Systems Biology Graphical Notation: Activity Flow Diagram Level 1

Draft

Date: 05 March, 2009

Editors:

Huaiyu Mi	*SRI International, USA*
Falk Schreiber	*IPK Gatersleben & MLU Halle, Germany*
Nicolas Le Novère	*EMBL European Bioinformatics Institute, UK*
Stuart Moodie	*CSBE, University of Edinburgh, UK*
Anatoly Sorokin	*University of Edinburgh, UK*[7pt]

Principal Authors:

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. *Bug reports and specific comments about the specification should be entered in the issue tracker* http://sourceforge.net/tracker/?group_id=178553&atid=1082245.

Preface

Acknowledgements

The authors are grateful to all the attendees of the SBGN meetings, as well as to the subscribers of the sbgn-discuss@sbgn.org mailing list.

The development of SBGN was mainly supported by a grant from the Japanese *New Energy and Industrial Technology Development Organization* (NEDO, http://www.nedo.go.jp/). The *Okinawa Institute of Science and Technology* (OIST, http://www.oist.jp/), the *AIST Computational Biology Research Center* (AIST CBRC, http://www.cbrc.jp/index.eng.html) the British *Biotechnology and Biological Sciences Research Council* (BBSRC, http://www.bbsrc.ac.uk/) through a Japan Partnering Award, the European Media Laboratory (EML Research gGmbH, http://www.eml-r.org/), and the Beckman Institute at the California Institute of Technology (http://bnmc.caltech.edu) provided additional support for SBGN workshops. Some help was provided by the *Japan Science and Technology Agency* (JST, http://www.jst.go.jp/) and the *Genome Network Project* of the Japanese Ministry of Education, Sports, Culture, Science, and Technology (MEXT, http://www.mext.go.jp/) for the development of the gene regulation network aspect of SBGN, and from the *Engineering and Physical Sciences Research Council* (EPSRC, http://www.epsrc.ac.uk) during the redaction of the specification.

Notes on typographical conventions

The concept represented by a glyph is written using a normal font, while a *glyph* means the SBGN visual representation of the concept.

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 *Activity Flow* visual language of SBGN. As explained more fully in Section 1.2, Activity Flow 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 Activity Flow 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 [1], 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 [2]. MIM is essentially a variation of the entity-relationship diagrams [3]. 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 [4], BioD [5], Patika [6, 7], and others), as well as extensions of Kohn's notation (e.g., the Diagrammatic Cell Language of Maimon and Browning [8]).

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 [9]. The Process Diagram notation was the first to be fully defined using a well-delineated set of symbols and syntax [10]. 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 [11] or Patika [6, 7]. 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 meeting, attendees also held the first in-depth discussions about the syntax of the Entity Relationship language. The specification for the Activity Flow language was initially discussed during a meeting held in Rostock, Germany, in October, 2008. 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; [12]): 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

Activity Flow Diagram glyphs

This chapter provides a catalog of the graphical symbols available for representing entities in Activity Flow diagrams. In Chapter 3 beginning on page 15, we describe the rules for combining these glyphs into a legal SBGN Activity Flow, and in Chapter 4 beginning on page 18, 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 Activity Flow notation with the help of an example shown in Figure 2.1.

Figure 2.1:

2.2 Controlled vocabularies used in SBGN Activity Flow Level 1

What controlled vocabulary should we use to describe the activity? SBO? GO Molecular function?

Some glyphs in SBGN Activity Flow can contain particular kinds of textual annotations conveying information relevant to the purpose of the glyph. These annotations are *units of information* (Section 2.3.8). An example is in the case of multimers, which can have a unit of information conveying the number of monomers composing the multimer. Other cases are described throughout the rest of this chapter.

