Systems Biology Graphical Notation: Process Description language Level 1

Version 2.0

Date: May 4, 2012

User Manual

Chapter 1

Introduction

With the rise of systems and synthetic biology, the use of graphical representations of pathways and networks to describe biological systems has become pervasive. It was therefore important to use a consistent notation that would allow people to interpret those maps easily and quickly, without the need of extensive legends. Furthermore, activities like synthetic biology, that reconstruct biological systems, need to exchange their descriptions unambiguously, as engineers exchange circuit diagrams.

The goal of the Systems Biology Graphical Notation (SBGN) is to standardize the graphical/visual representation of biochemical and cellular processes. SBGN defines comprehensive sets of symbols with precise semantics, together with detailed syntactic rules defining their use. It also describes the manner in which such graphical information should be interpreted. SBGN is made up of three different and complementary languages [1]. This document presents the graphical elements composing the *Process Description language* of SBGN. It is not a normative description, but rather a document aimed at end-users. People, such as software developers, looking for a normative description of SBGN Process Descriptions should rather read the technical specification of the language [**?**].

1.1 Overview of SBGN Process Descriptions

To quickly describe what SBGN Process Description language is about, let's give a brief overview of some of the relevant concepts with the help of an example shown in Figure 1.1. It is a simple map for part of a mitogen-activated protein kinase (MAPK) cascade. The larger nodes in the figure (some of which are in the shape of rounded rectangles and others in the shape of circles) represent biological materials—things like macromolecules and simple chemicals. The biological materials are altered via processes, which are indicated in Process Description language by lines with arrows and other decorations. In this particular map, all of the processes happen to be the same: processes catalyzed by biochemical entities. The directions of the arrows indicate the direction of the processes; for example, unphosphorylated RAF kinase proceeds to phosphorylated RAF kinase via a process catalyzed by RAS. Although ATP and ADP are shown as incidental to the phosphorylations on this particular graph, they are involved in the same process as the proteins getting phosphorylated. The small circles on the nodes for RAF and other entity pools represent state variables (in this case, phosphorylation sites).

The essence of the Process Descriptions is *change*: it shows how different entities in the system process from one form to another. The entities themselves can be many different things. In the example of Figure 1.1, they are either pools of macromolecules or pools of simple chemicals, but as will become clear later in this chapter, they can be other conceptual and material constructs as well. Note also that we speak of *entity pools* rather than individuals; this is because in biochemical network models, one does not focus on single molecules, but rather collections of molecules of the same kind. The molecules in a given pool are considered indistinguishable from each other. The way in which one type of entity is transformed into another is conveyed by a *process node* and arcs between entity pool nodes and process nodes indicate an influence by the entities on the processes. In the

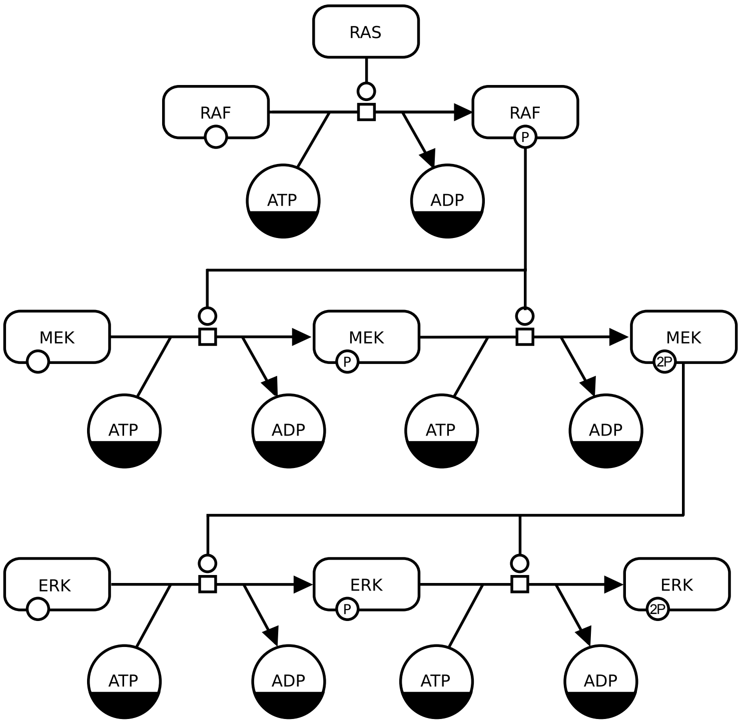


Figure 1.1: *This example of a Process Description uses two kinds of entity pool nodes: one for pools of different macromolecules (Section 2.1.2) and another for pools of simple chemicals (Section 2.1.3). Most macromolecule nodes in this map are adorned with state variables (Section 2.2.2) representing phosphorylation states. This map uses one type of process node, the process node (Section* **??***), and three kind of connecting arc, consumption (Section* **??***), production (Section* **??***) and catalysis (Section* **??***). Finally, some entity pool nodes have dark bands along their bottoms; these are clone markers (Section 2.2.3) indicating that the same pool nodes appear multiple times in the map.*

