```

In the example below, the role of participating **FunctionalComponent** is defined to be **inhibitor**, using the SBO:0000020 term. This component is specified using the participant property of the **Participation** entity.

```
<sbol:Participation rdf:about="http://sbolstandard.org/example/laci_inverter/LacI_placI/P03023">
  <sbol:role rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000020"/>
  <sbol:participant rdf:resource="http://sbolstandard.org/example/laci_inverter/TF"/>
</sbol:Participation>
```

7.10 Collection

The **Collection** class is a class that groups together a set of **TopLevel** objects that have something in common. Some examples of **Collection** objects:

- Results of a query to find all **ComponentDefinition** objects that function as promoters in a repository.
- A set of **ModuleDefinition** objects representing a library of NAND gates.
- A **ModuleDefinition** for a complex design, and all of the **ModuleDefinition**, **ComponentDefinition**, **Sequence**, and **Model** objects used to provide its full specification.

The *members* property

The **members** property has a data type of **URI** and has the URI for a **TopLevel** entity. A **Collection** may have any number of members, including none.

Serialization

The serialization of **Collection** objects has the following form:

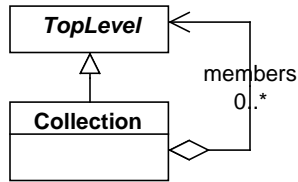


Figure 20: Diagram of the [Collection](#) class and its associated properties.

```

<sbol:Collection rdf:about="...">
  ... properties inherited from identified ...
  zero or more <sbol:member rdf:resource="..."> element
</sbol:Collection>

```

The example below shows the serialization of a [Collection](#) object grouping together a library of constitutive promoters.

```

<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:prov="http://www.
w3.org/ns/prov#" xmlns:sbol="http://sbols.org/v2#">
  <sbol:Collection rdf:about="http://parts.igem.org/Promoters/Catalog/Anderson">
    <sbol:persistentIdentity rdf:resource="http://parts.igem.org/Promoters/Catalog/Anderson"/>
    <sbol:displayId>Anderson</sbol:displayId>
    <dcterms:title>Anderson promoters</dcterms:title>
    <dcterms:description>The Anderson promoter collection</dcterms:description>
    <sbol:member rdf:resource="http://partsregistry.org/Part:BBa_J23119"/>
    ...
    <sbol:member rdf:resource="http://partsregistry.org/Part:BBa_J23118"/>
  </sbol:Collection>
</rdf:RDF>

```

7.11 SBOL Extension Mechanism

SBOL does not attempt to represent all information about a biological system, since many things do not yet have a clear “right way” to be represented, such as design intent, biological context, or performance data. Instead, SBOL allows the embedding of application specific data that are not captured by the SBOL standard. Such data are optional, but can be computationally generated and exchanged via SBOL documents without getting damaged or lost. This SBOL extension mechanism is designed to allow easy incorporation within the SBOL standard once there is community agreement on data content to be exchanged.

To do this, SBOL provides an “annotation” mechanism for attaching arbitrary information to SBOL objects, which allows SBOL models to be connected with any other models in an extensible manner. In particular, several methods are supported for connecting the SBOL data model with other, possibly application-specific data:

1. Information that is “part” of an SBOL object (i.e., a “filled diamond” relationship) is annotated simply by adding non-conflicting properties and custom entries to an SBOL object. An example might be source information about the registry from which a [ComponentDefinition](#) was imported.
2. Information that is an independent object is annotated by wrapping it inside of a [GenericTopLevel](#) object. An example might be a data sheet describing the performance of a [ModuleDefinition](#) in some particular context.
3. Conversely, rather than embedding external objects in SBOL, SBOL objects can also be linked to external data. The only requirement is that some URI resolution mechanism must be available that allows the links from SBOL objects to be followed when needed.

- Finally, just as external objects can be embedded in SBOL, external documents can embed or refer to SBOL objects. This last case needs no explicit support from this specification (it depends only on the external non-SBOL system managing its own relations to SBOL), and is included here for completeness.

7.11.1 Annotating SBOL objects

Each **Identified** object may have a number of annotations in the form of name/value property pairs. The **qName** property is specified by a qualified name (**QName**), which is composed of a namespace, a prefix, and a local name. The **qName** property must be stored in the data model to allow for proper serialization as described below. The **value** property can be a literal type (i.e., **String**, **Integer**, **Double**, **Boolean**), **URI**, or a **NestedAnnotations** object. The **NestedAnnotations** object is composed of a **nestedQName**, **nestedURI**, and an optional list of nested **annotations**.

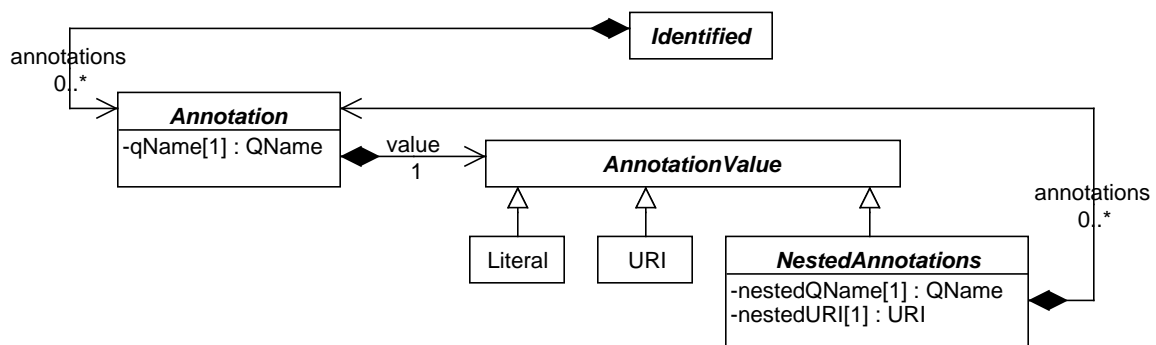


Figure 21: Diagram of the **Annotation** class and its association with **Identified** and **AnnotationValue** objects, which is used for annotating SBOL entities with application specific data.

Serialization

The serialization of **Annotation** objects has the following form:

```

<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:sbol="http://sbols.org/v2#"
  xmlns:prefix1="NAMESPACE_1"
  xmlns:prefix2="NAMESPACE_2"
  xmlns:nestedObjectPrefix="A_NESTED_OBJECT_NAMESPACE"
  ...
>
<sbol:A_TOPLEVELOBJECT rdf:about="...">
  ...
  zero or more <prefix1:LOCAL_NAME_1>A_LITERAL</prefix1:LOCAL_NAME_1> elements
  zero or more <prefix1:LOCAL_NAME_2 rdf:resource=URI/> elements
  zero or more <prefix2:LOCAL_NAME_3>
    <nestedObjectPrefix:NESTED_LOCAL_NAME rdf:about="...">
      ...
    </nestedObjectPrefix:NESTED_LOCAL_NAME>
  </prefix2:LOCAL_NAME_3> elements
</sbol:A_TOPLEVELOBJECT>

```

The **qName** property specifies the namespace, prefix, and localPart values. The use of such qualified names is described in detail by the w3c here:

<http://www.w3.org/TR/1999/REC-xml-names-19990114/#ns-using>

Essentially, the "xmlns" statement defines the prefix string to use as an alias for the namespace. The prefix can be

any arbitrary string, and its use is optional, since it simply replaces the full namespace making the serialization more readable.

The first annotation shown above is for a [literal](#) annotation. The second form is for a [URI](#) annotation. Finally, the third form is for an [NestedAnnotations](#) object annotation. In this last case, the [nestedQName](#) property specifies the nestedNamespace, nestedPrefix, and nestedLocalPart while the [nestedURI](#) property species the URI for the nested annotation.

The ComponentDefinition example for a promoter serialized below shows how annotations can be added to SBOL objects. Annotations are added using the relevant information from the Parts Registry. Annotation property names are qualified with the <http://www.partsregistry.org/> namespace, which is prefixed using **pr**. The first annotation is named as **pr:group**, indicating the iGEM group designing the promoter, and has a [String](#) value. The second **pr:experience** annotation has a [URI](#) value and is serialised as an RDF resource; in this case, the identifier also happens to be able to be resolved to the information Web page on the Parts Registry for the promoter. The **pr:information** property represents a complex annotation which is a type of **pr:Information** and includes information about the regulatory details of the promoter using Parts Registry categories.

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:pr="http://partsregistry.org" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:prov="http://www.w3.org/ns/prov#" xmlns:sbol="http://sbols.org/v2#">
  <sbol:ComponentDefinition rdf:about="http://partsregistry.org/cd/BBa_J23119">
    <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_J23119"/>
    <sbol:displayId>BBa_J23119</sbol:displayId>
    <pr:group>iGEM2006_Berkeley</pr:group>
    <pr:experience rdf:resource="http://parts.igem.org/cgi/partsdb/part_info.cgi?part_name=BBa_J23119"/>
    <pr:information>
      <pr:Information rdf:about="http://parts.igem.org/cgi/partsdb/part_info.cgi?part_name=BBa_J23119">
        <pr:sigmafactor>//rnap/prokaryote/ecoli/sigma70</pr:sigmafactor>
        <pr:regulation>//regulation/constitutive</pr:regulation>
      </pr:Information>
    </pr:information>
    <dcterms:title>J23119</dcterms:title>
    <dcterms:description>Constitutive promoter</dcterms:description>
    <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
    <sbol:role rdf:resource="http://identifiers.org/so/SO:0000167"/>
  </sbol:ComponentDefinition>
</rdf:RDF>
```

7.11.2 GenericTopLevel

SBOL documents can also be annotated at the top level. SBOL's [GenericTopLevel](#) is a top-level entity whose only purpose is to include a set of annotations as described above. Entities that have independent existence (i.e., would be another “top level” class) and are not recognized by the SBOL standard are loaded into these top level entities. These [GenericTopLevel](#) entities can thus be safely used by tools to exchange non-SBOL data embedded separately within SBOL. As with any other top level entities, [GenericTopLevel](#) entities may include SBOL properties such as [displayId](#), [name](#), [description](#), etc. The type of data found in the generic entity is indicated using the [rdfType](#) property which is of type [QName](#). Again, the [rdfType](#) property is used to set the prefix and localPart fields during serialization.

Serialization

The serialization of [GenericTopLevel](#) objects has the following form where the prefix, namespace, and localPart are defined by the [rdfType](#) property:

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

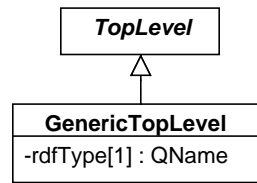


Figure 22: Diagram of the `GenericTopLevel` class and its associated properties, which is used for embedding externally defined entities into SBOL documents.

```

xmlns:sbol="http://sbols.org/v2#"
xmlns:applicationPrefix="APPLICATION_NAMESPACE"
...
>
<applicationPrefix:APPLICATION_OBJECT_NAME rdf:about="...">
  ... properties inherited from identified ...
  ... any non-conflicting application-specific properties ...
</applicationPrefix:APPLICATION_OBJECT_NAME>

```

The example below shows how a datasheet object can be added to an SBOL document using the `GenericTopLevel` class. The J23119 promoter example is annotated with the URI of a top Level Datasheet object, here defining the annotation properties using the custom `http://www.myapp.org/` namespace and the `myapp` prefix. The datasheet object, with the data type of `myapp:Datasheet`, is accessed using the URI specified by the `myapp:characterizationData` property of the promoter `ComponentDefinition`. The datasheet object is further annotated with the transcription rate and URI for the actual characterization data using the `myapp:transcriptionRate` and `myapp:characterizationData` properties, respectively. Finally, this data sheet is linked from the component it describes using an annotation with a `myapp:datasheet` property whose value is the datasheet's URI.

```

<?xml version="1.0" ?>
<rdf:RDF xmlns:myapp="http://www.myapp.org/" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:prov="http://www.w3.org/ns/prov#" xmlns:sbol="http://sbols.org/v2#">
  <sbol:ComponentDefinition rdf:about="http://www.partsregistry.org/cd/BBa_J23119">
    <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/cd/BBa_J23119"/>
    <sbol:displayId>BBa_J23119</sbol:displayId>
    <prov:wasDerivedFrom rdf:resource="http://www.partsregistry.org/Part/BBa_J23119"/>
    <myapp:datasheet rdf:resource="http://www.partsregistry.org/gen/datasheet1"/>
    <dcterms:title>J23119</dcterms:title>
    <dcterms:description>Constitutive promoter</dcterms:description>
    <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
    <sbol:role rdf:resource="http://identifiers.org/so/SO:0000167"/>
  </sbol:ComponentDefinition>
  <myapp:Datasheet rdf:about="http://www.partsregistry.org/gen/datasheet1">
    <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/gen/datasheet1"/>
    <sbol:displayId>datasheet1</sbol:displayId>
    <myapp:characterizationData rdf:resource="http://www.myapp.org/measurement/1"/>
    <myapp:transcriptionRate>1</myapp:transcriptionRate>
    <dcterms:title>Datasheet 1</dcterms:title>
  </myapp:Datasheet>
</rdf:RDF>

```


8 Data Model Examples

This section illustrates how to use the SBOL data model by specifying the design of a LacI/TetR toggle switch similar to those constructed in [Gardner et al. \(2000\)](#). This design is visualized conceptually in [Figure 23](#) and in detail in [Figure 24](#).

Conceptually, the toggle switch is constructed from two mutually repressing genes. With repressors LacI and TetR, this results in a bi-stable system that will tend to settle into a state where precisely one of the two repressors is strongly expressed, repressing the other. Each of these repressors can have its activity disrupted by a small molecule (IPTG for LacI, aTc for TetR), which allows the system to be “toggled” from one state to the other by dosing it with the appropriate small molecule.

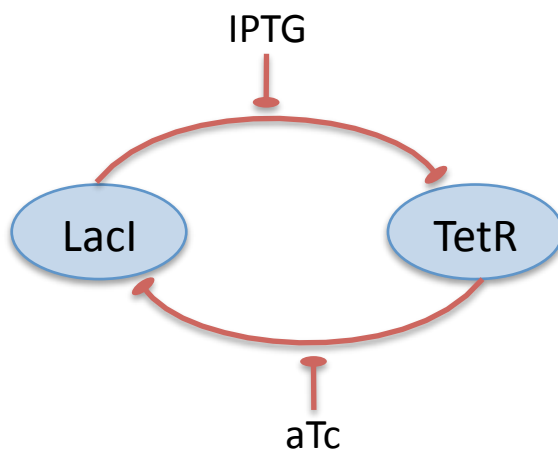


Figure 23: Conceptual diagram of LacI/TetR toggle switch: the LacI and TetR transcriptional factors are arranged to mutually repress, creating a bi-stable system. Transition between the two states is triggered by the small-molecule signals aTc (which disrupts TetR repression) and IPTG (which disrupts LacI repression).

