Systems Biology Graphical Notation: Entity Relationship Level 1

Draft of September 22, 2008

Disclaimer: This is a working draft of the SBGN Process Diagram Level 1 specification. It is not a normative document.

To discuss any aspect of SBGN, please send your messages to the mailing list sbgn-discuss@sbgn.org*. To get subscribed to the mailing list or to contact us directly, please write to* sbgn-team@sbgn.org*.*

Contents

Chapter 1

What is the Systems Biology Graphical Notation?

The goal of the **S**ystems **B**iology **G**raphical **N**otation (SBGN) is to standardize the graphical/visual representation of essential biochemical and cellular processes studied in systems biology. SBGN defines a comprehensive set 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.

Standardizing graphical notations for describing biological interactions is an important step towards the efficient and accurate transmission of biological knowledge between different communities. Traditionally, diagrams representing interactions among genes and molecules have been drawn in an informal manner, using simple unconstrained shapes and edges such as arrows. Until the development of SBGN, no standard agreed-upon convention existed defining exactly how to draw such diagrams in a way that helps readers interpret them consistently, correctly, and unambiguously. By standardizing the visual notation, SBGN can serve as a bridge between different communities such as computational and experimental biologists, and even more broadly in education, publishing, and more.

For SBGN to be successful, it must satisfy a majority of technical and practical needs, and must be embraced by the community of researchers in biology. With regards to the technical and practical aspects, a successful visual language must meet at least the following goals:

1. Allow the representation of diverse biological objects and interactions;
2. Be semantically and visually unambiguous;
3. Allow implementation in software that can aid the drawing and verification of diagrams;
4. Have semantics that are sufficiently well defined that software tools can convert graphical models into formal models, suitable for analysis if not for simulation;
5. Be unrestricted in use and distribution, so that the entire community can freely use the notation without encumbrance or fear of intellectual property infractions.

This document defines the *Entity Relationship* visual language of SBGN. As explained more fully in Section 1.2, Entity Relationship diagrams are one of three views of a model offered by SBGN. It is the product of many hours of discussion and development by many individuals and groups. In the following sections, we describe the background, motivations, and context of Entity Relationship diagrams.

1.1 History of SBGN development

Although problems surrounding the representation of biological pathways has been discussed for a long time, see for instance [**?**], the effort to create a well-defined visual notation was pioneered by Kurt Kohn with his Molecular Interaction Map (MIM), a notation defining symbols and syntax to describe the interactions of molecules [1]. MIM is essentially a variation of the entity-relationship diagrams [2]. Kohn's work was followed by numerous other attempts to define both alternative notations for diagramming cellular processes (e.g., the work of Pirson and colleagues [3], BioD [4], Patika [**?**, **?**], and others), as well as extensions of Kohn's notation (e.g., the Diagrammatic Cell Language of Maimon and Browning [5]).

Kitano originated the idea of having multiple views of the *same* model. This addresses two problems: no single view can satisfy the needs of all users, and a given view can only represent a subset of the semantics necessary to express biological knowledge. Kitano proposed the development of process diagrams, entity-relationship diagrams, timing charts (to describe temporal changes in a system), and abstract flow charts [6]. The Process Diagram notation was the first to be fully defined using a well-delineated set of symbols and syntax [7]. It led to a desire to establish a unified standard for graphical representation of biochemical entities, and from this arose the current SBGN effort. Separately and roughly concurrently, other groups designed similar notations, for example the Edinburgh Pathway Notation [8] or Patika [**?**, **?**]. All of these efforts began to attract attention as more emphasis in biological research was placed on networks of interactions and not just characterization of individual entities.

In 2005, thanks to funding from the Japanese agency *The New Energy and Industrial Technology Development Organization* (NEDO, `http://www.nedo.go.jp/`), Kitano initiated the Systems Biology Graphical Notation (SBGN) project as a community effort. The first SBGN workshop was held in February 2006 in Tokyo, with over 30 participants from major organizations interested in this effort. From the in-depth discussions held during that meeting emerged a set of decisions that are the basis of the current SBGN specification. These decisions are:

- SBGN should be made up of two different visual grammars, describing Entity Relationship and Process Diagram diagrams (called *State Transition* diagrams at the time). See Section 1.2.

- In order to promote wide acceptance, the initial version(s) of SBGN should stick to at most a few dozens symbols that non-specialists could easily learn.