case of Figure 1.1, those arcs describe consumption Section **??**, production Section **??** and catalysis Section **??**, but others are possible. Finally, nodes in Process Descriptions are usually not repeated; if they do need to be repeated, they are marked with *clone markers*—specific modifications to the appearance of the node (Section 2.2.3). The details of this and other aspects of Process Description notation are explained in the rest of this chapter.

1.2 SBGN levels and versions

It was clear at the outset of SBGN development that it would be impossible to design a perfect and complete notation right from the beginning. Apart from the prescience this would require (which, sadly, none of the authors possess), it also would likely need a vast language that most newcomers would shun as being too complex. Thus, the SBGN community followed an idea used in the development of other standards, i.e. stratify language development into levels.

A *level* of one of the SBGN languages represents a set of features deemed to fit together cohesively, constituting a usable set of functionality that the user community agrees is sufficient for a reasonable set of tasks and goals. Within *levels*, *versions* represent small evolution of a language, that may involve new glyphs, refined semantics, but no fundamental change of the way maps are to be generated and interpreted. In addition new versions should be backwards compatible, i.e., Process Description maps that conform to an earlier version of the Process Description language within the same level should still be valid. This does not apply to a new levels.

Capabilities and features that cannot be agreed upon and are judged insufficiently critical to require inclusion in a given level, are postponed to a higher level or version. In this way, the development of SBGN languages is envisioned to proceed in stages, with each higher levels adding richness compared to the levels below it.

1.3 How to get more information

The normative description of the language is the technical specification [**?**]. It is available from the SBGN website (`http://sbgn.org/`). This website is a portal for all things to the notation. In addition to the specifications, there are examples of maps, FAQs, and informations on part and forthcoming meetings.

The easiest and best way to get involved in SBGN discussions is to join the sbgn-discuss@caltech.edu mailing list. If you only want the announcements of meetings and new specifications, you can join the very low flux mailing list sbgn-announce@lists.sf.net instead.

Chapter 2

Symbols used in SBGN Process Descriptions

An SBGN Process Description map is mainly is bipartite graph, i.e. it is made up of two types of nodes that connect in an alternate way (some exceptions are described below, e.g. when *logical operators* or *tag* are used). The two types of nodes are the *process nodes* and the *entity pools nodes*, representing the things that are modified by processes. These nodes are connected by arcs. In addition, the *entity pools nodes* can be contained in *compartments*.

2.1 Entity pool nodes

An entity pool is a population of entities that cannot be distinguished from each other, when it comes to the SBGN Process Description Level 1 map. For instance all the molecular entities that fulfill the same role in a given process form an entity pool. As a result, an entity pool can represent different granularity levels, such as all the proteins, all the instances of a given protein, only certain forms of a given protein. It really depends on what we want to represent. To belong to different compartments is sufficient to belong to different entity pools. Calcium ions in the endoplasmic reticulum and calcium ions in the cytosol belong to different entity pools when it comes to representing calcium release from the endoplasmic reticulum.

The Process Description language contains six glyphs representing classes of material entities: *unspecified entity* (Section 2.1.1), *simple chemical* (Section 2.1.3), *macromolecule* (Section 2.1.2), *nucleic acid feature* (Section 2.1.4), and *complex* (Section 2.1.5). (Specific types of macromolecules, such as protein, RNA, DNA, polysaccharide, and specific simple chemicals are not defined by Process Description but may be part of future levels of SBGN). In addition to the material entities, the Process Description language represents two conceptual entities: An absorbing pool, called *source and sink* (Section 2.1.6), and a *perturbing agent* (Section 2.1.7). Material and conceptual entities can optionally carry auxiliary units such as *units of information* (Section 2.2.1), *state variables* (Section 2.2.2) and *clone markers* (Section 2.2.3).