The LacI/TetR toggle switch is modeled in SBOL as two parallel hierarchies of structure and function. The structural hierarchy of the toggle switch is represented using [ComponentDefinitions](#):

- The base elements of the hierarchy are DNA components, transcription factor proteins, and small molecules. As an example, [Figure 25](#) is a UML diagram of the [ComponentDefinitions](#) that represent these elements.
- Base elements are composed to form more complex structures at the top of the hierarchy, including genes and non-covalent complexes between transcription factor proteins and small molecules. As an example, [Figure 26](#) is a UML diagram of the composite [ComponentDefinitions](#) that represent the TetR gene and IPTG-LacI complex.

The functional hierarchy of the toggle switch is represented using [ModuleDefinitions](#):

- The base elements of the hierarchy are LacI-dependent repression of TetR expression (the LacI inverter) and TetR-dependent repression of LacI (the TetR inverter). As an example, [Figure 27](#) is a UML diagram of the [ModuleDefinition](#) that represents the LacI inverter.
- Base elements are composed to form the toggle switch at the top of the hierarchy. As an example, [Figure 28](#) is a UML diagram of the [ModuleDefinition](#) that represents the toggle switch.

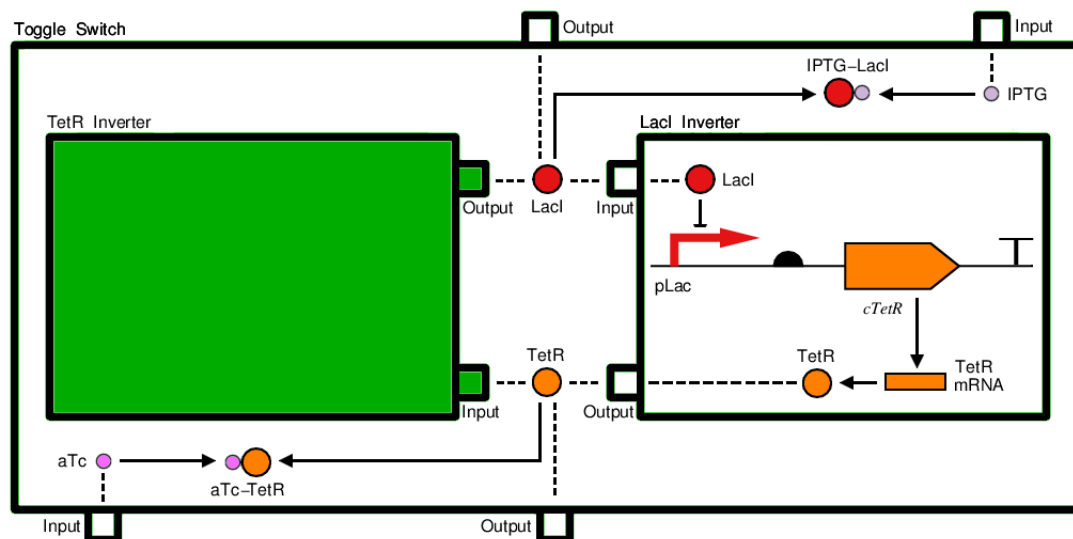


Figure 24: Design of a LacI/TetR toggle switch. This design is composed of two inverter sub-designs, each containing a single gene. These genes mutually repress each other's expression via their encoded protein transcription factors, LacI and TetR. Furthermore, both LacI and TetR are bound by specific small molecules that sequester them and prevent them from acting as repressors. In this design, arrows represent different molecular interactions, including the repression of pLac via LacI, the non-covalent binding of IPTG to LacI, the transcription of TetR mRNA, and the translation of TetR. Dashed lines serve to map between transcription factors in the inverter sub-designs and those in the overall toggle switch design.

ComponentDefinition.types in the following figure are not consistent with the list of BioPax ontological terms described previously in the Data Model section. Should now be consistent. - Nic

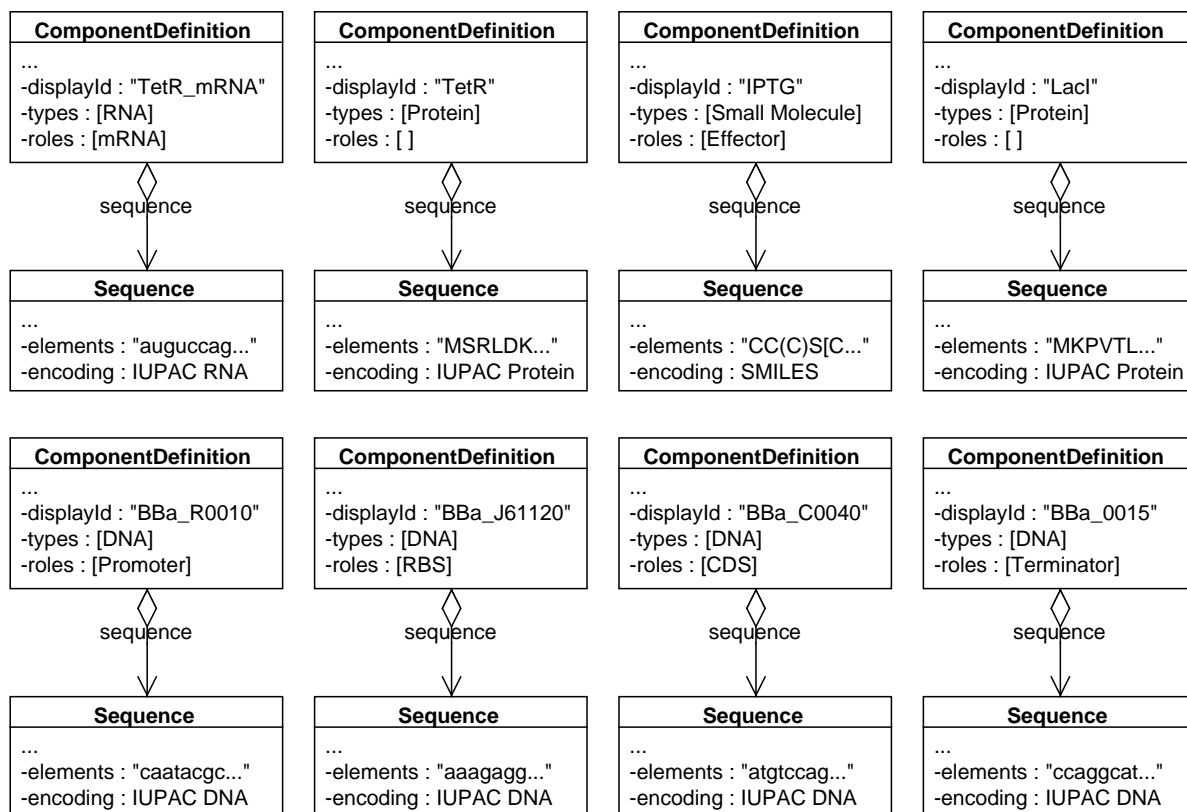


Figure 25: **ComponentDefinition** objects for the LacI inverter. These include **ComponentDefinition** objects based on DNA parts from the iGEM Registry and **ComponentDefinitions** that represent TetR mRNA, TetR, LacI, and IPTG. Each **ComponentDefinition** is associated with a **Sequence** that has an IUPAC DNA/RNA or IUPAC protein encoding, except the **ComponentDefinition** of IPTG, which is associated with a **Sequence** that has a SMILES encoding.

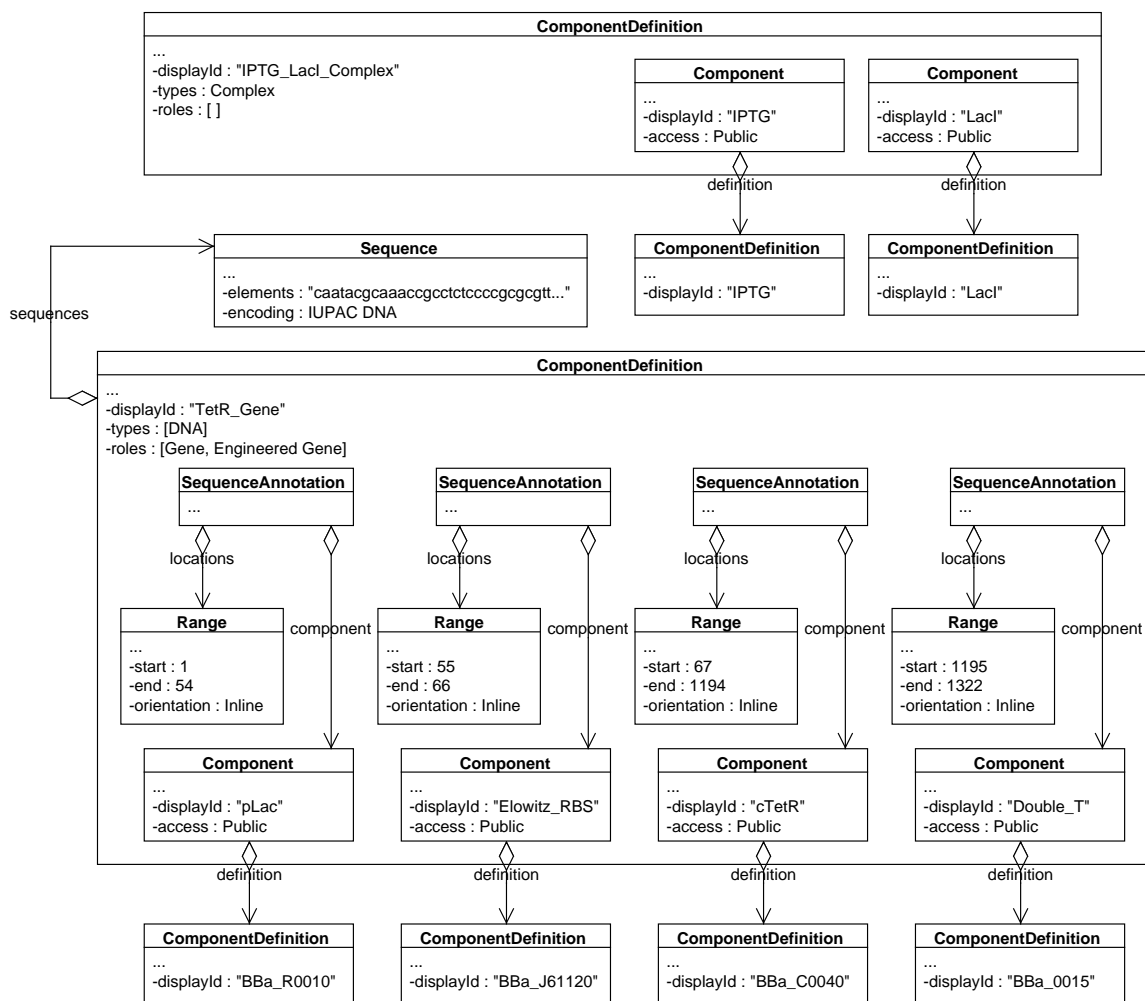


Figure 26: Composite **ComponentDefinition** objects for the LacI inverter. In the case of the **ComponentDefinition** that represents the TetR gene, its sub-**Component** objects are located as **Ranges** along its **Sequence** using **SequenceAnnotations**. The **ComponentDefinition** that represents the IPTG-LacI complex, however, has no **Sequence** and its sub-**Component** objects are composed without any data about their relative positions.

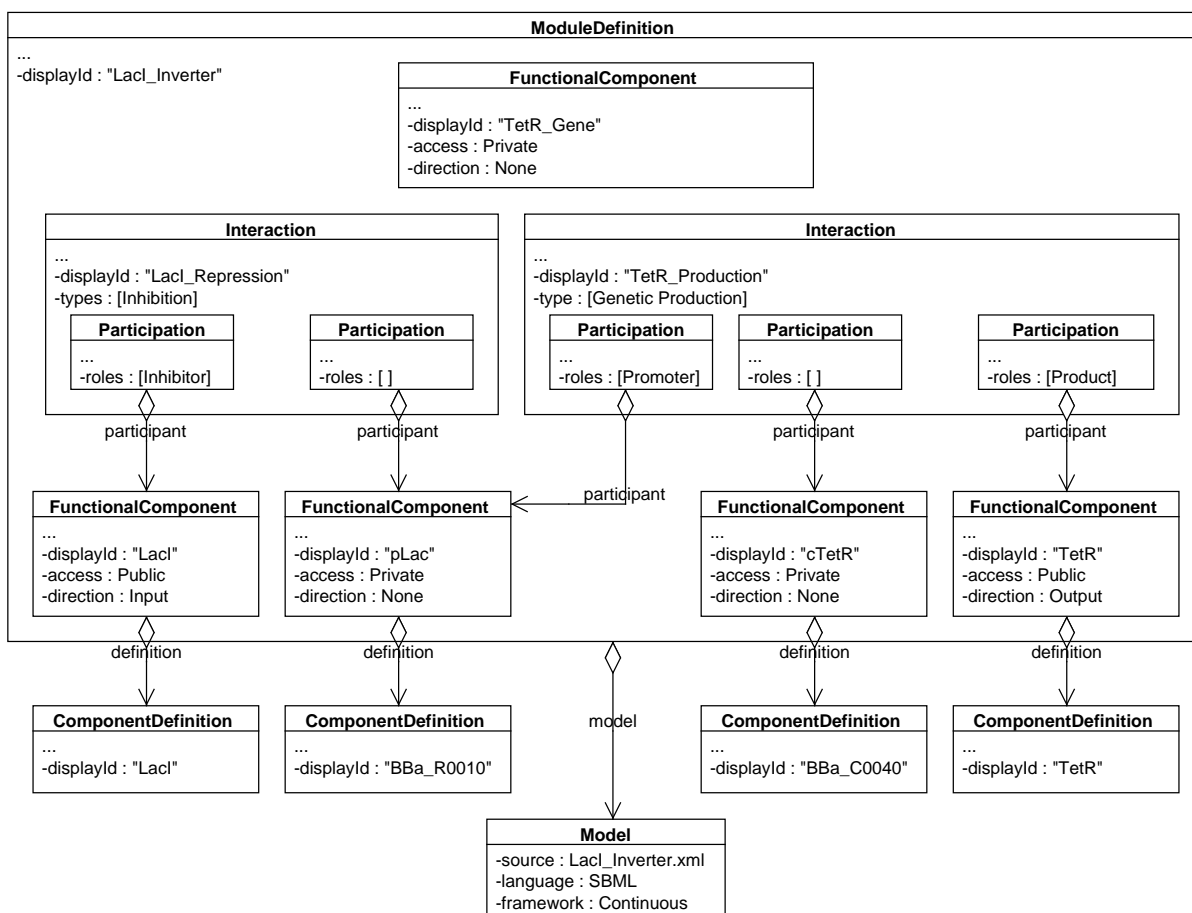


Figure 27: ModuleDefinition of the LacI inverter. This **ModuleDefinition** contains **FunctionalComponent** objects that instantiate the **ComponentDefinition** objects for the LacI/TetR transcription factors and TetR gene. The **FunctionalComponent** for the TetR gene as a whole does not participate in any **Interaction** and merely indicates the overall structure that is functionally described by the LacI inverter **ModuleDefinition**. The remaining **FunctionalComponent** objects participate in a repression **Interaction** and a genetic production **Interaction**, thereby indicating which parts of the overall structure carry out the function of the LacI inverter **ModuleDefinition**. In this case, the transcription and translation of TetR are represented as a single genetic production **Interaction** that abstracts away the presence of the intermediate TetR mRNA. In addition, this **ModuleDefinition** is also associated with a continuous **Model** written in the SBML source file "LacI_Inverter.xml."