The second SBGN workshop was held in October, 2006, in Yokohama, Japan. This meeting featured the first technical discussions about which symbols to include in SBGN Level 1, as well as discussions about the syntax, semantics, and layout of graphs. A follow-up technical meeting was held in March, 2007, in Heidelberg, Germany; the participants of that meeting fleshed out most of the design of SBGN. The third SBGN workshop, held in Long Beach in October, 2007, was dedicated to reaching agreement on the final outstanding issues of notation and syntax. The participants of that meeting collectively realized that a third language would be necessary: the Activity Flow diagrams. The specification for the Process Diagram language was finalized and largely completed during a follow-up technical meeting held in Okinawa, Japan, in January, 2008. At this last meeting, attendees also held the first in-depth discussions about the syntax of the Entity Relationship language.

The specification for SBGN Process Diagram Level 1 was publicly released on August 23$^{\mathrm{rd}}$ 2008 during the ICSB in Göteborg [9].

SBGN workshops are an opportunity for public discussions about SBGN, allowing interested persons to learn more about SBGN and help identify needs and issues. More meetings are expected to be held in the future, long after this specification document has been issued.

1.2 The three languages of SBGN

Readers may well wonder, why are there *three* languages in SBGN? The reason is that this approach solves a problem that was found insurmountable any other way: attempting to include all relevant facets of a biological system in a single diagram causes the diagram to become hopelessly complicated and incomprehensible to human readers.

The three different notations in SBGN correspond to three different *views* of the same model. These views are representations of different classes of information, as follows:

1. *Process Diagram*: the causal sequences of molecular processes and their results
2. *Entity Relationship*: the interactions between entities irrespective of sequence
3. *Activity Flow*: the flux of information going from one entity to another

In the Process Diagram view, each node in the diagram represents a given *state* of a species, and therefore a given species may appear multiple times in the same diagram if it represents the same entity in different states. Conversely, in the Entity Relationship view, a given species appears only once in a diagram. Process Diagrams are suitable for following the temporal aspects of interactions, and are easy to understand. The drawback of the Process Diagram, however, is that because the same entity appears multiple times in one diagram, it is difficult to understand which interactions actually exist for the entity. Conversely, Entity Relationship diagrams are suitable for understanding relationships involving each molecule, but the temporal course of events is difficult or impossible to follow because Entity Relationship diagrams do not describe the sequence of events.

Process Diagrams can quickly become very complex. Moreover, when diagramming a biochemical network, one often wants to ignore the biochemical basis underlying the action of one entity on the activity of another. A common desire is to represent only the flow of activity between nodes, without representing the transitions in the states of the nodes. This is the motivation for the creation of the Activity Flow view. Activity Flow diagrams permit the use of *modulation*, *stimulation* and *inhibition* and allow them to point to State/Entity nodes rather than process nodes. The Activity Flow view is thus a hybrid between Process Diagram and Entity Relationship diagrams. It is particularly convenient for representing the effect of perturbations, whether genetic or environmental in nature.

A recurring argument in SBGN development is that these these three types of diagrams should be merged into one. Unfortunately, each view has such different meanings that merging them would compromise the robustness of the representation and destroy the mathematical integrity of the notation system. While having three different notations makes the overall system more complex, much of the complexity and increase in burden on learning is mitigated by reusing most of the same symbols in all three notations. It is primarily the syntax and semantics that change between the different views, reflecting fundamental differences in the underlying mathematics of what is being described.

1.3 SBGN levels

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 require a vast language that most newcomers would shun as being too complex. Thus, the SBGN community followed an idea used in the development of the Systems Biology Markup Language (SBML; [10]): stratify language development into levels.

A *level* of SBGN 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. 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. In this way, SBGN development is envisioned to proceed in stages, with each higher SBGN level adding richness compared to the levels below it.

1.4 Developments, discussions, and notifications of updates

The SBGN website (http://sbgn.org) is a portal for all things related to SBGN. It provides a web forum interface to the SBGN discussion list (sbgn-discuss@sbgn.org) and information about how anyone may subscribe to it. The easiest and best way to get involved in SBGN discussions is to join the mailing list and participate.

Face-to-face meetings of the SBGN community are announced on the website as well as the mailing list. Although no set schedule currently exists for workshops and other meetings, we envision holding at least one public workshop per year. As with other similar efforts, the workshops are likely to be held as satellite workshops of larger conferences, enabling attendees to use their international travel time and money more efficiently.

Notifications of updates to the SBGN specification are also broadcast on the mailing list and announced on the SBGN website.

Chapter 2

Entity Relationship Glyphs

[Note on the color code: The glyphs that have been thorougly discussed, and are considered frozen, are represented in blue. The glyphs that have been thorougly discussed, but are still posing problems are represented in green. The glyphs that have been proposed but for which in-depth discussion is yet to come are represented in red.]