2.1.1 Glyph: *Unspecified entity*

The simplest type of EPN is the *unspecified entity*: one which type is unknown or simply not relevant to the purposes of the map. This arises, for example, when the existence of the entity has been inferred indirectly, or when the entity is merely a construct introduced for the needs of a map, without direct biological equivalent. For cases where the identity of the entities composing the pool *is* known, there exist other, more specific glyphs described below in the manual.

An *unspecified entity* is represented by an elliptic container, as shown in Figure 2.1. Note that this must remain an ellipse to avoid confusion with the Simple Chemical glyph, which is a circle (c.f. 2.1.3).

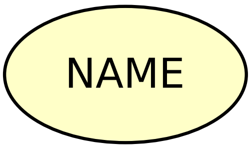


Figure 2.1: *The Process Description glyph for* unspecified entity.

2.1.2 Glyph: *Macromolecule*

Many biological processes involve *macromolecules*: biochemical substances that are built up from the covalent linking of pseudo-identical units. Examples of macromolecules include proteins, nucleic acids (RNA, DNA), and polysaccharides (glycogen, cellulose, starch, etc.). Attempting to define a separate glyph for all of these different molecules would lead to an explosion of symbols in SBGN, so instead, SBGN Process Description Level 1 defines only one glyph for all macromolecules. The same glyph is to be used for a protein, a nucleic acid, a complex sugar, and so on. The exact nature of a particular macromolecule in a map is then clarified using its label and decorations, as will become clear below. A *macromolecule* is represented by a rectangular container with rounded corners, as illustrated in Figure 2.2.

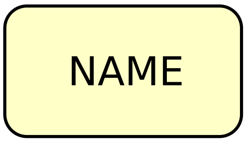


Figure 2.2: *The Process Description glyph for* macromolecule.

2.1.3 Glyph: *Simple chemical*

In SBGN Process Descriptions, a simple chemical is defined as the opposite of a macromolecule (Section 2.1.2): it is a chemical compound that is *not* formed by the covalent linking of pseudo-identical residues. Examples of simple chemicals are an atom, a monoatomic ion, a salt, a radical, a solid metal, a crystal, etc. A *simple chemical* is represented by a circular container, as depicted in Figure 2.3. To avoid confusion with the Unspecified Entity (2.1.1), this glyph must remain a circle and cannot be deformed into an eclipse.

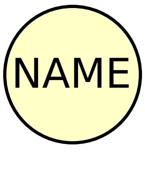


Figure 2.3: *The Process Description glyph for* simple chemical.

2.1.4 Glyph: *Nucleic acid feature*

The *Nucleic acid feature* construct in SBGN is meant to represent a fragment of a macromolecule carrying genetic information. A common use for this construct is to represent a gene or a transcript. The label of this EPN and its *units of information* (see section **??**) are often important for making the purpose clear to the reader of a map. A *nucleic acid feature* is represented by a rectangular container whose bottom half has rounded corners, as shown in Figure 2.4. This design reminds that we are fundamentally dealing with a unit of information, but this information is carried by a macromolecule.

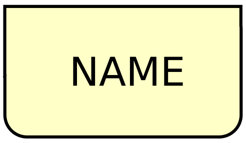


Figure 2.4: *The Process Description glyph for* nucleic acid feature.

2.1.5 Glyph: *Complex*

A *complex* node represents a biochemical entity composed of other biochemical entities, whether macromolecules, simple chemicals, multimers, or other complexes. The resulting entity may have its own identity, properties and function in an SBGN map. A *complex* possesses its own container box surrounding the juxtaposed container boxes of its components. This container box is a rectangle with cut-corners (an octagonal box with sides of two different lengths). The size of the cut-corners are adjusted so that there is no overlap between the container and the components. The container boxes of the components must not overlap.

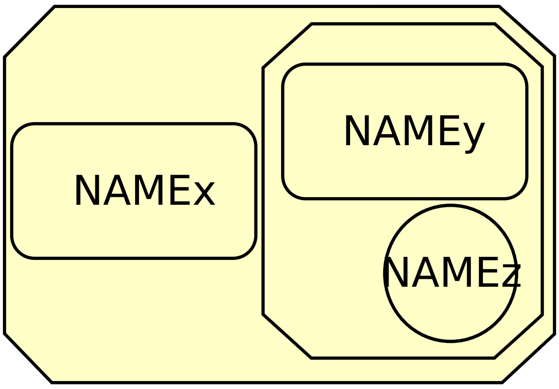


Figure 2.5: *The Process Description glyph for* complex.