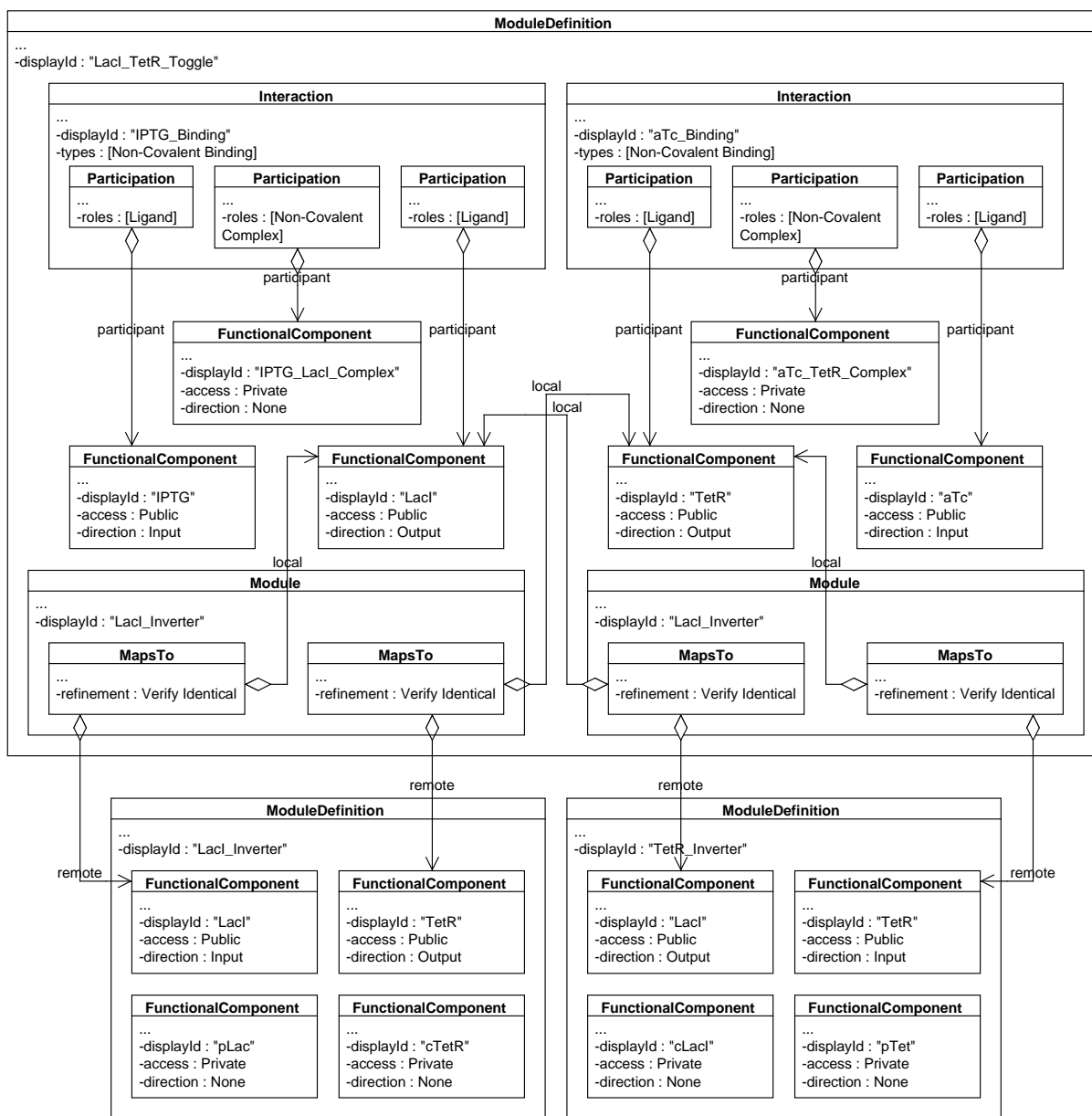


Figure 28: Composite **ModuleDefinition** of the LacI/TetR toggle switch. This **ModuleDefinition** contains the **Module** objects that instantiate **ModuleDefinition** object for the LacI and TetR inverter. It also contains **FunctionalComponent** objects that instantiate the **ComponentDefinition** objects for the LacI/TetR transcription factors and IPTG/aTc small molecules. These **FunctionalComponent** objects each participate in a non-covalent binding **Interaction**. To complete the composition of the toggle switch, **MapsTo** objects are used to indicate that the output of the LacI inverter **ModuleDefinition** is identical to the input of the TetR inverter **ModuleDefinition** and vice versa.

9 SBOL RDF Serialization

In order for SBOL objects to be readily stored and exchanged, it is important that they be able to be *serialized*, i.e., converted to a sequence of bytes that can be stored in a file or exchanged over a network. The serialization format for SBOL is designed to meet several competing requirements. First, SBOL needs to support ad-hoc annotations and extensions. Second, SBOL needs to support processing by general database and semantic web software tools that have little or no knowledge of the SBOL data model. Finally, it should be relatively simple to write a new software implementation, so that SBOL can be readily used even in software environments where community-maintained implementations are not available.

To meet these goals, the canonical serialization of SBOL has been selected to be a strict dialect of RDF/XML [Beckett and McBride \(2004\)](#), a syntactic standard defined for Semantic Web data exchange. This serialization provides a standard base from which to meet further requirements. Moreover, it allows any RDF/XML-aware software tool to consume and operate on an SBOL file without needing any customization to support SBOL. Where possible, we have re-used predicates from widely-used terminologies (such as Dublin Core [DCMI Usage Board \(2012\)](#)) to expose as much of the data as practical to such standard RDF tooling.

Arbitrary RDF/XML, however, provides a sometimes problematically large amount of flexibility in how equivalent data can be serialized. This flexibility can result in different serializations when processing RDF/XML files using standard off-the-shelf XML tools, such as DOM-OO mappings. To address this issue, we define a canonical association between the nesting of data structures within the SBOL UML data model and the RDF/XML file. For all ownership associations (filled diamonds), the RDF/XML for the owned entity is embedded within the owner's RDF/XML (note, however, that the property values may be listed in any order). For all associations that are by reference (open diamonds), the RDF/XML for the referenced property is linked via a resource URI. For example, the serialization of a [ComponentDefinition](#) embeds the serializations of the [SequenceConstraint](#) and [Component](#) objects associated with it. Those [SequenceConstraint](#) objects, however, link to the [Component](#) objects with a URI rather than embedding another copy.

Every SBOL document must be a valid RDF/XML document. Accordingly, each SBOL document starts with an XML declaration that has its XML version set to "1.0." As shown in the example below, this declaration is then followed by an `rdf:RDF` XML element that includes the namespace declarations for RDF, Dublin Core, PROV, and SBOL. The SBOL namespace, which is <http://sbols.org/v2#>, is used to indicate which entities and properties in the SBOL document are defined by SBOL, and SHOULD NOT be used for any entities or properties not defined in this specification.