2.2.1 Activity node material types

The material type of an AFN indicates its chemical structure. A list of common material types is shown in Table 2.1 on the following page, 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 Activity Flow 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

Table 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 Activity node conceptual types

An AFN's *conceptual type* indicates its function within the context of a given Activity Flow. A list of common conceptual types is shown in Table 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 Activity Flow 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

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

2.2.3 Physical characteristics of compartments

SBGN Activity Flow Level 1 defines a special unit of information for describing certain common physical characteristics of compartments. Table 2.3 lists the particular values defined by SBGN Activity Flow 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

Table 2.3: *A sample of values from the* physical characteristics *vocabulary (Section 2.2.3).*

2.2.4 Cardinality

SBGN Activity Flow Level 1 defines a special unit of information usable on multimers for describing the number of monomers composing the multimer. Table 2.4 shows the way in which the values must be written. Note that the value is a unitary number, and not (for example) a range. There is no provision in SBGN Activity Flow Level 1 for specifying a range in this context because it leads to problems of entity identifiability.

Name	Label	SBO term
cardinality	`N:#`	`SBO:0000364`

Table 2.4: *The format of the possible values for the* cardinality *unit of information (Section 2.2.4). Here, # stands for the number; for example, "`N:5`".*

2.3 Activity nodes

An activity node (AN) represents the activity of an entity or an entity pool, but not the entities themselves. For instance, multiple activity nodes can be used to represent different activities of a particular entity, while one activity node can be used to represent the activity of a complex multimer.

How about activities in different compartment? Is activity compartment specific? If so, how do we show the activity across the compartment boundary? Transportation?

SBGN Activity Flow Level 1 uses one glyph to represent activities from all biological molecules. *unspecified activity*, *simple chemical activity*, *macromolecule activity*, *nucleic acid feature activity*, and *complex activity*. Activities of specific types of macromolecules, such as protein, RNA, DNA, polysaccharide, and specific simple chemicals are not defined by SBGN Activity Flow Level 1 but may be part of future levels of SBGN. In addition to activities of material entities, SBGN Activity Flow Level 1 represents activity from two conceptual entities: *perturbation*, *observable*. Auxiliary units such as units of information, state variables are not shown on the activity nodes. Each activity is displayed only once in one compartment.

2.3.1 Glyph: *Unspecified activity*

The simplest type of AN is the *unspecified activity*: one whose type is unknown or simply not relevant to the purposes of the model. This arises, for example, when the nature of the activity is unknown, either to a known entity or an entity that has been inferred indirectly, or when the entity is merely a construct introduced for the needs of the model, without direct biological relevance.

SBO Term:

SBO:

Container:

An *unspecified activity* is represented by an elliptic container, as shown in Figure 2.2 on the following page.

Label:

An *unspecified actifity* 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 2.2: *The Activity Flow glyph for* unspecified activity.

2.3.2 Glyph: *Simple chemical activity*

A simple chemical activity in SBGN Actifity Flow is defined as the activity from a chemical compound that is not formed by the covalent linking of pseudo-identical residue, as opposite of a macromolecule activity (Section 2.3.3). Examples of simple chemicals are an atom, a monoatomic ion, a salt, a radical, a solid metal, a crystal, etc.

SBO Term:
SBO:

Container:
A *simple chemical activity* is represented by a circular container, as depicted in Figure 2.3.

Label:
The identification of the *simple chemical activity* 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 activity* may be decorated with one or more *units of information* (Section 2.3.8).

Figure 2.3: *The Activity Flow glyph for* simple chemical activity.

2.3.3 Glyph: *Macromolecule activity*

The *macromolecule activity* is defined, as the name implies, as the activities of macromolecules, which are 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 the activities of each of these different molecules would lead to an explosion of symbols in SBGN, so instead, SBGN Activity Flow Level 1 defines only one glyph for activity of all macromolecules. The same glyph is to be used for activity of a protein or peptide, a nucleic acid, a complex sugar, and so on. The exact nature of a particular macromolecule that the activity is coming from in a diagram is then clarified using its label and annotation. (Future levels of SBGN may subclass the macromolecule and introduce different glyphs to differentiate macromolecules.)