This chapter provides a catalog of the graphical symbols available for representing entities in Entity Relationship diagramss. There are different classes of glyphs corresponding to different classes of material or conceptual entities, containers, processes, connecting arcs, and logical operators. In Chapter **??** beginning on page **??**, we describe the rules for combining these glyphs into a legal SBGN Entity Relationship, and in Chapter **??** beginning on page **??**, we describe requirements and guidelines for the way that diagrams are visually organized.

2.1 Overview

To set the stage for what follows in this chapter, we first give a brief overview of some of the concepts in the Entity Relationship notation with the help of an example shown in Figure 2.1.

2.2 Controlled vocabularies used in SBGN Entity Relationship Level 1

Some glyphs in SBGN Entity Relationship diagrams can contain particular kinds of textual annotations conveying information relevant to the purpose of the glyph. These annotations are carried by *units of information* (Section 2.3.10) or *state variable* (Section 2.3.11).

The text that appears as the unit of information decorating an Entity Node (EN) must be prefixed with a controlled vocabulary term indicating the type of information being expressed. The prefixes are mandatory. Without the use of controlled vocabulary prefixes, it would be necessary to have different glyphs to indicate different classes of information; this would lead to an explosion in the number of symbols needed.

In the rest of this section, we describe the controlled vocabularies (CVs) used in SBGN Entity Relationship Level 1. They cover the following categories of information: an EPN's material type, an EPN's conceptual type, covalent modifications on macromolecules, the physical characteristics of compartments, and cardinality (e.g., of multimers). In each case, some CV terms are predefined by SBGN, but unless otherwise noted, *they are not the only terms permitted*. Authors may use other CV values not listed here, but in such cases, they should explain the terms' meanings in a figure legend or other text accompanying the diagram.

2.2.1 Entity Pool Node material types

The material type of an Entity Pool Node (EPN) indicates its chemical structure. A list of common material types is shown in Figure 2.1, but others are possible. The values are to be taken from the Systems Biology Ontology (`http://www.ebi.ac.uk/sbo/`), specifically from the branch having identifier `SBO:0000240` (*material entity* under *participant→physical participant*). The labels are defined by SBGN Entity Relationship Level 1.

Name	Label	SBO term
Non-macromolecular ion	`mt:ion`	SBO:0000327
Non-macromolecular radical	`mt:rad`	SBO:0000328
Ribonucleic acid	`mt:rna`	SBO:0000250
Deoxribonucleic acid	`mt:dna`	SBO:0000251
Protein	`mt:prot`	SBO:0000297
Polysaccharide	`mt:psac`	SBO:0000249

Figure 2.1: *A sample of values from the* material types *controlled vocabulary (Section 2.2.1).*

The material types are in contrast to the *conceptual types* (see below). The distinction is that material types are about physical composition, while conceptual types are about roles. For example, a strand of RNA is a physical artifact, but its use as messenger RNA is a role.

2.2.2 Entity Node conceptual types

An EPN's *conceptual type* indicates its function within the context of a given Process Diagram. A list of common conceptual types is shown in Figure 2.2, but others are possible. The values are to be taken from the Systems Biology Ontology (`http://www.ebi.ac.uk/sbo/`), specifically from the branch having identifier `SBO:0000241` (*conceptual entity* under *participant→physical participant*). The labels are defined by SBGN Entity Relationship Level 1.

Name	Label	SBO term
Gene	`ct:gene`	SBO:0000243
Transcription start site	`ct:tss`	SBO:0000329
Gene coding region	`ct:coding`	SBO:0000335
Gene regulatory region	`ct:grr`	SBO:0000369
Messenger RNA	`ct:mRNA`	SBO:0000278

Figure 2.2: *A sample of values from the* conceptual types *vocabulary (Section 2.2.2).*

2.2.3 Macromolecule covalent modifications

A common reason for the introduction of state variables on an entity is to allow access to the configuration of possible covalent modification sites on that entity. For instance, a macromolecule may have one or more sites where a phosphate group many be attached; this change in the site's configuration (i.e., being either phosphorylated or not) may factor into whether, and how, the entity can participate in different processes. Being able to describe such modifications in a consistent fashion is the motivation for the existence of SBGN's covalent modifications controlled vocabulary.

Figure 2.3 on the following page lists a number of common types of covalent modifications. The most common values are defined by the Systems Biology Ontology in the branch having identifier `SBO:0000210` (*addition* under *events→reaction→biochemical reaction→conversion→addition*). The labels shown in Figure 2.3 are defined by SBGN Entity Relationship Level 1; for all other kinds of modifications not listed here, the author of a Process Diagram must create a new label (and should also describe the meaning of the label in a legend or text accompanying the diagram).