2.1.6 Glyph: *Source* and *Sink*

It is useful to have the ability to represent the creation of an entity or a state from an unspecified source, that is, from something that one does not need or wish to make precise. For instance, in a model where the production of a protein is represented, it may not be desirable to represent all of the amino acids, sugars and other metabolites used, or the energy involved in the protein's creation. Similarly, we may not wish to bother representing the details of the destruction or decomposition of some biochemical species into a large number of more primitive entities, preferring instead to simply say that the species "disappears into a sink". Yet another example is that one may need to represent an input (respectively, output) into (resp. from) a compartment without explicitly representing a transport process from a source (resp. to a target).

For these and other situations, SBGN defines a glyph for explicitly representing the involvement of an unspecified source or sink. A *source* or *sink* is represented by the mathematical symbol for "empty set", that is, a circle crossed by a bar linking the upper-right and lower-left corners of an invisible square drawn around the circle ($\emptyset$). Figure 2.6 illustrates this. The symbol should be linked to one and only one edge in a map. The symbol used in SBGN is borrowed from the mathematical symbol for "empty set", but it is important to note that it does not actually represent a true absence of everything or a physical void—it represents the absence of the corresponding structures in the model, that is, the fact that these sources or sinks are conceptually outside the scope of the map.

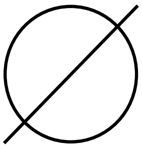


Figure 2.6: *The* source *and* sink *glyphs.*

2.1.7 Glyph: *Perturbing agent*

Biochemical networks can be affected by external influences. Those influences can be the effect of well-defined physical perturbing agents, such as a light pulse or a change in temperature; they can also be more complex and not well-defined phenomena, for instance the outcome of a biological process, an experimental setup, or a mutation. For these situations, SBGN provides the *perturbing agent* glyph. It is an EPN, and represents the amount to perturbing agent applied to a process. A *perturbing agent* is represented by a modified hexagon having two opposite concave faces, as illustrated in Figure 2.7.

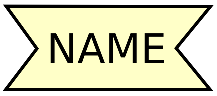


Figure 2.7: *The Process Description glyph for* perturbing agent.

2.2 Decorations of the entity pool nodes

SBGN Process Description provides glyphs that decorate other glyphs, providing additional information that may be useful to the reader. These can provide annotation (*unit of information*), state information (*state variable*) or indicate duplication of entity pool nodes (*clone marker*).

2.2.1 Glyph: *Unit of information*

When representing biological entities, it is often necessary to convey some abstract information about the entity's function that does is not related to its role in the map. The *unit of information* is a decoration that can be used in this situation to add information to an EPN. Some example uses include: characterizing a logical part of an entity such as a functional domain (a binding domain, a catalytic site, a promoter, etc.), or the information encoded in the entity (an exon, an open reading frame, etc.). A *unit of information* can also convey information about the physical environment, or the specific type of biological entity it is decorating. A *unit of information* is represented by a rectangle overlapping the border of the *EPN* being annotated.

The label carried by *unit of information* defines the information it carries. For certain predefined types of information having controlled vocabularies associated with them, SBGN defines specific prefixes that must be included in the label to indicate the type of information in question. The controlled vocabularies predefined in SBGN Process Description Level 1 are described in Section **??**.

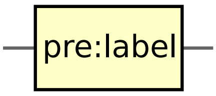


Figure 2.8: *The Process Description glyph for* unit of information.

2.2.2 Glyph: *State variable*

Many biological entities, such as molecules, can exist in different *states*, meaning different physical or informational configurations. These states can arise for a variety of reasons. For example, macromolecules can be subject to post-synthesis modifications, wherein residues of the macromolecules (amino acids, nucleosides, or glucid residues) are modified through covalent linkage to other chemicals. Other examples of states are alternative conformations as in the closed/open/desensitized conformations of a transmembrane channel, and the active/inactive forms of an enzyme.

SBGN provides a means of associating one or more *state variables* with an entity; each such variable can be used to represent a dimension along which the state of the overall entity can vary. When an entity can exist in different states, the state of the whole entity (i.e., the SBGN object) can be described by the current values of all its *state variables*, and the values of the *state variables* of all its possible components, recursively. A *state variable* is represented by an elliptical container overlapping the border of the *EPN* being annotated.

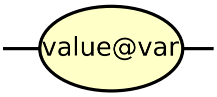


Figure 2.9: *The Process Description glyph for* state variable.