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:prov="http://www.
  w3.org/ns/prov#" xmlns:sbol="http://sbols.org/v2#">
...
</rdf:RDF>
```

All first-class SBOL data types (i.e., those enumerated in [Section 7](#)) have an associated identifying URI. In the RDF, this is the resource URI used by instances of that type. For example, [ComponentDefinition](#) has the type URI `sbol:ComponentDefinition`. Properties and associations are then asserted as nested RDF/XML assertions. [Section 7](#) provides the serialization template and an example at the end of its description of each data type. All of the data types that are [TopLevel](#) are so named because they always appear at the top-most level of the RDF/XML serialization. All other datatypes will always appear nested within their parent container and, ultimately, some [TopLevel](#) object. For example, a [ComponentDefinition](#) is a [TopLevel](#) and therefore listed at the top-most level of the RDF/XML serialization, and contains its [SequenceConstraint](#) objects, since they are not [TopLevel](#). Its [Sequence](#), however, is also [TopLevel](#) and is therefore not nested within and instead linked via a URI.

Each instance of a first-class SBOL datatype may also have annotations attached, as described in [Section 7.11](#). These annotations are composed of a name and a value. They are serialized to RDF as a conceptual triple with the subject being the identity of the instance they annotate, the predicate being the name of the annotation, and the

object being the value of that annotation. Annotation values are always nested within the RDF/XML serialization of the instance that they annotate. For example, a [ModuleDefinition](#) might add a DOI annotation that links to the scientific article that first described the system that it represents.

SBOL also supports top-level, user-defined annotations, again as described in [Section 7.11](#). This is to allow non-standardized but necessary information to be carried around as part of a design. For example, a particular sub-community may have an internal standard for genetic device characterization data sheets. Such data can be represented as a [GenericTopLevel](#) object with internal structured annotations. For example, each individual data sheet might be contained in its own [GenericTopLevel](#) instance. This annotation will be serialized into the RDF/XML in the usual way, as an RDF/XML block at the top level of the file. Other objects may refer to this entity through their annotations by reference, and this generic top-level entity may refer to other entities via references. For example, a [ModuleDefinition](#) might use an annotation to refer to the data sheet [GenericTopLevel](#) that documents its properties.

By adopting this paradigm of RDF/XML serialization, SBOL is able to adapt to future changes in the standard without requiring large-scale alterations to the RDF files. Since exactly the same scheme is used to serialize annotations as is used to serialize specification-defined properties and associations, it is possible to update the SBOL standard to recognize a different range of properties and associations. Those properties not recognized by the specification will always be available through the API as annotations. Similarly, by allowing arbitrary top-level entities in a SBOL file, we enable future specifications or extensions to ratify the structure of other top-level objects. These entities would then become part of the explicit data model, but the identical RDF serialization would be used. Applications lacking support for a given extension can safely read in, manipulate, and write out the top-level data that is not understood, treating it as a top-level structured annotation, without data loss or corruption. Finally, the very regimented control of nesting versus referencing also allows the XML structure to be very predictable, enabling XML/DOM-based tooling to work with SBOL RDF/XML files safely.

10 Recommended Best Practices

10.1 Use of the Version Property

Once an SBOL object has been published where others might have accessed it (e.g., to an online repository), it may be the case that others make copies of the object or else come to depend on the particular contents of the object. Thus, in order to avoid confusion, if a person wants to change the properties of a published object, the best practice is to do so by making a new copy that incorporates the change, with a new URI.

The relationship between the old and new objects (i.e., that the new object was derived from the old object), however, is not visible unless it is explicitly declared. This is RECOMMENDED to be done using the `persistentIdentity`, and `version` properties. The preferred practice for declaring such a relationship is to use the same `persistentIdentity` for both objects, and label the newer one as being the newer version. Then, when the new object is published, it can be clear to both humans and machines that this object is intended to replace the one that was published previously. In this way also, when the user of an object always wants the latest version, they can obtain it by referencing the object with the `persistentIdentity` rather than the `identity`.

As stated in [Section 7.4](#), it is RECOMMENDED that version numbering should follow the conventions of semantic versions (<http://semver.org/>), particularly as implemented by Maven (<http://maven.apache.org/>). This convention represents versions as sequences of numbers and qualifiers separated by the characters `.` and `-` and compared in lexicographical order (for example, `1 < 1.3.1 < 2.0-beta`). For a full explanation, see the linked resources.

Annotations: Annotating with created and modified dates, and how to add them. Is this section needed? – CJM

10.2 Compliant SBOL Objects

Maintaining unique identity URIs for all SBOL objects is a very challenging implementation task. To reduce the developer's burden, users of SBOL 2.0 are encouraged to follow a few simple rules when constructing the identity and related fields for SBOL objects. When these rules are followed, we say that the SBOL object is *compliant*. The rules are as follows:

1. The `identity` of an SBOL object should begin with a *URI prefix* that maps to a domain over which the user has control. Namely, the user can guarantee uniqueness of identities within this domain.
2. In a compliant SBOL object, the `persistentIdentity` and `displayId` properties are required.
3. The `persistentIdentity` of a compliant top level object must end with a delimiter (`'/'`, `'#'`, or `':'`) followed by the `displayId` of the object.
4. The `persistentIdentity` of a compliant child object must begin with the `persistentIdentity` of its parent object and be immediately followed by a delimiter (`'/'`, `'#'`, or `':'`) followed by the `displayId` of the object.
5. When a SBOL object is not given a `version`, the `identity` and `persistentIdentity` must be equal.
6. When a SBOL object is given a `version`, the `identity` must be equal to the "`<persistentIdentity>/<version>`".
7. The `version` of a compliant child object must be equal to the `version` of its parent object.
8. The `identity`, `persistentIdentity`, `displayId`, and `version` of a compliant object once set can never be changed.

Mike R: “The version of a compliant child object must be equal to the version of its parent object.” There is an alternative. Consider what happens in the following scenario: suppose my parent object is ‘dad’, and its two children are ‘daughter’ and ‘son’. dad starts at v1.0, and likewise daughter starts at v1.0 and son is v1.0. Now I want to revise just the son, so I give him a new haircut and I increment his version to v1.1. Since the dad’s child version changed, then dad needs to change as well, so let’s increment his version to v1.1. However, there is no reason why daughter’s version should change to v1.1. Daughter is exactly the same as she was at v1.0, and retaining her old version saves us the trouble of diffing v1.0 and the identical v1.1. In other words, version changes should be minimally promoted up the tree, but not back down to siblings and relatives if they didn’t change.

If you don’t want to do it this way, I do see why, but I’ve found it helpful on past projects to increment versions only on the things that contain diffs (including their containers, recursively). That’s the way that many dependency management systems work, as well as certain aspects of version control: when you ask for a log on an individual file in a version control system, you don’t see every revision that ever touched the repo, you only see the revisions that modified that file.”

ANSWER: I can see your point. It would, however, complicate a few things in our library implementation, and it would be very difficult to change at this point. We can consider this for a future version.

For examples, see any example in this specification, as all have been formulated using compliant URIs.

10.3 Annotations: Embedded Objects vs. External References

When annotating an SBOL model with additional information, there are two general methods that can be used:

- Embed the information in the SBOL model, either as non-SBOL properties or wrapped in [GenericTopLevel](#) objects.
- Store the information separately and annotate the SBOL model with URIs that point to it.

In theory, either method can be used in any case. (Note that a third case not discussed here is to use SBOL to annotate external objects by linking to SBOL documents rather than from the external objects.)

In practice, embedding massive amounts of non-SBOL data into SBOL models is likely to cause problems for people and software tools trying to manage and exchange such models. Therefore, it is RECOMMENDED that small amounts of information (e.g., design notes, preferred graphical layout) be embedded in the SBOL model, while large amounts of information (e.g., the contents of the scientific publication from which a model was derived, flow cytometry data characterizing performance) be linked with URIs pointing to external resources. The boundary between “small” and “large” is left deliberately vague, recognizing that it will likely depend on the particulars of a given SBOL application.

10.4 Completeness and Validation

RDF documents containing serialized SBOL objects may or may not be entirely self-contained. A SBOL document is self-contained or “complete” if every SBOL object referred to in the document is contained in the document. It is RECOMMENDED that serializations be complete whenever practical. In other words, when serializing an SBOL object, serialize all of the other objects that it points to, then serialize all of the additional objects that they point to, etc., until the document is complete.

It is important to note that there is no guarantee that an RDF document contains valid SBOL. When an RDF document is de-serialized into SBOL objects, the program doing so SHOULD verify that all of the property values encoded therein have the right type (e.g., that the object pointed to by the [sequences](#) property of a [ComponentDefinition](#) really is a [Sequence](#) object). For complete files, this can be carried out readily and entirely locally. For files that are not complete, an implementation needs to either have a means of validating those external references (e.g., by retrieving them from various repositories) or else to mark them as being unverified and not depend on their correctness.

Mike B. “the program doing [the deserialization] SHOULD verify that all of the property values encoded therein have the right type...” Not everybody knows how to apply XML+XSD. Maybe we should give new users a big hint: they can just use any correctly implemented SBOL 2.0 XSD file along with a validating XML parser to do this stuff automatically. We really don't want implementators getting scared off by the burden imposed by this section of the spec. Let's perhaps additionally recommend as a bootstrapping exercise to grab an SBOL 2.0 XSD file out of a recognized reference implementation (libSBOL, for instance), then start by feeding the XSD and the user's SBOL document in question to a validating XML parser such as xmllint, which is FOSS, works great, and installs in seconds. (Side note: if you choose to give these helpful hints, please make it clear that a given reference implementation is not a spec; if it contains errors, the spec wins.) A section about completeness and validation that doesn't talk about the available official or unofficial XSDs is incomplete. This part was left out of SBOL 1.1, but now that serialization is a big part of SBOL 2.0, an XSD should be included somehow, if only by reference.” ANSWER: agreed, we need to do an XSD as soon as possible. Help would be appreciated. -CIM I'll get in touch with the libsbol devs and offer to lend a hand. -MB

10.5 Recommended Ontologies for External Terms

External ontologies and controlled vocabularies are an integral part of SBOL. SBOL utilizes these resources to access existing biological information where possible. New SBOL specific terms are defined only when necessary. Instead, SBOL provides placeholders that can point to external terms. For example, types of components, such as DNA or protein, are indicated using BioPAX. Similarly, the role of a DNA component is indicated via the SO terms. Although preferred ontologies have been indicated in relevant sections where possible, other resources providing similar terms can also be used. A summary of these external sources can be found in [Table 11](#).

SBOL Entity	Property	Preferred External Resource	More Information
ComponentDefinition	types	BioPAX	http://www.biopax.org
	roles	SO (<i>DNA</i> or <i>RNA</i>)	http://www.sequenceontology.org
	roles	CHEBI (<i>small molecule</i>)	https://www.ebi.ac.uk/chebi/
Interaction	types	SBO (occurring entity branch)	http://www.ebi.ac.uk/sbo/main/
Participation	roles	SBO (participant roles branch)	http://www.ebi.ac.uk/sbo/main/
Model	language	EDAM	http://bioportal.bioontology.org/ontologies/EDAM
	framework	SBO (modeling framework branch)	http://www.ebi.ac.uk/sbo/main/

Table 11: Preferred external resources from which to draw values for various SBOL properties.

Goksel and Neil need to sort out GO vs. UniProt, and possibly just recommend both here.

References

- Beckett, D. and McBride, B. (2004). RDF/XML syntax specification (revised). *W3C recommendation*, 10.
- Canton, B., Labno, A., and Endy, D. (2008). Refinement and standardization of synthetic biological parts and devices. *Nature Biotechnology*, 26(7):787–793.
- DCMI Usage Board (2012). DCMI metadata terms. DCMI recommendation, Dublin Core Metadata Initiative.
- Finney, A., Hucka, M., Bornstein, B. J., Keating, S. M., Shapiro, B. M., Matthews, J., Kovitz, B. K., Schilstra, M. J., Funahashi, A., Doyle, J., and Kitano, H. (2006). Software infrastructure for effective communication and reuse of computational models. In Szallasi, Z., Stelling, J., and Periwal, V., editors, *System Modeling in Cell Biology: From Concepts to Nuts and Bolts*. MIT Press.
- Galdzicki, M., Clancy, K. P., Oberortner, E., Pocock, M., Quinn, J. Y., Rodriguez, C. A., Roehner, N., Wilson, M. L., Adam, L., Anderson, J. C., et al. (2014). The synthetic biology open language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature biotechnology*, 32(6):545–550.
- Gardner, T. S., Cantor, C. R., and Collins, J. J. (2000). Construction of a genetic toggle switch in *escherichia coli*. *Nature*, 403:339–342.
- Garny, A., Nickerson, D., Cooper, J., dos Santos, R. W., Miller, A., McKeever, S., Nielsen, P., and Hunter, P. (2008). Cellml and associated tools and techniques. *Philos. Transact. A: Math Phys Eng Sci*, 366(1878):3017–43.
- MathWorks (2015). Matlab.
- Norrande, J., Kempe, T., and Messing, J. (1983). Construction of improved m13 vectors using oligodeoxynucleotide-directed mutagenesis. *Gene*, 26:101–106.
- Roehner, N., Oberortner, E., Pocock, M., Beal, J., Clancy, K., Madsen, C., Misirli, G., Wipat, A., Sauro, H., and Myers, C. J. (2015). Proposed data model for the next version of the synthetic biology open language. *ACS synthetic biology*, 4(1):57–71.

A Validation Rules

New validation rules have been added. Needs review.

This section summarizes all the conditions that MUST be or are RECOMMENDED to be true of an SBOL Version 2 document. There are different degrees of rule strictness. Rules of the former kind are strict SBOL validation rules—data encoded in SBOL MUST conform to all of them in order to be considered valid. Rules of the latter kind are consistency rules that are RECOMMENDED for following best practices. To help highlight these differences, we use the following symbols next to the rule numbers:

- ☑ A checked box indicates a strong REQUIRED condition for SBOL conformance. If a SBOL document does not follow this rule, it does not conform to the SBOL specification. (Mnemonic intention behind the choice of symbol: “This must be checked.”)
- ▲ A triangle indicates a weak REQUIRED condition for SBOL conformance. While this rule MUST be followed, it is difficult, if not impossible, for a machine to automatically check whether the rule has been followed. (Mnemonic intention behind the choice of symbol: “This is a cause for warning.”)
- ★ A star indicates a RECOMMENDED condition for following best practices. This rule is not strictly a matter of SBOL conformance, but its recommendation comes from logical reasoning. If an SBOL document does not follow this rule, it is still valid SBOL, but it may have degraded functionality in some tools. (Mnemonic intention behind the choice of symbol: “You’re a star if you heed this.”)

The validation rules listed in the following subsections should all be stated or implied in the rest of this specification document. They are enumerated here for convenience and to provide a “master checklist” for SBOL compliance. In case of a conflict between this section and other portions of the specification (though there should be none), this section is considered authoritative for purpose of determining SBOL document compliance.

☞ For convenience and brevity, we use the shorthand “**sbol:x**” to stand for an attribute or element name **x** in the namespace for the SBOL specification, using the namespace prefix **sbol**. In reality, the prefix string may be different from the literal “**sbol**” used here (and indeed, it can be any valid XML namespace prefix that the software chooses). We use “**sbol:x**” because it is shorter than to write a full explanation everywhere we refer to an attribute or element in the SBOL specification namespace.

General rules about an SBOL document

- sbol-10101** ☑ An SBOL document MUST declare the use of the following XML namespace:
“<http://sbols.org/v2#>”.
Reference: [Section 9 on page 55](#)
- sbol-10102** ☑ An SBOL document MUST declare the use of the following XML namespace:
“<http://www.w3.org/1999/02/22-rdf-syntax-ns#>”.
Reference: [Section 9 on page 55](#)
- sbol-10103** ☑ An SBOL document MUST declare the use of the following XML namespace when it includes any [name](#) or [description](#) properties:
“<http://purl.org/dc/terms/>”.
Reference: [Section 9 on page 55](#)
- sbol-10104** ☑ An SBOL document MUST declare the use of the following XML namespace when it includes any [wasDerivedFrom](#) properties:
“<http://www.w3.org/ns/prov#>”.
Reference: [Section 9 on page 55](#)

Rules for the Identified class

- sbol-10201** ✓ The **identity** is a REQUIRED property for all **Identified** objects and has a data type of URI with a syntax defined by:
 “<http://www.w3.org/1999/02/22-rdf-syntax#about>”
 Reference: [Section 7.4 on page 19](#)
- sbol-10202** ✓ The **persistentIdentity** is an OPTIONAL property for all **Identified** objects and, if provided, has a data type of URI with a syntax defined by:
 “<http://www.w3.org/1999/02/22-rdf-syntax#about>”
 Reference: [Section 7.4 on page 19](#)
- sbol-10203** ✓ The **displayId** is an OPTIONAL property for all **Identified** objects and, if provided, has a data type of String that is composed only of alphanumeric or underscore characters and MUST NOT begin with a digit.
 Reference: [Section 7.4 on page 19](#)
- sbol-10204** ✓ The **version** is an OPTIONAL property for all **Identified** objects and, if provided, has a data type of String that is composed only of alphanumeric characters, underscores, hyphens, and periods and MUST begin with a digit.
 Reference: [Section 7.4 on page 19](#)
- sbol-10205** ✓ The **annotations** field is an OPTIONAL list of for all **Identified** objects and, if provided, includes references to **Annotation** objects.
 Reference: [Section 7.4 on page 19](#)
- sbol-10206** ✓ The **wasDerivedFrom** property is OPTIONAL for all **Identified** objects and, if provided, has a data type of URI.
 Reference: [Section 7.4 on page 19](#)
- sbol-10207** ✓ The **name** is an OPTIONAL property for all **Identified** objects and, if provided, has a data type of String.
 Reference: [Section 7.4 on page 19](#)
- sbol-10208** ✓ The **description** is an OPTIONAL property for all **Identified** objects and, if provided, has a data type of String.
 Reference: [Section 7.4 on page 19](#)
- sbol-10209** ★ The **displayId** of a compliant object is REQUIRED.
 Reference: [Section 10.2 on page 57](#)
- sbol-10210** ★ The **persistentIdentity** of a compliant top level object is REQUIRED and MUST end with a delimiter (‘/’, ‘#’, or ‘:’) followed by the **displayId** of the object.
 Reference: [Section 10.2 on page 57](#)
- sbol-10211** ★ The **persistentIdentity** of a compliant child object is REQUIRED and MUST begin with the **persistentIdentity** of its parent object and be immediately followed by a delimiter (‘/’, ‘#’, or ‘:’) and the **displayId** of the object.
 Reference: [Section 10.2 on page 57](#)
- sbol-10212** ★ The **identity** of a compliant object MUST either be equal to the **persistentIdentity** when no **version** is specified or equal to “<persistentIdentity>/<version>” when a **version** is provided.
 Reference: [Section 10.2 on page 57](#)
- sbol-10213** ★ The **version** of a compliant child object is REQUIRED to be equal to the **version** of its parent object.
 Reference: [Section 10.2 on page 57](#)

Rules for the *TopLevel* class

- sbol-10301** ✓ A [TopLevel](#) object inherits all properties of a [Identified](#) object.
Reference: [Section 7.5 on page 21](#)

Rules for the *Sequence* class

- sbol-10401** ✓ A [Sequence](#) MUST inherit all properties of the [TopLevel](#) class.
Reference: [Section 7.6 on page 22](#)
- sbol-10402** ✓ The [elements](#) property of a [Sequence](#) is REQUIRED and MUST contain a [String](#).
Reference: [Section 7.6 on page 22](#)
- sbol-10403** ✓ The [encoding](#) property of [Sequence](#) is REQUIRED and MUST contain a [URI](#).
Reference: [Section 7.6 on page 22](#)
- sbol-10404** ▲ The [encoding](#) property of a [Sequence](#) MUST contain a [URI](#) from [Table 1](#) if it is well-described by this [URI](#).
Reference: [Section 7.6 on page 22](#)
- sbol-10405** ▲ The [elements](#) property of a [Sequence](#) MUST be consistent with its [encoding](#) property.
Reference: [Section 7.6 on page 22](#)

Rules for the *ComponentDefinition* class

- sbol-10501** ✓ A [ComponentDefinition](#) MUST inherit all properties of the [TopLevel](#) class.
Reference: [Section 7.7 on page 23](#)
- sbol-10502** ✓ The [types](#) property of a [ComponentDefinition](#) is REQUIRED and MUST contain a non-empty set of [URIs](#).
Reference: [Section 7.7 on page 23](#)
- sbol-10503** ▲ Each [URI](#) contained by the [types](#) property of a [ComponentDefinition](#) MUST refer to an ontology term that describes the category of biochemical or physical entity that is represented by the [ComponentDefinition](#).
Reference: [Section 7.7 on page 23](#)
- sbol-10504** ▲ All [URIs](#) contained by the [types](#) property of a [ComponentDefinition](#) MUST refer to non-conflicting ontology terms.
Reference: [Section 7.7 on page 23](#)
- sbol-10505** ✓ The [types](#) property of a [ComponentDefinition](#) MUST NOT contain more than one [URI](#) from [Table 2](#).
Reference: [Section 7.7 on page 23](#)
- sbol-10506** ▲ The [types](#) property of a [ComponentDefinition](#) MUST contain a [URI](#) from [Table 2](#) if it is well-described by this [URI](#).
Reference: [Section 7.7 on page 23](#)
- sbol-10507** ✓ The [types](#) property of a [ComponentDefinition](#) MUST NOT contain more than one [URI](#) from [Table 2](#).
Reference: [Section 7.7 on page 23](#)
- sbol-10508** ✓ The [roles](#) property of a [ComponentDefinition](#) is OPTIONAL and MAY contain a set of [URIs](#).
Reference: [Section 7.7 on page 23](#)
- sbol-10509** ▲ Each [URI](#) contained by the [roles](#) property of a [ComponentDefinition](#) MUST refer to an ontology term that clarifies the potential function of the [ComponentDefinition](#) in a biochemical

	or physical context.	1
	Reference: Section 7.7 on page 23	2
sbol-10510 ▲	Each URI contained by the roles property of a ComponentDefinition MUST refer to an ontology term that is consistent with its types property.	3
	Reference: Section 7.7 on page 23	4
sbol-10511 ★	The roles property of a ComponentDefinition SHOULD only contain a URI provided in Table 3 if one of its types is cross-listed with the URI .	6
	Reference: Section 7.7 on page 23	7
sbol-10512 ▲	The roles property of a ComponentDefinition MUST contain a URI from Table 3 if it is well-described by this URI .	9
	Reference: Section 7.7 on page 23	10
sbol-10513 ✓	The sequences property of a ComponentDefinition is OPTIONAL and MAY contain a set of URI references to Sequence objects.	12
	Reference: Section 7.7 on page 23	13
sbol-10514 ▲	The Sequence objects referred to by the sequences property of a ComponentDefinition MUST be consistent with each other, such that well-defined mappings exist between their elements properties in accordance with their encoding properties.	15
	Reference: Section 7.7 on page 23	16
sbol-10515 ★	If a ComponentDefinition refers to more than one Sequence with the same encoding , then the elements of these Sequence objects SHOULD have equal lengths.	17
	Reference: Section 7.7 on page 23	18
sbol-10516 ▲	The sequences property of a ComponentDefinition MUST NOT refer to Sequence objects with conflicting encoding properties.	19
	Reference: Section 7.7 on page 23	20
sbol-10517 ✓	The sequences property of a ComponentDefinition MUST NOT refer to Sequence objects with conflicting IUPAC encoding URIs from Table 1 .	21
	Reference: Section 7.7 on page 23	22
sbol-10518 ✓	If the sequences property of a ComponentDefinition refers to one or more Sequence objects, and one of the types of this ComponentDefinition comes from Table 2 , then one of the Sequence objects MUST have the encoding that is cross-listed with this type in Table 1 .	23
	Reference: Section 7.7 on page 23	24
sbol-10519 ✓	If the sequences property of a ComponentDefinition refers to a Sequence with an encoding from Table 1 , then the types property of the ComponentDefinition MUST contain the type from Table 2 that is cross-listed with this encoding in Table 1 .	25
	Reference: Section 7.7 on page 23	26
sbol-10520 ✓	The components property of a ComponentDefinition is OPTIONAL and MAY contain a set of Component objects.	27
	Reference: Section 7.7 on page 23	28
sbol-10521 ★	If a ComponentDefinition in a ComponentDefinition-Component hierarchy refers to one or more Sequence objects, and there exist ComponentDefinition objects lower in the hierarchy that refer to Sequence objects with the same encoding , then the elements properties of these Sequence objects SHOULD be consistent with each other, such that well-defined mappings exist from the “lower level” elements to the “higher level” elements in accordance with their	29
		30
		31
		32
		33
		34
		35
		36
		37
		38
		39
		40
		41
		42
		43

shared [encoding](#) (subject to any restrictions on the positions of [Component](#) objects in the hierarchy that are imposed by [SequenceAnnotation](#) or [SequenceConstraint](#) objects).

Reference: [Section 7.7 on page 23](#)

sbol-10522 ✓ The [sequenceAnnotations](#) property of a [ComponentDefinition](#) is OPTIONAL and MAY contain a set of [SequenceAnnotation](#) objects.
Reference: [Section 7.7 on page 23](#)

sbol-10523 ✓ If the [sequenceAnnotations](#) property of a [ComponentDefinition](#) contains two or more [SequenceAnnotation](#) objects that refer to the same [Component](#), then their [Location](#) objects MUST NOT specify regions that have conflicting [orientation](#) properties or occupy non-overlapping positions.
Reference: [Section 7.7 on page 23](#)

sbol-10524 ★ If the [sequences](#) property of a [ComponentDefinition](#) refers to a [Sequence](#) with an IUPAC [encoding](#) from [Table 1](#), then each [SequenceAnnotation](#) that includes a [Range](#) and/or [Cut](#) in the [sequenceAnnotations](#) property of the [ComponentDefinition](#) SHOULD specify a region on the [elements](#) of this [Sequence](#).
Reference: [Section 7.7 on page 23](#)

sbol-10525 ✓ The [sequenceConstraints](#) property of a [ComponentDefinition](#) is OPTIONAL and MAY contain a set of [SequenceConstraint](#) objects.
Reference: [Section 7.7 on page 23](#)

Rules for the *ComponentInstance* class

sbol-10601 ✓ A [ComponentInstance](#) MUST inherit all properties of the [Identified](#) class.
Reference: [Section 7.7.1 on page 27](#)

sbol-10602 ✓ The [access](#) property of a [ComponentInstance](#) is REQUIRED and MUST contain a [URI](#) from [Table 4](#).
Reference: [Section 7.7.1 on page 27](#)

sbol-10603 ★ It is RECOMMENDED that the [access](#) property of a [ComponentInstance](#) contain the [URI](#) <http://sbols.org/v2#public>.
Reference: [Section 7.7.1 on page 27](#)

sbol-10604 ✓ The [definition](#) property of a [ComponentInstance](#) is REQUIRED and MUST contain a [URI](#) reference to a [ComponentDefinition](#).
Reference: [Section 7.7.1 on page 27](#)

sbol-10605 ✓ The [definition](#) property of a [ComponentInstance](#) MUST NOT contain a [URI](#) reference to the [ComponentDefinition](#) that contains the [ComponentInstance](#).
Reference: [Section 7.7.1 on page 27](#)

sbol-10606 ▲ [ComponentInstance](#) objects MUST NOT form circular reference chains via their [definition](#) properties and parent [ComponentDefinition](#) objects.
Reference: [Section 7.7.1 on page 27](#)

sbol-10607 ✓ The [mapsTo](#)s property of a [ComponentInstance](#) is OPTIONAL and MAY contain a set of [MapsTo](#) objects.
Reference: [Section 7.7.1 on page 27](#)

Rules for the *Component* class

sbol-10701 ✓ A [Component](#) MUST inherit all properties of the [ComponentInstance](#) class.
Reference: [Section 7.7.1 on page 27](#)

Rules for the *MapsTo* class

- sbol-10801** ✓ A **MapsTo** MUST inherit all properties of the **Identified** class.
Reference: [Section 7.7.3 on page 29](#)
- sbol-10802** ✓ The **local** property of a **MapsTo** is REQUIRED and MUST contain a URI reference to a **ComponentInstance**.
Reference: [Section 7.7.3 on page 29](#)
- sbol-10803** ✓ The **local** property of a **MapsTo** MUST refer to a **ComponentInstance** with an **access** property that contains the URI <http://sbols.org/v2#public>.
Reference: [Section 7.7.3 on page 29](#)
- sbol-10804** ✓ If a **MapsTo** is contained by a **Component** in a **ComponentDefinition**, then the **local** property of the **MapsTo** MUST refer to another **Component** in the **ComponentDefinition**.
Reference: [Section 7.7.3 on page 29](#)
- sbol-10805** ✓ If a **MapsTo** is contained by a **FunctionalComponent** or **Module** in a **ModuleDefinition**, then the **local** property of the **MapsTo** MUST refer to another **FunctionalComponent** in the **ModuleDefinition**.
Reference: [Section 7.7.3 on page 29](#)
- sbol-10806** ✓ The **remote** property of a **MapsTo** is REQUIRED and MUST contain a URI reference to a **ComponentInstance**.
Reference: [Section 7.7.3 on page 29](#)
- sbol-10807** ▲ The **remote** property of a **MapsTo** MUST refer to a **ComponentInstance** with an **access** property that contains the URI <http://sbols.org/v2#public>.
Reference: [Section 7.7.3 on page 29](#)
- sbol-10808** ▲ If a **MapsTo** is contained by a **ComponentInstance**, then the **remote** property of the **MapsTo** MUST refer to a **Component** in the **ComponentDefinition** that is referenced by the **definition** of the **ComponentInstance**.
Reference: [Section 7.7.3 on page 29](#)
- sbol-10809** ▲ If a **MapsTo** is contained by a **Module**, then the **remote** property of the **MapsTo** MUST refer to a **FunctionalComponent** in the **ModuleDefinition** that is referenced by the **definition** of the **Module**.
Reference: [Section 7.7.3 on page 29](#)
- sbol-10810** ✓ The **refinement** property is REQUIRED and MUST contain a URI from [Table 5](#).
Reference: [Section 7.7.3 on page 29](#)

Rules for the *SequenceAnnotation* class

- sbol-10901** ✓ A **SequenceAnnotation** MUST inherit all properties of the **Identified** class.
Reference: [Section 7.7.4 on page 31](#)
- sbol-10902** ✓ The **locations** property of a **SequenceAnnotation** is REQUIRED and MUST contain a non-empty set of **Location** objects.
Reference: [Section 7.7.4 on page 31](#)
- sbol-10903** ▲ Each **Location** object in the list of **locations** should reference a valid location on the corresponding **Sequence** within the **ComponentDefinition** that contains the **SequenceAnnotation**.
- sbol-10904** ✓ The **component** property is OPTIONAL and MAY contain a URI reference to a **Component**.
Reference: [Section 7.7.4 on page 31](#)

- sbol-10905** ✓ The [Component](#) referenced by the [component](#) property of a [SequenceAnnotation](#) MUST be contained by the [ComponentDefinition](#) that contains the [SequenceAnnotation](#).
Reference: [Section 7.7.4 on page 31](#)

Rules for the *Location* class

- sbol-11001** ✓ A [Location](#) MUST inherit all properties of the [Identified](#) class.
Reference: [Section 7.7.5 on page 32](#)
- sbol-11002** ✓ The [orientation](#) property of a [Location](#) is OPTIONAL and MAY contain a [URI](#) from [Table 6](#).
Reference: [Section 7.7.5 on page 34](#)

Rules for the *Range* class

- sbol-11101** ✓ A [Range](#) MUST inherit all properties of the [Location](#) class.
Reference: [Section 7.7.5 on page 33](#)
- sbol-11102** ✓ The [start](#) property of a [Range](#) is REQUIRED and MUST contain an [Integer](#) greater than zero.
Reference: [Section 7.7.5 on page 33](#)
- sbol-11103** ✓ The [end](#) property of a [Range](#) is REQUIRED and MUST contain an [Integer](#) greater than zero.
Reference: [Section 7.7.5 on page 33](#)
- sbol-11104** ✓ The value of the [end](#) property of a [Range](#) MUST be greater than or equal to the value of its [start](#) property.
Reference: [Section 7.7.5 on page 33](#)

Rules for the *Cut* class

- sbol-11201** ✓ A [Cut](#) MUST inherit all properties of the [Location](#) class.
Reference: [Section 7.7.5 on page 34](#)
- sbol-11202** ✓ The [at](#) property is REQUIRED and MUST contain an [Integer](#) greater than or equal to zero.
Reference: [Section 7.7.5 on page 34](#)

Rules for the *GenericLocation* class

- sbol-11301** ✓ A [GenericLocation](#) MUST inherit all properties of the [Location](#) class.
Reference: [Section 7.7.5 on page 34](#)

Rules for the *SequenceConstraint* class

- sbol-11401** ✓ A [SequenceConstraint](#) MUST inherit all properties of the [Identified](#) class.
Reference: [Section 7.7.6 on page 35](#)
- sbol-11402** ✓ The [subject](#) property is REQUIRED and MUST contain a [URI](#) reference to a [Component](#).
Reference: [Section 7.7.6 on page 35](#)
- sbol-11403** ✓ The [Component](#) referenced by the [subject](#) property of a [SequenceConstraint](#) MUST be contained by the [ComponentDefinition](#) that contains the [SequenceConstraint](#).
Reference: [Section 7.7.6 on page 35](#)
- sbol-11404** ✓ The [object](#) property is REQUIRED and MUST contain a [URI](#) reference to a [Component](#).
Reference: [Section 7.7.6 on page 35](#)
- sbol-11405** ✓ The [Component](#) referenced by the [object](#) property of a [SequenceConstraint](#) MUST be contained by the [ComponentDefinition](#) that contains the [SequenceConstraint](#).
Reference: [Section 7.7.6 on page 35](#)

- sbol-11406** ✓ The **object** property of a **SequenceConstraint** MUST NOT refer to the same **Component** as the **subject** property of the **SequenceConstraint**.
Reference: [Section 7.7.6 on page 35](#)
- sbol-11407** ✓ The **restriction** property is REQUIRED and MUST contain a **URI**.
Reference: [Section 7.7.6 on page 35](#)
- sbol-11408** ★ The **URI** contained by the **restriction** property SHOULD come from [Table 7](#).
Reference: [Section 7.7.6 on page 35](#)

Rules for the *Model* class

- sbol-11501** ✓ A **Model** object inherits all properties of a **TopLevel** object.
Reference: [Section 7.8 on page 37](#)
- sbol-11502** ✓ The **source** property is a REQUIRED **URI** that specifies the location of the model source file.
Reference: [Section 7.8 on page 37](#)
- sbol-11503** ✓ The **language** property is a REQUIRED **URI** that specifies the language in which the model is encoded.
Reference: [Section 7.8 on page 37](#)
- sbol-11504** ★ The **language** property SHOULD be a **URI** from the EMBRACE Data and Methods (EDAM) ontology.
Reference: [Section 7.8 on page 37](#)
- sbol-11505** ✓ The **framework** property is a REQUIRED **URI** that specifies the modeling framework.
Reference: [Section 7.8 on page 37](#)
- sbol-11506** ★ The **framework** property SHOULD be a **URI** from the modeling framework branch of the SBO.
Reference: [Section 7.8 on page 37](#)
- sbol-11507** ▲ The **source** property MUST specify the location of the model source file in the specified **language** using the specified **framework**.
Reference: [Section 7.8 on page 37](#)

Rules for the *ModuleDefinition* class

- sbol-11601** ✓ A **ModuleDefinition** object inherits all properties of a **TopLevel** object.
Reference: [Section 7.9 on page 38](#)
- sbol-11602** ✓ The **roles** property is an OPTIONAL set of **URIs**.
Reference: [Section 7.9 on page 38](#)
- sbol-11603** ✓ The **modules** property is an OPTIONAL set of **Module** objects.
Reference: [Section 7.9 on page 38](#)
- sbol-11604** ✓ The **interactions** property is an OPTIONAL set of **Interaction** objects.
Reference: [Section 7.9 on page 38](#)
- sbol-11605** ✓ The **functionalComponents** property is an OPTIONAL set of **FunctionalComponent** objects.
Reference: [Section 7.9 on page 38](#)
- sbol-11606** ✓ The **models** property is an OPTIONAL set of **URIs** that reference **Model** objects.
Reference: [Section 7.9 on page 38](#)
- sbol-11607** ★ Each **URI** in the set of **models** SHOULD reference a **Model** object.
Reference: [Section 7.9 on page 38](#)

Rules for the *FunctionalComponent* class

- sbol-11701** ✓ A **FunctionalComponent** MUST inherit all properties of the **ComponentInstance** class.
Reference: [Section 7.7.1 on page 27](#)
- sbol-11702** ✓ The **direction** property of a **FunctionalComponent** is REQUIRED and MUST contain a **URI** from [Table 10](#).
Reference: [Section 7.9.1 on page 40](#)

Rules for the *Module* class

- sbol-11801** ✓ A **Module** object inherits all properties of a **Identified** object.
Reference: [Section 7.9.2 on page 41](#)
- sbol-11802** ✓ The **definition** property is a REQUIRED **URI** reference to a **ModuleDefinition** object.
Reference: [Section 7.9.2 on page 41](#)
- sbol-11803** ✓ The **mapsTo** property is an OPTIONAL set of **MapsTo** objects.
Reference: [Section 7.9.2 on page 41](#)

Rules for the *Interaction* class

- sbol-11901** ✓ An **Interaction** object inherits all properties of an **Identified** object.
Reference: [Section 7.9.3 on page 42](#)
- sbol-11902** ✓ The **types** property is a set of **URIs**, and it is REQUIRED to include at least one entry.
Reference: [Section 7.9.3 on page 42](#)
- sbol-11903** ★ A least one type in the set of **types** SHOULD be a **URI** from the occurring entity relationship branch of the SBO.
Reference: [Section 7.9.3 on page 42](#)
- sbol-11904** ✓ The **participations** property is an OPTIONAL set of **Participation** objects.
Reference: [Section 7.9.3 on page 42](#)

Rules for the *Participation* class

- sbol-12001** ✓ A **Participation** object inherits all properties of an **Identified** object.
Reference: [Section 7.9.4 on page 44](#)
- sbol-12002** ✓ The **participant** property is a REQUIRED **URI** that MUST reference a **FunctionalComponent** that is specified within the same **ModuleDefinition**.
Reference: [Section 7.9.4 on page 44](#)
- sbol-12003** ✓ The **roles** property is an OPTIONAL set of **URIs**.
Reference: [Section 7.9.4 on page 44](#)
- sbol-12004** ★ A least one role in the set of **roles** SHOULD be a **URI** from the participant role branch of the SBO.
Reference: [Section 7.9.4 on page 44](#)

Rules for the *Collection* class

- sbol-12101** ✓ A **Collection** object inherits all properties of a **TopLevel** object.
Reference: [Section 7.10 on page 44](#)
- sbol-12102** ✓ The **members** property is an OPTIONAL set of **URIs**. that reference **TopLevel** objects.
Reference: [Section 7.10 on page 44](#)

sbol-12103 ★ Each [URI](#) in the set of [members](#) SHOULD reference a [TopLevel](#) object.
Reference: [Section 7.10 on page 44](#)

Rules for the Annotation class

sbol-12201 ✓ The [name](#) property is REQUIRED, and it has data type [QName](#).
Reference: [Section 7.11 on page 45](#)

sbol-12202 ✓ The [value](#) property is REQUIRED, and it has data type [AnnotationValue](#).
Reference: [Section 7.11 on page 45](#)

sbol-12203 ✓ The [AnnotationValue](#) class MUST be of data type [String](#), [Integer](#), [Double](#), [Boolean](#), [URI](#), or [NestedAnnotations](#).
Reference: [Section 7.11 on page 45](#)

sbol-12204 ✓ The [nestedURI](#) property is REQUIRED for a [NestedAnnotations](#) object, and it has data type [URI](#).
Reference: [Section 7.11 on page 45](#)

sbol-12205 ✓ The [annotations](#) property is an OPTIONAL set for a [NestedAnnotations](#) object, and each member is of data type [Annotation](#).
Reference: [Section 7.11 on page 45](#)

Rules for the GenericTopLevel class

sbol-12301 ✓ A [GenericTopLevel](#) object inherits all properties of a [TopLevel](#) object.
Reference: [Section 7.11.2 on page 47](#)

sbol-12302 ✓ The [rdfType](#) property is REQUIRED, and it has data type [QName](#).
Reference: [Section 7.11.2 on page 47](#)

B Examples of Serialization

Mike B: "DNA sequences run off the page. Historically, this has often been a problem with long DNA string datatypes that don't permit inline whitespace. They're impossible to print or justify for web presentation without breaking the validity of the document." Goksel: I like the Genbank files having 60 nucleotides in one row. :) I keep Mike's comment here in case if someone else has any suggestion.

B.1 PoPS Receiver

This example shows the serialization of the PoPS Receiver device designed by Canton and co-workers [Canton et al. \(2008\)](#). In particular, this is a [ComponentDefinition](#) comprising five other [Component](#) objects to construct a detector for the cell-cell signaling molecule 3OC₆HSL. The five components are arranged in a sequence: first come four components together implement constitutive expression of the LuxR protein, which responds to 3OC₆HSL: a constitutive promoter, 5'UTR, coding sequence for LuxR, and terminator. Finally, after this comes the pLuxR promoter, which is activated in the presence of LuxR and 3OC₆HSL. Complete details of the device can be found in the cited paper and also at http://parts.igem.org/Part:BBa_F2620.

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:pr="http://partsregistry.org" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.
  org/dc/terms/" xmlns:prov="http://www.w3.org/ns/prov#" xmlns:sbol="http://sbols.org/v2#">
  <sbol:ComponentDefinition rdf:about="http://partsregistry.org/cd/BBa_F2620">
    <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620"/>
    <sbol:displayId>BBa_F2620</sbol:displayId>
    <dcterms:title>BBa_F2620</dcterms:title>
    <dcterms:description>3OC6HSL -&gt; PoPS Receiver</dcterms:description>
    <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
    <sbol:role rdf:resource="http://identifiers.org/so/SO:00001411"/>
    <sbol:component>
      <sbol:Component rdf:about="http://partsregistry.org/cd/BBa_F2620/pLuxR">
        <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/pLuxR"/>
        <sbol:displayId>pLuxR</sbol:displayId>
        <sbol:access rdf:resource="http://sbols.org/v2#public"/>
        <sbol:definition rdf:resource="http://partsregistry.org/cd/BBa_R0062"/>
      </sbol:Component>
    </sbol:component>
    <sbol:component>
      <sbol:Component rdf:about="http://partsregistry.org/cd/BBa_F2620/luxR">
        <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/luxR"/>
        <sbol:displayId>luxR</sbol:displayId>
        <sbol:access rdf:resource="http://sbols.org/v2#public"/>
        <sbol:definition rdf:resource="http://partsregistry.org/cd/BBa_C0062"/>
      </sbol:Component>
    </sbol:component>
    <sbol:component>
      <sbol:Component rdf:about="http://partsregistry.org/cd/BBa_F2620/pTetR">
        <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/pTetR"/>
        <sbol:displayId>pTetR</sbol:displayId>
        <sbol:access rdf:resource="http://sbols.org/v2#public"/>
        <sbol:definition rdf:resource="http://partsregistry.org/cd/BBa_R0040"/>
      </sbol:Component>
    </sbol:component>
    <sbol:component>
      <sbol:Component rdf:about="http://partsregistry.org/cd/BBa_F2620/ter">
        <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/ter"/>
        <sbol:displayId>ter</sbol:displayId>
        <sbol:access rdf:resource="http://sbols.org/v2#public"/>
        <sbol:definition rdf:resource="http://partsregistry.org/cd/BBa_B0015"/>
      </sbol:Component>
    </sbol:component>
    <sbol:component>
      <sbol:Component rdf:about="http://partsregistry.org/cd/BBa_F2620/rbs">
```



```

<sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/rbs"/>
<sbol:displayId>rbs</sbol:displayId>
<sbol:access rdf:resource="http://sbols.org/v2#public"/>
<sbol:definition rdf:resource="http://partsregistry.org/cd/BBa_B0034"/>
</sbol:Component>
</sbol:component>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://partsregistry.org/cd/BBa_F2620/anno3">
    <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno3"/>
    <sbol:displayId>anno3</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://partsregistry.org/cd/BBa_F2620/anno3/location3">
        <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno3/location3"/>
        <sbol:displayId>location3</sbol:displayId>
        <sbol:start>69</sbol:start>
        <sbol:end>770</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://partsregistry.org/cd/BBa_F2620/luxR"/>
  </sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://partsregistry.org/cd/BBa_F2620/anno5">
    <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno5"/>
    <sbol:displayId>anno5</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://partsregistry.org/cd/BBa_F2620/anno5/location5">
        <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno5/location5"/>
        <sbol:displayId>location5</sbol:displayId>
        <sbol:start>901</sbol:start>
        <sbol:end>956</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://partsregistry.org/cd/BBa_F2620/pLuxR"/>
  </sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://partsregistry.org/cd/BBa_F2620/anno1">
    <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno1"/>
    <sbol:displayId>anno1</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://partsregistry.org/cd/BBa_F2620/anno1/location1">
        <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno1/location1"/>
        <sbol:displayId>location1</sbol:displayId>
        <sbol:start>1</sbol:start>
        <sbol:end>55</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://partsregistry.org/cd/BBa_F2620/pTetR"/>
  </sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://partsregistry.org/cd/BBa_F2620/anno4">
    <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno4"/>
    <sbol:displayId>anno4</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://partsregistry.org/cd/BBa_F2620/anno4/location4">
        <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno4/location4"/>
        <sbol:displayId>location4</sbol:displayId>
        <sbol:start>771</sbol:start>
        <sbol:end>900</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://partsregistry.org/cd/BBa_F2620/ter"/>
  </sbol:SequenceAnnotation>