Name	Label	SBO term
Acetylation	`Ac`	SBO:0000215
Glycosylation	`G`	SBO:0000217
Hydroxylation	`OH`	SBO:0000233
Methylation	`Me`	SBO:0000214
Myristoylation	`My`	SBO:0000219
Palmytoylation	`Pa`	SBO:0000218
Phosphorylation	`P`	SBO:0000216
Prenylation	`Pr`	SBO:0000221
Protonation	`H`	SBO:0000212
Sulfation	`S`	SBO:0000220
Ubiquitination	`Ub`	SBO:0000224

Figure 2.3: *A sample of values from the* covalent modifications *vocabulary (Section 2.2.3).*

2.2.4 Physical characteristics of compartments

SBGN Entity Relationship Level 1 defines a special unit of information for describing certain common physical characteristics of compartments. Figure 2.4 lists the particular values defined by SBGN Entity Relationship Level 1. The values correspond to the Systems Biology Ontology branch with identifier `SBO:0000255` (*physical characteristic* under *quantitative parameter*).

Name	Label	SBO term
Temperature	`pc:T`	SBO:0000147
Voltage	`pc:V`	SBO:0000259
pH	`pc:pH`	SBO:0000304

Figure 2.4: *A sample of values from the* physical characteristics *vocabulary (Section 2.2.4).*

2.3 Entity nodes

SBGN Entity Relationship Level 1 contains four glyphs representing classes of material entities: *unspecified entity*, *simple chemical*, *macromolecule* and *genetic entity*. (Specific types of macromolecules, such as protein, RNA, DNA, polysaccharide, and specific simple chemicals are not defined by SBGN Entity Relationship Level 1 but may be part of future levels of SBGN.) In addition to the material entities, SBGN Entity Relationship Level 1 represents five conceptual entities: *source/sink*, *perturbation*, *observable*, *tag* and *outcome*. Material and conceptual entities can optionally carry auxiliary units such as *units of information* and *state variables*.

2.3.1 Glyph: *Unspecified entity*

The simplest type of EN is the *unspecified entity*: one whose type is unknown or simply not relevant to the purposes of the model. 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 the model, without direct biological relevance. These are examples of situations where the *unspecified entity* glyph is appropriate. (Conversely, for cases where the identity of the entities *is* known, there exist other, more specific glyphs described elsewhere in the SBGN Entity Relationship Level 1 specification.)

SBO Term:

SBO:0000285 ! material entity of unknown nature

Container:

An *unspecified entity* is represented by an elliptic container, as shown in Figure 2.5.

Label:

An *unspecified entity* is identified by a label placed in an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the container. The label may spill outside of the container.

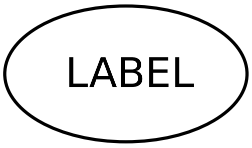

Figure 2.5: *The Entity Relationship glyph for* unspecified entity*.*

2.3.2 Glyph: *Simple chemical*

A simple chemical in SBGN Entity Relationship is defined as the opposite of a macromolecule (Section 2.3.3): 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.

SBO Term:

SBO:0000247 ! simple chemical

Container:

A *simple chemical* is represented by a circular container, as depicted in Figure 2.6 on the following page.

Label:

The identification of the *simple chemical* is carried by an unbordered box containing a string of characters. The characters may be distributed on several lines to improve readability, although this is not mandatory. The label box has to be attached to the center of the circular container. The label is permitted to spill outside the container.

Auxiliary items:

A *simple chemical* may be decorated with one or more *units of information* (Section 2.3.10). A particular *unit of information* describes the material type.

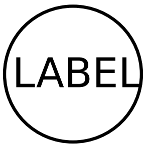

Figure 2.6: *The Entity Relationship glyph for* simple chemical.

2.3.3 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 Entity Relationship 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 diagram is then clarified using its label and decorations, as will become clear below. (Future levels of SBGN may subclass the *macromolecule* and introduce different glyphs to differentiate macromolecules.)

SBO Term:

SBO:0000245 ! macromolecule

Container:

A macromolecule is represented by a rectangular container with rounded corners, as illustrated in Figure 2.7.

Label:

A *macromolecule* is identified by a label placed in an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the container. The label may spill outside of the container.

Auxiliary items:

A *macromolecule* can carry state variables that can add information about its state (Section 2.3.11). The state of a macromolecule is therefore defined as the vector of all its state variable values. A state variable is represented by an ellipsoid container, with the long axis of the ellipsoid placed on the border of the *macromolecule*'s container as illustrated in Figure 2.7. The label of the state variable (which can precise the type of characteristic represented by the state variable, residue type, residue number etc.) is written within the state variable's container.

A *macromolecule* can also carry one or several *units of information* (Section 2.3.10). The units of information can characterise a domain, such as a binding site. Particular *units of information* are available for describing the material type (Section 2.2.1) and the conceptual type (Section 2.2.2) of a macromolecule. The center of the bounding box of a *unit of information* is located on the mid-line of the border of the macromolecule.