A *state variable* does not necessarily have to be Boolean-valued. For example, an ion channel can possess several conductance states; a receptor can be inactive, active and desensitized; and so on. As another example, a *state variable* “ubiquitin” could also carry numerical values corresponding to the number of ubiquitin molecules present in the tail. However, in all cases, a *state variable* on an EPN can only take *one* defined value. Further, an EPN’s *state variable* should always be displayed and always set to a value. An “empty” *state variable* is a *state variable* that is set to the value “unset”, it is not a *state variable* with no value. Note that the value “unset” is *not* synonymous to “any value” or “unknown value”.

2.2.3 Glyph: *Clone marker*

It is sometimes necessary to represent the same *EPN* several times. Otherwise, the resulting graph is so tightly connected that the map becomes unreadable. An example would be the representation of currency molecules such as ATP. However, we must indicate the fact, so that a reader knows the processes involving this particular glyph are not the only processes involving the *EPN*. If an *EPN* is duplicated on a map, we therefore mark all its graphical reprensation with a *clone marker* auxiliary unit. This marker provides the reader with a visual indication that this node has been cloned, and that at least one other occurrence of the *EPN* can be found in the map (or in a submap; see Section **??**). The clone marker takes two forms, simple and labeled, depending on whether the node being cloned can carry state variables. Note that an *EPN* belongs to a single compartment. If two glyphs labelled “X” are located in two different compartments, such as ATP in cytosol and ATP in mitochondrial lumen, they represent different *EPNs*, and therefore do not need to be marked as cloned (and if they are, they are not part of the same clone).

The simple (unlabeled) *clone marker* is a portion of the surface of an *EPN* that has been modified visually through the use of a different shade, texture, or color. Figure 2.10 illustrates this. The *clone marker* occupies the lower part of the *EPN*. The filled area must be smaller than the unfilled one.

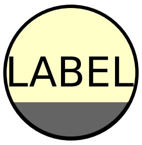


Figure 2.10: *The Process Description glyph for* simple clone marker *applied to a* simple chemical

Unlike the *simple clone marker*, the *labeled clone marker* includes (unsurprisingly, given its name) an identifying label that can be used to identify equivalent clones elsewhere in the map. This is particularly useful for stateful *EPNs*, because these can have a large number of state variables displayed and therefore may be difficult to visually identify as being identical. The filled area must be smaller than the unfilled one, but the be large enough to have a height larger than the *clone marker*'s label (cf below).