SBO Term:
SBO:

Container:

A *macromolecule activity* is represented by a rectangular container with rounded corners, as illustrated in Figure 2.4.

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.

A *macromolecule* can also carry one or several *units of information* (Section 2.3.8). The units of information can characterize 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.

Figure 2.4: *The Activity Flow glyph for* macromolecule activity.

2.3.4 Glyph: *Nucleic acid feature activity*

In SBGN, the nucleic acid feature construct 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. Therefore, the *nucleic acid feature activity* is used to indicate the activity derived from the genetic information.

SBO Term:

SBO:

Container:

A *nucleic acid feature activity* is represented by a rectangular container whose bottom half has rounded corners, as shown in Figure 2.5. This design reminds that we are fundamentally dealing with a unit of information, but this information is carried by a macromolecule.

Label:

The identity of a particular *Nucleic acid feature activity* 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.

A *nucleic acid feature* can also carry one or several *units of information* (Section 2.3.8).

Figure 2.5: *The Activity Flow glyph for* nucleic acid feature activity.

2.3.5 Glyph: *Complex activity*

A *complex activity* node represents the activity from a biochemical entity composed of other biochemical entities, whether macromolecules, simple chemicals, multimers, or other complexes. For example, a heterotetramer of a voltage-gated potassium channel has a channel pore formed by four different subunits. On the other hand, if the activity is known to come from a particular component of the complex, the macromolecule activity node should be used.

SBO Term:
SBO:

Container:
A *complex activity* is represented by a rectangle with cut-corners (an octagonal box with sides of two different lengths). Individual subunits or component of the complex should not be represented.

Label:
The identification of a named *complex activity* 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 midway between the border of the complex's container box and the border of the components' container boxes.

Auxiliary items:
A *complex activity* can also carry one or several *units of information* (see Section 2.3.8).

Figure 2.6: *An example Activity Flow glyph for* complex activity.

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 2.7.

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.

Figure 2.7: *The Activity Flow 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 2.8.

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.

Figure 2.8: *The Activity Flow glyph for* observable.

2.3.8 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 *AFN* 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 *AFN*.

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 Activity Flow Level 1 are described in Section **??** and summarized in the following list:

`pc` container physical characteristic
`mt` activity material type
`ct` activity conceptual type
`N` multimer cardinality

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

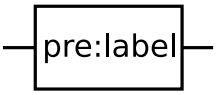

Figure 2.9: *The Activity Flow glyph for* unit of information.

2.4 Container nodes

Containers are SBGN constructions that contain one or several other SBGN constructs. There are two container nodes in SBGN Activity Flow Level 1: *compartment* and *submap*.

2.4.1 Glyph: *Compartment*

In order to describe biochemical and cellular events, it is useful to define the notion of pools. A pool is an ensemble of participants that can be considered to be identical for the events in which they are involved. A compartment is a logical or physical structure that contains pools. A pool can only belong to one compartment. Therefore, the "same" biochemical species located in two different compartments are in fact two different pools.

SBO Term:
SBO:0000289 ! functional compartment

Container:
A compartment is represented by a surface enclosed in a continuous border or located between continuous borders. These borders should be noticeably thicker than the borders of the AFNs. A compartment can take **any** geometry. A compartment must always be entirely enclosed.

Label:
The identification of the compartment is carried by 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 can be attached anywhere in the container box. Note that the label can spill-over from the container box.

Auxiliary items:
A *compartment* can carry a certain number of *units of information*, that will add information for instance about the physical environment, such as pH, temperature or voltage, see Section 2.3.8. The center of the bounding box of a *unit of information* is located on the mid-line of the border of the compartment.

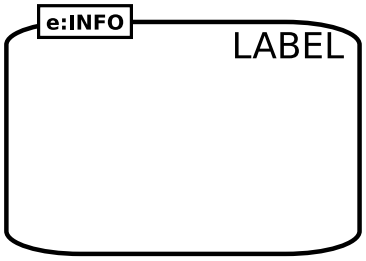

Figure 2.10: *The Activity Flow glyph for* compartment.