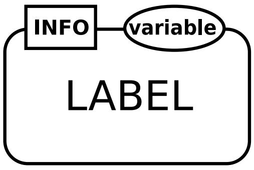

Figure 2.7: *The Entity Relationship glyph for* macromolecule.

2.3.4 Glyph: *Genetic Entity*

The *genetic entity* 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 transcript. The label of this EN and its *units of information* are often important for making the purpose clear to the reader of a diagram.

SBO Term:

SBO:0000354 ! genetic entity

Container:

A *genetic entity* is represented by a rectangular container whose bottom half has rounded corners, as shown in Figure 2.8.

Label:

The identity of a particular *genetic entity* is established by a label placed in an unordered box containing a string of characters. The characters may be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the container. The label may spill outside of the container.

Auxiliary items:

A *genetic entity* can carry state variables (Section 2.3.11) that add information about its precise state. The state of a genetic entity is therefore defined as the vector of all its state variables. A state variable is represented by an ellipsoid container, with the long axis of the ellipsoid placed on the border of the *genetic entity*'s container as illustrated in Figure 2.8. The label of the state variable (type of characteristic, nucleotide number) is written within the variable's container itself.

A *genetic information* can also carry one or several *units of information* (Section 2.3.10). These can characterise a *genetic entity*'s domain, such as a binding site, or an exon. Particular *units of information* carry the material type (Section 2.2.1) and the conceptual type (Section 2.2.2) of the of the genetic entity. The center of the bounding box of a *unit of information* is located on the mid-line of the border of the *genetic entity*.

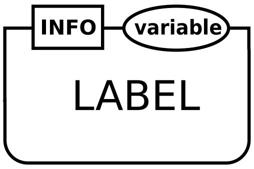

Figure 2.8: *The Process Diagram glyph for* genetic entity.

2.3.5 Glyph: *Source/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 symbol for explicitly representing the involvement of an unspecified source or sink. 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 diagram.

A frequently asked question is, why bother having an explicit symbol at all? The reason is that one cannot simply use an arc that does not terminate on a node, because the dangling end could be mistaken to be pointing to another node in the diagram. This is specially true if the diagram is rescaled, causing the spacing of elements in the diagram to change, and in the case of automatic layout. The availability and use of an explicit symbol for sources and sinks is therefore critical.

SBO Term:

SBO:0000291 ! empty set

Container:

A *source/sink* is represented by a glyph 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. Figure 2.9 illustrates this. The symbol should only be linked to one and only one edge in a diagram.

Label:

An *source/sink* does not carry any labels.

Auxiliary items:

An *source/sink* does not carry any auxiliary items.

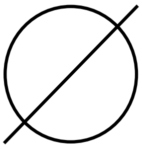

Figure 2.9: *The Entity Relationship glyph for* source/sink.

2.3.6 Glyph: *Perturbation*

Biochemical networks can be affected by external influences. Those influences can be well-defined physical perturbations, such as a light pulse or a change in temperature; they can also be more complex and not well-defined phenomena, for instance a biological process, an experimental setup, or a mutation. For these situations, SBGN provides the *perturbation* glyph.

SBO Term:

SBO:0000357 ! perturbation

Container:

A *perturbation* is represented by a modified hexagon having two opposite concave faces, as illustrated in Figure 2.10 on the following page.

Label:

A *perturbation* is identified by a label placed in an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the *perturbation* container. The label may spill outside of the container.

Auxiliary items:

A *perturbation* may carry a *clone marker* (Section ??).

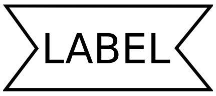

Figure 2.10: *The Entity Relationship glyph for* perturbation.

2.3.7 Glyph: *Observable*

A biochemical network can generate phenotypes or affect biological processes. Such processes can take place at different levels and are independent of the biochemical network itself. To represent these processes in a diagram, SBGN defines the *observable* glyph.

SBO Term:

SBO:0000358 ! observable

Container:

An *observable* is represented by an elongated hexagon, as illustrated in Figure 2.11.

Label:

An *observable* is identified by a label placed in an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the *observable* container. The label may spill outside of the container.

Auxiliary items:

An *observable* may carry a *clone marker* (Section ??).

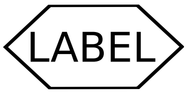

Figure 2.11: *The Entity Relationship glyph for* observable.

2.3.8 Glyph: *Tag*

A *tag* is a named handle, or reference, to another EN (Section 2.3) or a container node (Section ??). *Tags* can be used to identify elements in SBGN *submaps* (Section 2.4).

SBO Term:

Not applicable.