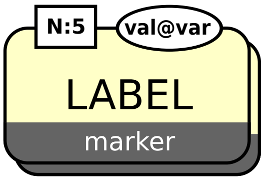
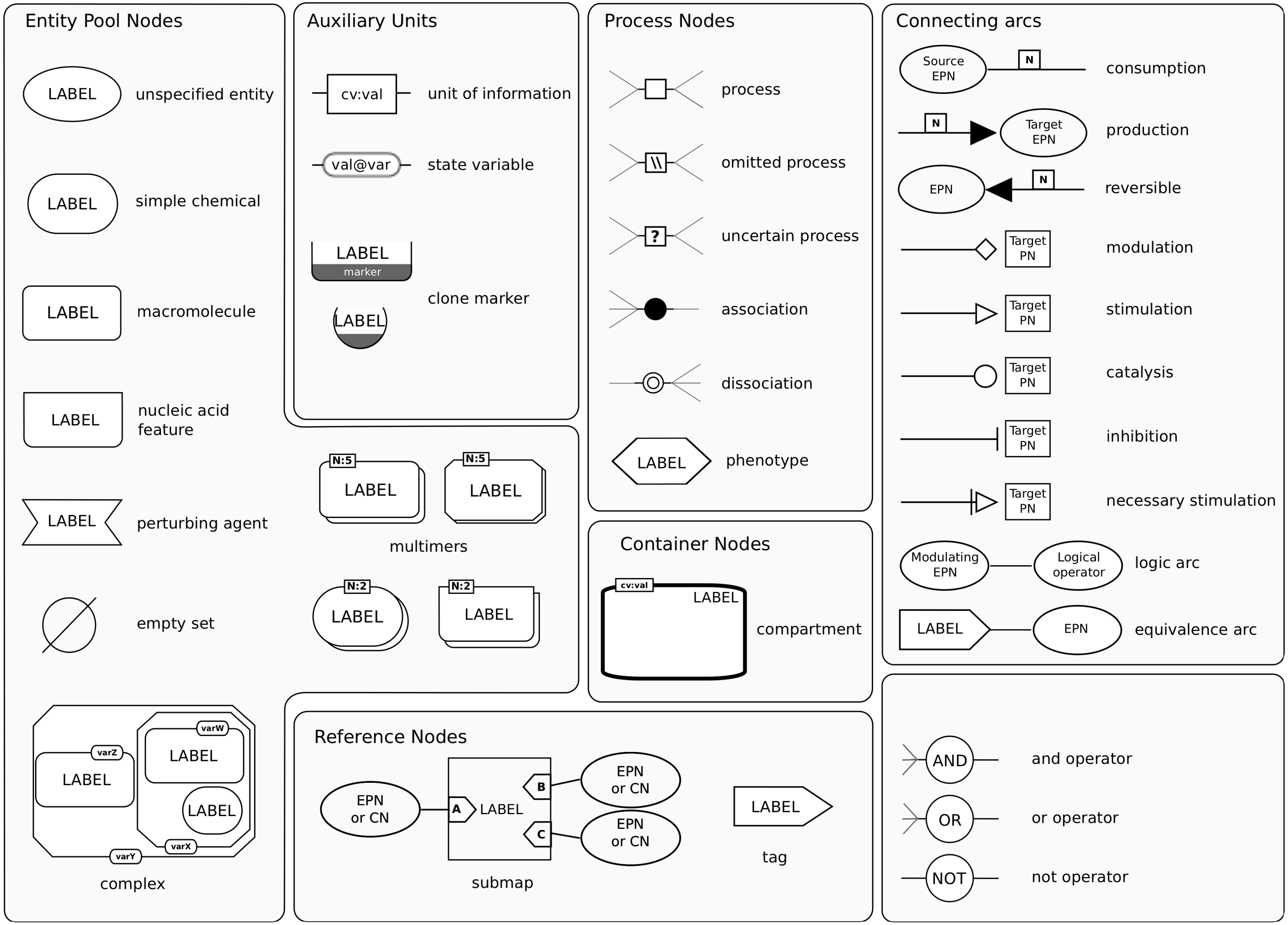


Figure 2.11: *The Process Description glyph for* labeled clone marker *applied to a* multimer *of* macromolecules.

Appendix A

Reference card

Print the summary of SBGN symbols on the next page for a quick reference.


Entity Pool Nodes
LABEL
unspecified entity
LABEL
simple chemical
LABEL
macromolecule
LABEL
nucleic acid feature
LABEL
perturbing agent
empty set
LABEL
varZ
LABEL
varW
LABEL
varX
varY
complex
Auxiliary Units
cv:val
unit of information
val@var
state variable
LABEL
marker
LABEL
clone marker
N:5
LABEL
N:5
LABEL
multimers
N:2
LABEL
N:2
LABEL
Reference Nodes
EPN or CN
A
LABEL
B
C
EPN or CN
EPN or CN
submap
LABEL
tag
Process Nodes
process
omitted process
?
uncertain process
association
dissociation
LABEL
phenotype
Container Nodes
cv:val
LABEL
compartment
Connecting arcs
Source EPN
N
consumption
N
Target EPN
production
EPN
N
reversible
Target PN
modulation
Target PN
stimulation
Target PN
catalysis
Target PN
inhibition
Target PN
necessary stimulation
Modulating EPN
Logical operator
logic arc
LABEL
EPN
equivalence arc
AND
and operator
OR
or operator
NOT
not operator

Bibliography

[1] N Le Novère, Michael Hucka, Huaiyu Mi, Stuart Moodie, Falk Schreiber, Anatoly Sorokin, Emek Demir, Katja Wegner, Mirit I Aladjem, M Sarala Wimalaratne, Frank T Bergman, Ralph Gauges, Peter Ghazal, Hideya Kawaji, Lu Li, Yukiko Matsuoka, Alice Villeger, Sarah E Boyd, Laurence Calzone, Mèlanie Courtot, Ugur Dogrusoz, Tom C Freeman, Akira Funahashi, Samik Ghosh, Akiya Jouraku, Sohyoung Kim, Fedor Kolpakov, Augustin Luna, Sven Sahle, Esther Schmidt, Steven Watterson, Guanming Wu, Igor Goryanin, Douglas B Kell, Chris Sander, Herbert Sauro, Jacky L Snoep, Kurt Kohn, and Hiroaki Kitano. The systems biology graphical notation. *Nat Biotechnol*, 27(8):735–741, 2009. 10.1038/nbt.1558.