```



```

</sbol:sequenceAnnotation>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://partsregistry.org/cd/BBa_F2620/anno2">
    <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno2"/>
    <sbol:displayId>anno2</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://partsregistry.org/cd/BBa_F2620/anno2/location2">
        <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_F2620/anno2/location2"/>
        <sbol:displayId>location2</sbol:displayId>
        <sbol:start>56</sbol:start>
        <sbol:end>68</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://partsregistry.org/cd/BBa_F2620/rbs"/>
  </sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://partsregistry.org/cd/BBa_R0062">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_R0062"/>
  <sbol:displayId>BBa_R0062</sbol:displayId>
  <dcterms:title>pLuxR</dcterms:title>
  <dcterms:description>LuxR inducible promoter</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000167"/>
  <sbol:sequence rdf:resource="http://partsregistry.org/seq/BBa_R0062"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://partsregistry.org/cd/BBa_B0015">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_B0015"/>
  <sbol:displayId>BBa_B0015</sbol:displayId>
  <dcterms:title>BBa_B0015</dcterms:title>
  <dcterms:description>Double terminator</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000141"/>
  <sbol:sequence rdf:resource="http://partsregistry.org/seq/BBa_B0015"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://partsregistry.org/cd/BBa_C0062">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_C0062"/>
  <sbol:displayId>BBa_C0062</sbol:displayId>
  <dcterms:title>luxR</dcterms:title>
  <dcterms:description>luxR coding sequence</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000316"/>
  <sbol:sequence rdf:resource="http://partsregistry.org/seq/BBa_C0062"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://partsregistry.org/cd/BBa_R0040">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_R0040"/>
  <sbol:displayId>BBa_R0040</sbol:displayId>
  <dcterms:title>pTetR</dcterms:title>
  <dcterms:description>TetR repressible promoter</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000167"/>
  <sbol:sequence rdf:resource="http://partsregistry.org/seq/BBa_R0040"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://partsregistry.org/cd/BBa_B0034">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/cd/BBa_B0034"/>
  <sbol:displayId>BBa_B0034</sbol:displayId>
  <dcterms:title>BBa_B0034</dcterms:title>
  <dcterms:description>RBS based on Elowitz repressilator</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000139"/>
  <sbol:sequence rdf:resource="http://partsregistry.org/seq/BBa_B0034"/>
</sbol:ComponentDefinition>
<sbol:Sequence rdf:about="http://partsregistry.org/seq/BBa_B0034">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/seq/BBa_B0034"/>
  <sbol:displayId>BBa_B0034</sbol:displayId>
  <sbol:elements>aaagaggagaaa</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>