Container:

A *tag* is represented by a rectangle fused to an empty arrowhead, as illustrated in Figure 2.12 on the following page. The symbol should only be linked to one and only one edge (i.e., it should reference only one EN or container).

Label:

A *tag* is identified by a label placed in an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the container. The label may spill outside of the container.

Auxiliary items:

A *tag* does not carry any auxiliary items.

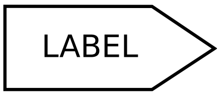

Figure 2.12: *The Entity Relationship glyph for* tag.

2.3.9 *Outcome*

2.3.9.1 Introduction

In Entity Relationship, and *outcome* represents the result of an *interaction* (section **??**) or an *assignment* (**??**). For instance, if an *interaction* represents a non-covalent binding, the *outcome* represents the complex. If an *interaction* represents a genetic interaction, for instance derived from genetic screenings, the *outcome* represents the result of the presence of the two polymorphisms. If an *assignment* represents the phosphorylation of a protein, the *outcome* represents the phosphorylated form of this protein.

SBO Term:

SBO:New ! to be determined

Container:

An *outcome* is represented by a black dot located on the arc of an *interaction* (see section **??**) or an *assignment* (see section **??**). The diameter of the dot has to be larger than the thickness of the arc.

Label:

An *outcome* has no identity on its own and does not carry any label.

Auxiliary items:

An *outcome* does not carry any auxiliary items.

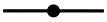

Figure 2.13: *Examples of the Entity Relationship glyph for* outcome.

2.3.10 Glyph: *Unit of information*

When representing biological entities, it is often necessary to convey some abstract information about the entity's function that cannot (or does not need to) be easily related to its structure. The SBGN *unit of information* is a decoration that can be used in this situation to add information to a glyph. 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.

SBO Term:

Not applicable.

Container:

A unit of information is represented by a rectangle. The long side of the rectangle should be oriented parallel to the border of the *EN* being annotated by the *unit of information*. The center of the bounding box of a *state of information* should be located on the mid-line of the border of the *EN*.

Label:

A *unit of information* is identified by a label placed in an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the container. The label may spill outside of the container.

The label defines the information carried by the *unit of information*. 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 Diagram Level 1 are described in Section 2.2 and summarized in the following list:

`pc`	container physical characteristic
`mt`	entity material type
`ct`	entity conceptual type

Auxiliary items:

A *unit of information* does not carry any auxiliary items.

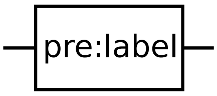

Figure 2.14: *The Entity Relationship glyph for* unit of information.

2.3.11 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.

In SBGN Entity Relationship Level 1, *state variables* are also used to describe the localisation in compartments (a transport is therefore described as a state variable assignment, see Section **??**).

SBO Term:

Not applicable.

Container:

A *state variable* is represented by an ”sausage” container, that is two hemicercles of same radius joined by parellel segments, as shown in Figure 2.15. The parallel segament axis should be tangent to the border of the glyph of the *EN* being modified by the *state variable*. The center of the bounding box of a *state variable* should be located on the mid-line of the border of the *EN*.

Label:

An *unspecified entity* is identified by a label placed in an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the container. The label may spill outside of the container.

Auxiliary items:

A *state variable* does not carry any auxiliary items.

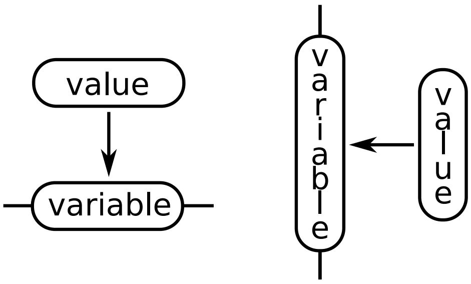

Figure 2.15: *Examples of the Process Diagram glyph for* state variable.

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.

The state variable is assigned state-values (see Section **??**). Those values are contained in a glyph similar to the *stateVariable*, although not carried by another *EN*.

2.3.12 Examples of complex ENs

In this section, we provide examples of Entity Node representations drawn using the SBGN Entity Relationship Level 1 glyphs described above. The first is a representation of the calcium/calmodulin kinase II, with the phosphorylation sites threonine 286 and 306, as well as catalytic and autoinhibitory domains. This is shown in Figure 2.16 on the following page. Note the use of *units of information* and *state variables*.

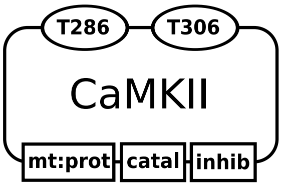

Figure 2.16: *An example representation of calcium/calmodulin kinase II.*

The next EN example is a representation of the glutamate receptor in with a pore, and with both phosphorylation and glycosylation. The entity carries two functional domains, the ligand-binding domain and the ion pore. Figure 2.17 gives the diagram.