2.4.2 Glyph: *Submap*

SBO Term:

To be determined.

Container:

The *submap* is represented as a square box to remind the viewer that it is fundamentally a process.

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.

2.5 Connecting arcs

2.5.1 Glyph: *Positive influence*

2.5.2 Glyph: *Negative influence*

2.5.3 Glyph: *Unknown influence*

2.5.4 Glyph: *Trigger*

2.6 Logical operators

2.6.1 Glyph: *And*

The glyph *and* is used to denote that all the *AFNs* linked as input are necessary to produce the output.

SBO Term:

SBO:0000173 ! and.

Origin:

More than one AFN (section **??**) or logical operator (section 2.6).

Target:

Node:

And is represented by a circle carrying the word “AND”.

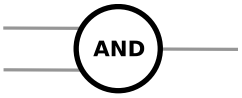

Figure 2.11: *The Activity Flow glyph for* and. *Only two inputs are represented, but more would be allowed.*

2.6.2 Glyph: *Or*

The glyph *or* is used to denote that any of the *AFNs* linked as input is sufficient to produce the output.

SBO Term:
SBO:0000174 ! or.

Origin:
More than one AFN (section **??**) or logical operator (section 2.6).

Target:

Node:
Or is represented by a circle carrying the word "OR".

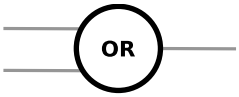

Figure 2.12: *The Activity Flow glyph for* or. *Only two inputs are represented, but more would be allowed.*

2.6.3 Glyph: *Not*

The glyph *not* is used to denote that the *AFN* linked as input cannot produce the output.

SBO Term:
SBO:0000238 ! not.

Origin:
One AFN (section **??**) or logical operator (section 2.6).

Target:

Node:
Not is represented by a circle carrying the word "NOT".

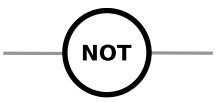

Figure 2.13: *The Activity Flow glyph for* not.

Chapter 3

Process Diagram grammar

3.1 Overview

3.2 Concepts

3.3 The conceptual model

3.4 Syntax

3.5 Semantic rules

3.5.1 Namespaces

The notation has a concept of a namespace within which entities with the same identifying attributes are regarded as identical. The SBGN namespaces are shown in table 3.1.

Table 3.1: *Namespace scope definitions.*

Namespace Scope	Entities affected	Notes
MapDiagram	CompartmentNode, SubMapDiagram, EquivalenceNode	
CompartmentShape	BasicEntityNode	If no *compartment* is drawn then all BasicEntityNodes are assumed to belong to an invisible "default" compartment.
EntityType	StateVariable, Annotation	
ComplexType	EntityType	

3.5.2 Compartment spanning

In all cases an AFN cannot *belong* to more than one compartment. However, an AFN can be *drawn* over more than one compartment. In such cases the decision on which is the owning compartment is deferred to the drawing tool or the author. ComplexNodes may contain AFNs which belong to different compartments and in this way a complex can be used to describe an entity that spans more than one compartment.

This restriction makes it impossible to represent in a semantically correct way a macromolecule that spans more then one compartment — for example a receptor protein. It is clearly desirable to be able to show a macromolecule in a manner that the biologist expects (i.e., spanning from the outside through the membrane to the inside). Therefore, the author is recommended to draw the macromolecule across compartment boundaries, but the underlying SBGN semantic model will assign it to only one. The assignment to a compartment may be decided by the software drawing tool or the author. Note that this has implications for auto-layout algorithms as they will only be able to treat such entity nodes as contained within a compartment and will have no way of knowing a macromolecule spans a compartment.

The current solution is consistent with other Systems Biology representations such as SBML and BioPAX. For more information about the problems representing membrane spanning proteins and the rationale behind the current solution see Section **??**.