```

```

<sbol:Sequence rdf:about="http://partsregistry.org/seq/BBa_R0040">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/seq/BBa_R0040"/>
  <sbol:displayId>BBa_R0040</sbol:displayId>
  <sbol:elements>tccttatcagtgatagagattgcacccctatcagtgatagagatactgagcac</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://partsregistry.org/seq/BBa_C0062">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/seq/BBa_C0062"/>
  <sbol:displayId>BBa_C0062</sbol:displayId>
  <sbol:elements>atgccttatctgatagactaaaatgggtacattgtgaatattatttactcgcatcatttatcctcattcta
  tgggttaaatctgatatttcaatcctagataattaccctaaaaatggaggcaatattatgatgacgctaatttaaaaaatgat
  cctatagtagattattctaactccaatcattcaccaatttaattggaatatatttgaaacaatgctgtaataaaaaatctccaaa
  tgtaattaaagaagcgaaacatcaggtcttatcactgggttagtttcctattcaccgctaacaatggcttcggaatgctta
  gttttgcacattcagaaaaagacaactatatagatagtttattttacatgctgtatgaacataccattaattgttccttctcta
  gttgataattatcgaaaaataatatagcaataataatcaacaacgatttaaccaaagagaaaaagaatgtttagcgtgggc
  atgcgaagggaaaaagctcttgggatatttcaaaaatattaggttgacgtgagcgtactgtcactttccatttaaccaatgcgcaa
  tgaactcaatcacacaacgcgtgccaagtatttctaaagcaattttaacaggagcaattgattgccatactttaaaaattaa
  taacactgatagtgtagtgtgatcac</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://partsregistry.org/seq/BBa_R0062">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/seq/BBa_R0062"/>
  <sbol:displayId>BBa_R0062</sbol:displayId>
  <sbol:elements>acctgtaggatcgtagcaggtttacgcaagaaaatgggtttgttatagtcgaataaa</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://partsregistry.org/seq/BBa_B0015">
  <sbol:persistentIdentity rdf:resource="http://partsregistry.org/seq/BBa_B0015"/>
  <sbol:displayId>BBa_B0015</sbol:displayId>
  <sbol:elements>ccaggcatcaataaaacgaaaggctcagtcgaagactgggcctttcgttttatctgtgtttgttcgggtg
  aacgctctctactagagtcacactggctcaccttcgggtgggcctttctgcgtttata</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
</rdf:RDF>

```

B.2 Toggle Switch

This example shows the serialization of an SBOL data model for a LacI/TetR toggle switch similar to those constructed in [Gardner et al. \(2000\)](#). This design is essentially similar to the one presented in [Section 8](#), except that it uses some alternate groupings in how the total design is built up out of smaller entities.

```

<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:prov="http://www.w3.org/ns/prov#" xmlns:sbol="http://sbols.org/v2#">
  <sbol:ModuleDefinition rdf:about="http://sbolstandard.org/example/toggle_switch">
    <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/toggle_switch"/>
    <sbol:displayId>toggle_switch</sbol:displayId>
    <sbol:role rdf:resource="http://sbolstandard.org/example/module_role/toggle_switch"/>
    <sbol:functionalComponent>
      <sbol:FunctionalComponent rdf:about="http://sbolstandard.org/example/toggle_switch/LacI">
        <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/toggle_switch/LacI"/>
        <sbol:displayId>LacI</sbol:displayId>
        <sbol:definition rdf:resource="http://identifiers.org/uniprot/P03023"/>
        <sbol:access rdf:resource="http://sbols.org/v2#public"/>
        <sbol:direction rdf:resource="http://sbols.org/v2#inout"/>
      </sbol:FunctionalComponent>
    </sbol:functionalComponent>
    <sbol:functionalComponent>
      <sbol:FunctionalComponent rdf:about="http://sbolstandard.org/example/toggle_switch/TetR">
        <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/toggle_switch/TetR"/>
        <sbol:displayId>TetR</sbol:displayId>
        <sbol:definition rdf:resource="http://identifiers.org/uniprot/Q6QR72"/>
        <sbol:access rdf:resource="http://sbols.org/v2#public"/>
        <sbol:direction rdf:resource="http://sbols.org/v2#inout"/>
      </sbol:FunctionalComponent>
    </sbol:functionalComponent>
  </sbol:ModuleDefinition>
</rdf:RDF>