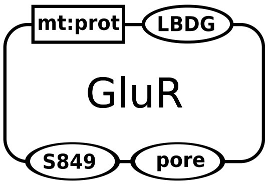

Figure 2.17: *An example of a glutamate receptor in the open state.*

2.4 Glyph: *Submap*

A *submap* is used to encapsulate processes and relationships (including all types of nodes and edges) within one glyph. The submap hides its content to the users, and display only input terminals (or ports), linked to *ENs* (Section 2.3). A submap is not equivalent to an omitted transition (see Section 2.5.2). In the case of an SBGN diagram that is made available through a software tool, the content of a submap may be available to the tool. A user could then ask the tool to expand the submap, for instance by clicking on the icon for the submap. The tool might then expand and show the submap within the same diagram (on the same canvas), or it might open it in a different canvas.

SBO Term:

To be determined.

Container:

The *submap* is represented as a square box.

Label:

The identification of the *submap* is carried by an unbordered box containing a string of characters. The characters may be distributed on several lines to improve readability, although this is not mandatory. The label box has to be attached to the center of the container box.

Auxiliary items:

A *submap* carries labeled terminals. When the submap is represented folded, those terminals are linked to external *ENs* (Section 2.3). In the unfolded view, exposing the internal structure of the submap, a set of *tags* point to the corresponding internal *ENs*.

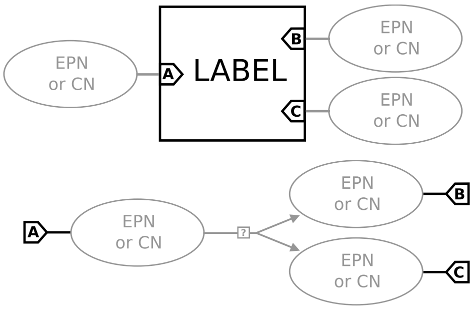

Figure 2.18: *The Entity Relationship glyph for* submap*. (Upper part) folded submap. (Lower part) content of the submap.*

Figure 2.19 represents a *submap* that transforms glucose into fructose-6-phosphate. The *submap* carries four terminals to ENs. Note that the terminals do not define a "direction", such as input or output. The flux of the reactions is determined by the context.

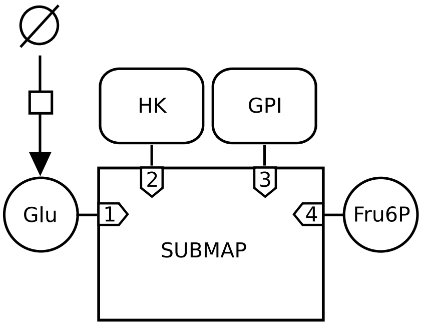

Figure 2.19: *Example of a submap with contents elided.*

The diagram in Figure 2.20 represents an unfolded version of a submap. Here, anything outside the submap has disappeared, and the internal *tags* are not linked to the corresponding external *terminals*.

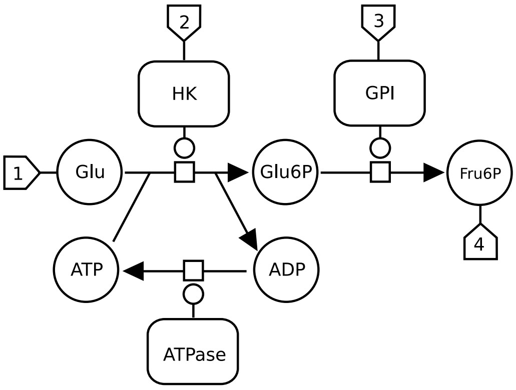

Figure 2.20: *Example of an unfolded submap. The unfolded submap corresponds to the folded submap of Figure 2.19.*

2.5 Process nodes

Process nodes represent processes that transform one or several ENs into one or several different ENs.

2.5.1 Glyph: *Transition*

A transition is a process transforming a set of entities (represented by *ENs* in SBGN Process Diagram Level 1) into another set entities.

SBO Term:

SBO:0000167 ! reaction

Origin:

One or several *consumption* arcs (Section **??**) or one or several *production* arcs (Section **??**).

Target:

One or several *production* arcs (Section **??**).

Node:

A transition is represented by a square box linked to two connectors, small arcs attached to the centers of opposite sides. The consumption (Section **??**) and production (Section **??**) arcs are linked to the extremities of those connectors. The modulatory arcs (Section 2.6) point to the other two sides of the box. A *transition* connected to *production* arcs on opposite sides is a reversible transition. The connectors and the box move as a rigid entity.