```

```

<sbol:model rdf:resource="http://sbolstandard.org/example/toogleswitch"/>
<sbol:module>
  <sbol:Module rdf:about="http://sbolstandard.org/example/toggle_switch/laci_inverter">
    <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/toggle_switch/laci_inverter"/>
    <sbol:displayId>laci_inverter</sbol:displayId>
    <sbol:definition rdf:resource="http://sbolstandard.org/example/laci_inverter"/>
    <sbol:mapsTo>
      <sbol:MapsTo rdf:about="http://sbolstandard.org/example/toggle_switch/laci_inverter/LacI_mapping">
        <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/toggle_switch/laci_inverter/LacI_mapping"/>
        <sbol:displayId>LacI_mapping</sbol:displayId>
        <sbol:refinement rdf:resource="http://sbols.org/v2#useRemote"/>
        <sbol:remote rdf:resource="http://sbolstandard.org/example/toggle_switch/LacI"/>
        <sbol:local rdf:resource="http://sbolstandard.org/example/laci_inverter/TF"/>
      </sbol:MapsTo>
    </sbol:mapsTo>
  </sbol:Module>
</sbol:module>
<sbol:module>
  <sbol:Module rdf:about="http://sbolstandard.org/example/toggle_switch/tetr_inverter">
    <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/toggle_switch/tetr_inverter"/>
    <sbol:displayId>tetr_inverter</sbol:displayId>
    <sbol:definition rdf:resource="http://sbolstandard.org/example/tetr_inverter"/>
    <sbol:mapsTo>
      <sbol:MapsTo rdf:about="http://sbolstandard.org/example/toggle_switch/tetr_inverter/TetR_mapping">
        <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/toggle_switch/tetr_inverter/TetR_mapping"/>
        <sbol:displayId>TetR_mapping</sbol:displayId>
        <sbol:refinement rdf:resource="http://sbols.org/v2#useRemote"/>
        <sbol:remote rdf:resource="http://sbolstandard.org/example/toggle_switch/TetR"/>
        <sbol:local rdf:resource="http://sbolstandard.org/example/tetr_inverter/TF"/>
      </sbol:MapsTo>
    </sbol:mapsTo>
  </sbol:Module>
</sbol:module>
</sbol:ModuleDefinition>
<sbol:ModuleDefinition rdf:about="http://sbolstandard.org/example/laci_inverter">
  <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/laci_inverter"/>
  <sbol:displayId>laci_inverter</sbol:displayId>
  <sbol:role rdf:resource="http://parts.igem.org/cgi/partsdb/pgroup.cgi?pgroup=inverter"/>
  <sbol:functionalComponent>
    <sbol:FunctionalComponent rdf:about="http://sbolstandard.org/example/laci_inverter/TF">
      <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/laci_inverter/TF"/>
      <sbol:displayId>TF</sbol:displayId>
      <sbol:definition rdf:resource="http://identifiers.org/uniprot/P03023"/>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:direction rdf:resource="http://sbols.org/v2#inout"/>
    </sbol:FunctionalComponent>
  </sbol:functionalComponent>
  <sbol:functionalComponent>
    <sbol:FunctionalComponent rdf:about="http://sbolstandard.org/example/laci_inverter/promoter">
      <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/laci_inverter/promoter"/>
      <sbol:displayId>promoter</sbol:displayId>
      <sbol:definition rdf:resource="http://www.partsregistry.org/BBa_R0010"/>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:direction rdf:resource="http://sbols.org/v2#inout"/>
    </sbol:FunctionalComponent>
  </sbol:functionalComponent>
  <sbol:interaction>
    <sbol:Interaction rdf:about="http://sbolstandard.org/example/laci_inverter/LacI_pLacI">
      <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/laci_inverter/LacI_pLacI"/>
      <sbol:displayId>LacI_pLacI</sbol:displayId>
      <sbol:type rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000169"/>
      <sbol:participation>
        <sbol:Participation rdf:about="http://sbolstandard.org/example/laci_inverter/LacI_pLacI/BBa_R0010">
          <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/laci_inverter/LacI_pLacI/BBa_R0010"/>
          <sbol:displayId>BBa_R0010</sbol:displayId>
          <sbol:role rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000598"/>
          <sbol:participant rdf:resource="http://sbolstandard.org/example/laci_inverter/promoter"/>
        </sbol:Participation>
      </sbol:participation>
    </sbol:interaction>
  </sbol:interaction>

```

```

<sbol:participation>
  <sbol:Participation rdf:about="http://sbolstandard.org/example/laci_inverter/LacI_pLacI/P03023">
    <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/laci_inverter/LacI_pLacI/P03023"/>
    <sbol:displayId>P03023</sbol:displayId>
    <sbol:role rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000020"/>
    <sbol:participant rdf:resource="http://sbolstandard.org/example/laci_inverter/TF"/>
  </sbol:Participation>
</sbol:participation>
</sbol:interaction>
</sbol:ModuleDefinition>
<sbol:ModuleDefinition rdf:about="http://sbolstandard.org/example/tetr_inverter">
  <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/tetr_inverter"/>
  <sbol:displayId>tetr_inverter</sbol:displayId>
  <sbol:role rdf:resource="http://parts.igem.org/cgi/partsdb/pgroup.cgi?pgroup=inverter"/>
  <sbol:functionalComponent>
    <sbol:FunctionalComponent rdf:about="http://sbolstandard.org/example/tetr_inverter/promoter">
      <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/tetr_inverter/promoter"/>
      <sbol:displayId>promoter</sbol:displayId>
      <sbol:definition rdf:resource="http://www.partsregistry.org/BBa_R0040"/>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:direction rdf:resource="http://sbols.org/v2#inout"/>
    </sbol:FunctionalComponent>
  </sbol:functionalComponent>
  <sbol:functionalComponent>
    <sbol:FunctionalComponent rdf:about="http://sbolstandard.org/example/tetr_inverter/TF">
      <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/tetr_inverter/TF"/>
      <sbol:displayId>TF</sbol:displayId>
      <sbol:definition rdf:resource="http://identifiers.org/uniprot/Q6QR72"/>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:direction rdf:resource="http://sbols.org/v2#inout"/>
    </sbol:FunctionalComponent>
  </sbol:functionalComponent>
  <sbol:interaction>
    <sbol:Interaction rdf:about="http://sbolstandard.org/example/tetr_inverter/LacI_pLacI">
      <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/tetr_inverter/LacI_pLacI"/>
      <sbol:displayId>LacI_pLacI</sbol:displayId>
      <sbol:type rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000169"/>
      <sbol:participation>
        <sbol:Participation rdf:about="http://sbolstandard.org/example/tetr_inverter/LacI_pLacI/Q6QR72">
          <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/tetr_inverter/LacI_pLacI/Q6QR72"/>
          <sbol:displayId>Q6QR72</sbol:displayId>
          <sbol:role rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000020"/>
          <sbol:participant rdf:resource="http://sbolstandard.org/example/tetr_inverter/TF"/>
        </sbol:Participation>
      </sbol:participation>
      <sbol:participation>
        <sbol:Participation rdf:about="http://sbolstandard.org/example/tetr_inverter/LacI_pLacI/BBa_R0040">
          <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/tetr_inverter/LacI_pLacI/BBa_R0040"/>
          <sbol:displayId>BBa_R0040</sbol:displayId>
          <sbol:role rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000598"/>
          <sbol:participant rdf:resource="http://sbolstandard.org/example/tetr_inverter/promoter"/>
        </sbol:Participation>
      </sbol:participation>
    </sbol:Interaction>
  </sbol:interaction>
</sbol:ModuleDefinition>
<sbol:Model rdf:about="http://sbolstandard.org/example/toogleswitch">
  <sbol:persistentIdentity rdf:resource="http://sbolstandard.org/example/toogleswitch"/>
  <sbol:displayId>toogleswitch</sbol:displayId>
  <sbol:source rdf:resource="http://virtualparts.org/part/pIKE_Toggle_1"/>
  <sbol:language rdf:resource="http://identifiers.org/edam/format_2585"/>
  <sbol:framework rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000062"/>
</sbol:Model>
<sbol:ComponentDefinition rdf:about="http://www.partsregistry.org/BBa_J61130">
  <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/BBa_J61130"/>
  <sbol:displayId>BBa_J61130</sbol:displayId>
  <dcterms:title>BBa_J61101 RBS</dcterms:title>
  <dcterms:description>RBS2</dcterms:description>

```

```

<sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
<sbol:role rdf:resource="http://identifiers.org/so/SO:0000139"/>
<sbol:sequence rdf:resource="http://www.virtualparts.org/part/BBa_J61130"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1"/>
  <sbol:displayId>pIKELeftCassette_1</sbol:displayId>
  <dc:terms:title>TetR Inverter</dc:terms:title>
  <dc:terms:description>TetR Inverter</dc:terms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000280"/>
  <sbol:component>
    <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/ECK120029600">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/ECK120029600"/>
      <sbol:displayId>ECK120029600</sbol:displayId>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:definition rdf:resource="http://www.partsregistry.org/ECK120029600"/>
    </sbol:Component>
  </sbol:component>
  <sbol:component>
    <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/BBa_R0040">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/BBa_R0040"/>
      <sbol:displayId>BBa_R0040</sbol:displayId>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:definition rdf:resource="http://www.partsregistry.org/BBa_R0040"/>
    </sbol:Component>
  </sbol:component>
  <sbol:component>
    <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/BBa_C0012">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/BBa_C0012"/>
      <sbol:displayId>BBa_C0012</sbol:displayId>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:definition rdf:resource="http://www.partsregistry.org/BBa_C0012"/>
    </sbol:Component>
  </sbol:component>
  <sbol:component>
    <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/BBa_J61101">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/BBa_J61101"/>
      <sbol:displayId>BBa_J61101</sbol:displayId>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:definition rdf:resource="http://www.partsregistry.org/BBa_J61101"/>
    </sbol:Component>
  </sbol:component>
  <sbol:sequenceAnnotation>
    <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/anno3">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/anno3"/>
      <sbol:displayId>anno3</sbol:displayId>
      <sbol:location>
        <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/anno3/range">
          <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/anno3/range"/>
          <sbol:displayId>range</sbol:displayId>
          <sbol:start>69</sbol:start>
          <sbol:end>1197</sbol:end>
          <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
        </sbol:Range>
      </sbol:location>
      <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/BBa_C0012"/>
    </sbol:SequenceAnnotation>
  </sbol:sequenceAnnotation>
  <sbol:sequenceAnnotation>
    <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/anno4">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/anno4"/>
      <sbol:displayId>anno4</sbol:displayId>
      <sbol:location>
        <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/anno4/range">
          <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/anno4/range"/>
          <sbol:displayId>range</sbol:displayId>
          <sbol:start>1198</sbol:start>
          <sbol:end>1288</sbol:end>

```

```

    <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
  </sbol:Range>
</sbol:location>
  <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/ECK120029600"/>
</sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/anno2">
    <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/anno2"/>
    <sbol:displayId>anno2</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/anno2/range">
        <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/anno2/range"/>
        <sbol:displayId>range</sbol:displayId>
        <sbol:start>56</sbol:start>
        <sbol:end>68</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/BBa_J61101"/>
  </sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/anno1">
    <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/anno1"/>
    <sbol:displayId>anno1</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKELeftCassette_1/anno1/range">
        <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/anno1/range"/>
        <sbol:displayId>range</sbol:displayId>
        <sbol:start>1</sbol:start>
        <sbol:end>55</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1/BBa_R0040"/>
  </sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.partsregistry.org/BBa_C0012">
  <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/BBa_C0012"/>
  <sbol:displayId>BBa_C0012</sbol:displayId>
  <dcterms:title>lacI</dcterms:title>
  <dcterms:description>lacI coding sequence</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000316"/>
  <sbol:sequence rdf:resource="http://www.virtualparts.org/part/BBa_C0012"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.partsregistry.org/ECK120033736">
  <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/ECK120033736"/>
  <sbol:displayId>ECK120033736</sbol:displayId>
  <dcterms:title>ECK120033736</dcterms:title>
  <dcterms:description>Terminator2</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000141"/>
  <sbol:sequence rdf:resource="http://www.virtualparts.org/part/ECK120033736"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.partsregistry.org/BBa_R0040">
  <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/BBa_R0040"/>
  <sbol:displayId>BBa_R0040</sbol:displayId>
  <dcterms:title>pTetR</dcterms:title>
  <dcterms:description>pTet promoter</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000167"/>
  <sbol:sequence rdf:resource="http://www.virtualparts.org/part/BBa_R0040"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://identifiers.org/uniprot/Q6QR72">
  <sbol:persistentIdentity rdf:resource="http://identifiers.org/uniprot/Q6QR72"/>
  <sbol:displayId>Q6QR72</sbol:displayId>

```



```

<dcterms:title>TetR</dcterms:title>
<dcterms:description>TetR protein</dcterms:description>
<sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#Protein"/>
<sbol:role rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000020"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://identifiers.org/uniprot/P03023">
  <sbol:persistentIdentity rdf:resource="http://identifiers.org/uniprot/P03023"/>
  <sbol:displayId>P03023</sbol:displayId>
  <dcterms:title>LacI</dcterms:title>
  <dcterms:description>LacI protein</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#Protein"/>
  <sbol:role rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000020"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.partsregistry.org/BBa_J61120">
  <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/BBa_J61120"/>
  <sbol:displayId>BBa_J61120</sbol:displayId>
  <dcterms:title>BBa_J61101 RBS</dcterms:title>
  <dcterms:description>RBS2</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000139"/>
  <sbol:sequence rdf:resource="http://www.virtualparts.org/part/BBa_J61120"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.partsregistry.org/BBa_E0040">
  <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/BBa_E0040"/>
  <sbol:displayId>BBa_E0040</sbol:displayId>
  <dcterms:title>gfp</dcterms:title>
  <dcterms:description>gfp coding sequence</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000316"/>
  <sbol:sequence rdf:resource="http://www.virtualparts.org/part/BBa_E0040"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.partsregistry.org/ECK120029600">
  <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/ECK120029600"/>
  <sbol:displayId>ECK120029600</sbol:displayId>
  <dcterms:title>ECK120029600</dcterms:title>
  <dcterms:description>Terminator1</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000141"/>
  <sbol:sequence rdf:resource="http://www.virtualparts.org/part/ECK120029600"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1"/>
  <sbol:displayId>pIKERightCassette_1</sbol:displayId>
  <dcterms:title>LacI Inverter</dcterms:title>
  <dcterms:description>LacI Inverter</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000280"/>
  <sbol:component>
    <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_R0010">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_R0010"/>
      <sbol:displayId>BBa_R0010</sbol:displayId>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:definition rdf:resource="http://www.partsregistry.org/BBa_R0010"/>
    </sbol:Component>
  </sbol:component>
  <sbol:component>
    <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_C0040">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_C0040"/>
      <sbol:displayId>BBa_C0040</sbol:displayId>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:definition rdf:resource="http://www.partsregistry.org/BBa_C0040"/>
    </sbol:Component>
  </sbol:component>
  <sbol:component>
    <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_J61130">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_J61130"/>
      <sbol:displayId>BBa_J61130</sbol:displayId>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:definition rdf:resource="http://www.partsregistry.org/BBa_J61130"/>
    </sbol:Component>
  </sbol:component>