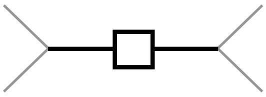

Figure 2.21: *The Entity Relationship glyph for* transition.

A transition is the basic process node in SBGN. It describes a process that transforms a given set of biochemical entities—macromolecules, simple chemicals or unspecified entities—into another set of biochemical entities. Such a transformation might imply modification of covalent bonds (conversion), modification of the relative position of constituents (conformational transition) or movement from one compartment to another (translocation).

A cardinality label may be associated with *consumption* (Section **??**) or *production* (Section **??**) arcs to indicate the stoichiometry of the process. This label becomes a requirement when the exact composition of the number of copies of the inputs or outputs to a reaction are ambiguous in the diagram.

The example in Figure 2.22 on the following page illustrates the use of a *transition* node to represent a reaction between two reactants that generates three products.

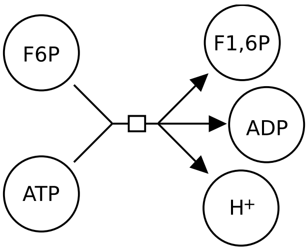

Figure 2.22: *A reaction that generates three products.*

2.5.2 Glyph: *Omitted process*

Omitted processes are processes that are known to exist, but are omitted from the diagram for the sake of clarity or parsimony. A single *omitted process* can represent any number of actual processes. The *omitted process* is different from a *submap*. While a *submap* possess an explicit content that is hidden in the main map, the *omitted process* does not "hide" anything within the context of the diagram, and cannot be "unfolded".

SBO Term:
To be determined.

Origin:
One or several *consumption* arcs (Section **??**) or one or several *production* arcs (Section **??**).

Target:
One or several *production* arcs (Section **??**).

Node:
Omitted processes are represented as a transition in which the square box contains a two parallel slanted lines oriented northwest-to-southeast and separated by an empty space.

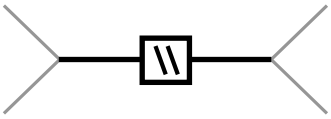

Figure 2.23: *The Entity Relationship glyph for* omitted.

2.5.3 Glyph: *Uncertain process*

Uncertain processes are processes that may not exist. A single *uncertain process* can represent any number of actual processes.

SBO Term:
To be determined.

Origin:
One or several *consumption* arcs (Section **??**) or one or several *production* arcs (Section **??**).

Target:
One or several *production* arcs (Section **??**).

Node:

Uncertain processes are represented as a transition which square box contains a question mark.

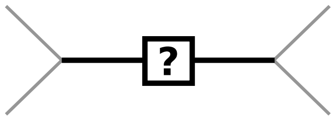

Figure 2.24: *The Entity Relationship glyph for an* uncertain process.

2.6 Connecting arcs

Connecting arcs are lines that link EPNs and PNs together. The symbols attached to their extremities precise their semantics.

Bibliography

[1] Kurt W. Kohn. Molecular interaction map of the mammalian cell cycle control and dna repair systems. *Molecular Biology of the Cell*, 10(8):2703–2734, 1999.

[2] Peter Pin-Shan S. Chen. The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.

[3] I. Pirson, N. Fortemaison, C. Jacobs, S. Dremier, J. E. Dumont, and C. Maenhaut. The visual display of regulatory information and networks. *Trends in Cell Biology*, 10(10):404–408, 2000.

[4] Daniel L. Cook, J. F. Farley, and S. J. Tapscott. A basis for a visual language for describing, archiving and analyzing functional models of complex biological systems. *Genome Biology*, 2(4):research0012.1–research0012.10., 2001.

[5] Ron Maimon and Sam Browning. Diagrammatic notation and computational structure of gene networks. In Hiroaki Kitano, editor, *Proceedings of the 2nd International Conference on Systems Biology*, pages 311–317, Madison, WI, 2001. Omnipress.

[6] Hiroaki Kitano. A graphical notation for biochemical networks. *BioSilico*, 1:169–176, 2003.

[7] Hiroaki Kitano, Akira Funahashi, Yukiko Matsuoka, and Kanae Oda. Using process diagrams for the graphical representation of biological networks. *Nature Biotechnology*, 23(8):961–966, 2005.

[8] Stuart L. Moodie, Anatoly A. Sorokin, Igor Goryanin, and Peter Ghazal. A graphical notation to describe the logical interactions of biological pathways. *Journal of Integrative Bioinformatics*, 3:36, 2006.

[9] N. Le Novère, S. Moodie, A. Sorokin, M. Hucka, Schreiber F., E. Demir, H. Mi, Y. Matsuoka, K. Wegner, and Kitano H. Systems biology graphical notation: Process diagram level 1. Technical report, 2008.

[10] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novère, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.