```

```

</sbol:Component>
</sbol:component>
<sbol:component>
  <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_E0040">
    <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_E0040"/>
    <sbol:displayId>BBa_E0040</sbol:displayId>
    <sbol:access rdf:resource="http://sbols.org/v2#public"/>
    <sbol:definition rdf:resource="http://www.partsregistry.org/BBa_E0040"/>
  </sbol:Component>
</sbol:component>
<sbol:component>
  <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/ECK120033736">
    <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/ECK120033736"/>
    <sbol:displayId>ECK120033736</sbol:displayId>
    <sbol:access rdf:resource="http://sbols.org/v2#public"/>
    <sbol:definition rdf:resource="http://www.partsregistry.org/ECK120033736"/>
  </sbol:Component>
</sbol:component>
<sbol:component>
  <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_J61120">
    <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_J61120"/>
    <sbol:displayId>BBa_J61120</sbol:displayId>
    <sbol:access rdf:resource="http://sbols.org/v2#public"/>
    <sbol:definition rdf:resource="http://www.partsregistry.org/BBa_J61120"/>
  </sbol:Component>
</sbol:component>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno3">
    <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno3"/>
    <sbol:displayId>anno3</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno3/range">
        <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno3/range"/>
        <sbol:displayId>range</sbol:displayId>
        <sbol:start>69</sbol:start>
        <sbol:end>729</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_C0040"/>
  </sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno2">
    <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno2"/>
    <sbol:displayId>anno2</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno2/range">
        <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno2/range"/>
        <sbol:displayId>range</sbol:displayId>
        <sbol:start>56</sbol:start>
        <sbol:end>68</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_J61120"/>
  </sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno4">
    <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno4"/>
    <sbol:displayId>anno4</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno4/range">
        <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno4/range"/>
        <sbol:displayId>range</sbol:displayId>
        <sbol:start>730</sbol:start>
        <sbol:end>742</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
  </sbol:SequenceAnnotation>

```



```

    </sbol:Range>
  </sbol:location>
  <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_J61130"/>
</sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno1">
    <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno1"/>
    <sbol:displayId>anno1</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno1/range">
        <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno1/range"/>
        <sbol:displayId>range</sbol:displayId>
        <sbol:start>1</sbol:start>
        <sbol:end>55</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_R0010"/>
  </sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno5">
    <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno5"/>
    <sbol:displayId>anno5</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno5/range">
        <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno5/range"/>
        <sbol:displayId>range</sbol:displayId>
        <sbol:start>743</sbol:start>
        <sbol:end>1463</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/BBa_E0040"/>
  </sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
<sbol:sequenceAnnotation>
  <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno6">
    <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno6"/>
    <sbol:displayId>anno6</sbol:displayId>
    <sbol:location>
      <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKERightCassette_1/anno6/range">
        <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/anno6/range"/>
        <sbol:displayId>range</sbol:displayId>
        <sbol:start>1464</sbol:start>
        <sbol:end>1554</sbol:end>
        <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
      </sbol:Range>
    </sbol:location>
    <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1/ECK120033736"/>
  </sbol:SequenceAnnotation>
</sbol:sequenceAnnotation>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.partsregistry.org/BBa_J61101">
  <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/BBa_J61101"/>
  <sbol:displayId>BBa_J61101</sbol:displayId>
  <dcterms:title>BBa_J61101 RBS</dcterms:title>
  <dcterms:description>RBS1</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000139"/>
  <sbol:sequence rdf:resource="http://www.virtualparts.org/part/BBa_J61101"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.partsregistry.org/BBa_R0010">
  <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/BBa_R0010"/>
  <sbol:displayId>BBa_R0010</sbol:displayId>
  <dcterms:title>pLacI</dcterms:title>
  <dcterms:description>pLacI promoter</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>

```

```

<sbol:role rdf:resource="http://identifiers.org/so/SO:0000167"/>
<sbol:sequence rdf:resource="http://www.virtualparts.org/part/BBa_R0010"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.virtualparts.org/part/pIKE_Toggle_1">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKE_Toggle_1"/>
  <sbol:displayId>pIKE_Toggle_1</sbol:displayId>
  <dcterms:title>LacI/TetR Toggle Switc</dcterms:title>
  <dcterms:description>LacI/TetR Toggle Switc</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
  <sbol:role rdf:resource="http://identifiers.org/so/SO:0000280"/>
  <sbol:component>
    <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKE_Toggle_1/pIKERightCassette_1">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKE_Toggle_1/pIKERightCassette_1"/>
      <sbol:displayId>pIKERightCassette_1</sbol:displayId>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:definition rdf:resource="http://www.virtualparts.org/part/pIKERightCassette_1"/>
    </sbol:Component>
  </sbol:component>
  <sbol:component>
    <sbol:Component rdf:about="http://www.virtualparts.org/part/pIKE_Toggle_1/pIKELeftCassette_1">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKE_Toggle_1/pIKELeftCassette_1"/>
      <sbol:displayId>pIKELeftCassette_1</sbol:displayId>
      <sbol:access rdf:resource="http://sbols.org/v2#public"/>
      <sbol:definition rdf:resource="http://www.virtualparts.org/part/pIKELeftCassette_1"/>
    </sbol:Component>
  </sbol:component>
  <sbol:sequenceAnnotation>
    <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKE_Toggle_1/anno1">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKE_Toggle_1/anno1"/>
      <sbol:displayId>anno1</sbol:displayId>
      <sbol:location>
        <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKE_Toggle_1/anno1/range">
          <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKE_Toggle_1/anno1/range"/>
          <sbol:displayId>range</sbol:displayId>
          <sbol:start>1</sbol:start>
          <sbol:end>1285</sbol:end>
          <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
        </sbol:Range>
      </sbol:location>
      <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKE_Toggle_1/pIKELeftCassette_1"/>
    </sbol:SequenceAnnotation>
  </sbol:sequenceAnnotation>
  <sbol:sequenceAnnotation>
    <sbol:SequenceAnnotation rdf:about="http://www.virtualparts.org/part/pIKE_Toggle_1/anno2">
      <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKE_Toggle_1/anno2"/>
      <sbol:displayId>anno2</sbol:displayId>
      <sbol:location>
        <sbol:Range rdf:about="http://www.virtualparts.org/part/pIKE_Toggle_1/anno2/range">
          <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/pIKE_Toggle_1/anno2/range"/>
          <sbol:displayId>range</sbol:displayId>
          <sbol:start>1286</sbol:start>
          <sbol:end>2834</sbol:end>
          <sbol:orientation rdf:resource="http://sbols.org/v2#inline"/>
        </sbol:Range>
      </sbol:location>
      <sbol:component rdf:resource="http://www.virtualparts.org/part/pIKE_Toggle_1/pIKERightCassette_1"/>
    </sbol:SequenceAnnotation>
  </sbol:sequenceAnnotation>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://identifiers.org/uniprot/P42212">
  <sbol:persistentIdentity rdf:resource="http://identifiers.org/uniprot/P42212"/>
  <sbol:displayId>P42212</sbol:displayId>
  <dcterms:title>GFP</dcterms:title>
  <dcterms:description>GFP protein</dcterms:description>
  <sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#Protein"/>
  <sbol:role rdf:resource="http://identifiers.org/biomodels.sbo/SBO:0000011"/>
</sbol:ComponentDefinition>
<sbol:ComponentDefinition rdf:about="http://www.partsregistry.org/BBa_C0040">
  <sbol:persistentIdentity rdf:resource="http://www.partsregistry.org/BBa_C0040"/>

```

```

<sbol:displayId>BBa_C0040</sbol:displayId>
<dcterms:title>tetR</dcterms:title>
<dcterms:description>tetR coding sequence</dcterms:description>
<sbol:type rdf:resource="http://www.biopax.org/release/biopax-level3.owl#DnaRegion"/>
<sbol:role rdf:resource="http://identifiers.org/so/SO:0000316"/>
<sbol:sequence rdf:resource="http://www.virtualparts.org/part/BBa_C0040"/>
</sbol:ComponentDefinition>
<sbol:Sequence rdf:about="http://www.virtualparts.org/part/BBa_J61101">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/BBa_J61101"/>
  <sbol:displayId>BBa_J61101</sbol:displayId>
  <sbol:elements>aaagacaggacc</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://www.virtualparts.org/part/BBa_J61120">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/BBa_J61120"/>
  <sbol:displayId>BBa_J61120</sbol:displayId>
  <sbol:elements>aaagacaggacc</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://www.virtualparts.org/part/BBa_E0040">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/BBa_E0040"/>
  <sbol:displayId>BBa_E0040</sbol:displayId>
  <sbol:elements>atgcgtaagggaagaacttttctactggagttgttcccaattcttgttgaattagatgggtgatgttaattgg
gcacaaattttctgtcagtgaggagggatgaaggatgcaacatacggaaaacttacccttaattttatgtcactactggaaaac
tacctgttccatggccaaacttgtcactactttcggttatgtgttcaatgctttgagagataccagatcatatgaaacagcat
gactttttcaagagtgcctgcccgaagggttatgtacaggaagaactatattttcaagatgacgggaactacaagacacgtgc
tgaagtcaagtttgaagggtgatacccttgttaataagaatcgagttaaaaggattgtattttaaagaagatggaacattcttggac
acaaattggaatacaactataactcacacaatgtatacatcatggcagacaaacaaaagaatggaatcaagtttaacttcaaaatt
agacacaacattgaagatggaagcgttcaactagcagaccattatcaacaaaatactccaattggcgatggccctgtccctttacc
agacaaccattacgttccacacaatctgccctttcgaagatcccaacgaaaagagagaccacatggtccttcttgagtttgtaa
cagctgctgggtattacatggcatggatgaactatacaataataa</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://www.virtualparts.org/part/ECK120033736">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/ECK120033736"/>
  <sbol:displayId>ECK120033736</sbol:displayId>
  <sbol:elements>ttcagccaaaaaacttaagaccgcggtctgtccactacctgcagtaatgcgtggacaggatcgccgggtt
ttctttctcttctcaa</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://www.virtualparts.org/part/BBa_R0010">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/BBa_R0010"/>
  <sbol:displayId>BBa_R0010</sbol:displayId>
  <sbol:elements>tccctatcagtgatagagattgacatccctatcagtgatagagatactgagcac</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://www.virtualparts.org/part/BBa_R0040">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/BBa_R0040"/>
  <sbol:displayId>BBa_R0040</sbol:displayId>
  <sbol:elements>tccctatcagtgatagagattgacatccctatcagtgatagagatactgagcac</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://www.virtualparts.org/part/BBa_J61130">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/BBa_J61130"/>
  <sbol:displayId>BBa_J61130</sbol:displayId>
  <sbol:elements>aaagaacgaca</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://www.virtualparts.org/part/BBa_C0040">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/BBa_C0040"/>
  <sbol:displayId>BBa_C0040</sbol:displayId>
  <sbol:elements>atgtccagattagataaaagtaaagtattaacagcgcatagagctgcttaatgaggtcggaatcgaaagg
tttaacaacccgtaaaactcgccagaagctaggtgtagagcagcctacattgtattggcatgtaaaaataagcgggctttgtctg
acgccttagccattgagatgttagataggcaccatactcacttttgccctttagaaggggaaagctggcaagattttttacgtaat
aacgctaaaagtttttagatgtgctttactaagtcatcgcatggagcaaaagtacatttaggtacacggcctacagaaaaacagta
tgaaactctcgaaaaatcaattagcctttttatgccaacaaggtttttcactagagaatgcattatatgcactcagcgtgtggggc
attttacttttaggttgcgtattggaagatcaagagcatcaagtcgctaagaagaagaaaggaacacactactactgatagatgcgcg
ccattattacgacaagctatcgaaattatttgatcaccaaggtgcagagccagccttcttattcgcccttgaattgatcatatgcgg
attagaaaaacaacttaaatgtgaaagtggtgcgctgcaaacgacgaaaactacgcttttagtgcttaataa</sbol:elements>

```

```

ts>
<sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://www.virtualparts.org/part/BBa_C0012">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/BBa_C0012"/>
  <sbol:displayId>BBa_C0012</sbol:displayId>
  <sbol:elements>atggatgaatgtgaaccagtaacgttatcacgatgtcgagagatgccggtgtctcttatcagaccgtttccc
gcgtggtgaaccaggccagccacgtttctgcgaaaacgcgggaaaaagtggagcggcgatggcggagctgaattacattcccaaccg
cgtggcacaacaactggcgggcaaacagtcgttgcgtgattggcgttgcacacctccagtcgtggccctgcacgcgcgtcgcaaatgtc
gcggcgattaaatctcgccgatcaactgggtgccagcgtggtggtgctgatggtagaacgaagcggcgtcgaagcctgtaaagcgg
cgggtgcacaatcttctcgcaacgcgtcagtggtgatcattaaactatccgctggatgaccaggatgccattgctgtggaagctgc
ctgcactaatgttcggcggttatttcttgatgtctctgaccagacacccatcaacagtattattttctccatgaagacggtagcga
ctgggcgtggagcatctggtcgcatgtgggtcaccagcaaatcgcgctgttagcgggcccattaagttctgtctcggcgctctgcgtc
tggctggctggcataaataatctcactcgcaatcaattcagccgatagcggaaacgggaaggcgactggagtgccatgtccggttttca
acaaaccatgcaaatgctgaatgagggcatcttccactgcgatgctggttccaacgatcagatggcgctgggcgaatgcgcgcc
attaccgagtcgggctgcgcttgggtgcgatatctcggtagtgggatacgcgataccgaagacagctcatgttatatcccgcgt
taaccacatcaaacaggattttcgctgctggggcaaacacgcgtggacgccttgcgcaactctctcagggccaggcgggtgaaggg
caatcagctgttgcccgtctcactggtgaaagaaaaaccacctggcgcccaatgcgcaaacgcctctctccgcgcttggccgat
tcattaatgcagctggcagcaggtttcccgactggaaagcgggcaggctgcaaacgacgaaaactacgcttttagtagcttaataa</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
<sbol:Sequence rdf:about="http://www.virtualparts.org/part/ECK120029600">
  <sbol:persistentIdentity rdf:resource="http://www.virtualparts.org/part/ECK120029600"/>
  <sbol:displayId>ECK120029600</sbol:displayId>
  <sbol:elements>ttcagcaaaaaacttaagaccgcggtcttgtccactaccttcagtaaatgcggtggacaggatcggcggtt
ttctttctcttctcaa</sbol:elements>
  <sbol:encoding rdf:resource="http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html"/>
</sbol:Sequence>
</rdf:RDF>

```