3.5.3 Compartments

The layout of compartments in an SBGN diagram does not imply anything about the topology of compartments in the cell. Compartments should be bounded and may overlap. However, adjacency and the nesting of compartments does not imply that these compartments are next to each other physically or hat one compartment contains the other.

3.5.4 Modulation

It is implied, but not defined explicitly that a process has a rate at which it converts its input AFNs to its output AFNs. This concept is important in understanding how SBGN describes process modulation.

1. A process with no modulations has an underlying "basal rate" which describes the rate at which it converts inputs to outputs.
2. Modulation changes the basal rate in an unspecified fashion.
3. Stimulation is a modulation that's effect is to increase the basal rate.
4. Inhibition is a modulation that's effect is to decrease the basal rate.
5. The above types of modulation, when assigned to the same process are combined and have a multiplicative effect on the basal rate of the process.
6. Modulators that do not interact with each other in the above manner should be drawn as modulating different process nodes. Their effect is therefore additive.
7. At most one trigger can be assigned to a process. Two triggers would imply an implicit Boolean AND or OR operator. For clarity only one trigger can be assigned to a process and such combinations must be explicitly expressed as the Boolean operators.
8. At most one catalysis can be assigned to a process. A catalysis modulation implies that the exact biochemical mechanism underlying the process is known. In this context two catalysis reactions cannot be assigned to the same process as they are independent reactions. Other AFNs that modulate the catalysis can be assigned to the same process as modulators, stimulators, and inhibitors and will have a multiplicative modulation on the reaction rate defined by the catalysis.

3.5.5 Submaps

Submaps are a visual device that allow a map to be split into several views. They remain, however, part of the main map and share its namespace. As a test of validity it should be possible to reintroduced a submap into the main map by eliminating the SubMapNode and merging the equivalent EntityPoolNodes in both maps.

3.5.5.1 Rules for mapping to submaps

An ActivityFlowNode in the main map can be mapped to one in the submap using a TagNode in the submap and SubMapTerminals (see Section 2.4.2) in the main map. For a mapping between map and submap to exists the following must be true:

1. The identifiers in the TagNode and SubMapTerminals must be identical.
2. The AFNs must be identical.

3.5.5.2 Requirement to define a mapping

If a map and submap both contain the same AFN, then a mapping between them must be defined as above.

Chapter 4

Layout Guidelines for a Activity Flow Diagram

4.1 Introduction

The previous chapters describe the appearance and meaning of SBGN Activity Flow Level 1 components. Objects are *AFNs*, *container nodes*, *logical operators* as well as *connecting arcs*. The components of a Activity Flow have to be placed in a meaningful way – a random distribution with spaghetti-like connections will most likely hide the information encoded in the underlying model, whereas an elegant placement of the objects, giving a congenial appearance of the diagrams, may reveal new insights. The arrangement of components in a diagram is called a *layout*.

SBGN Activity Flow diagrams should be easily recognisable not only by the glyphs used, but also by the general style of the layout. However, the arrangement of the components is a complex art in itself, and there is no simple rule which can be applied to all cases. Therefore this section provides guidelines for the layout of process diagrams, divided into two categories:

1. requirements, i.e., rules which **must** be fulfilled by a layout, and

2. recommendations, i.e., rules which **should** be followed if possible.

In addition, we provide a list of additional suggestions which may help in producing aesthetically more pleasant layouts, possibly easier to understand.

Those layout guidelines are independent of the method used to produce the diagram, and apply to both manually drawn diagrams as well as diagrams produced by an automatic layout algorithm. The guidelines do not deal with interactive aspects (e.g., the effect of zooming). Further information about automatic network layout (graph drawing) can be found, for example, in the books of Di Battista and co-authors [13] and Kaufmann and Wagner [14].

Please note that the color of objects do not carry any meaning in SBGN. Although one can use colors to emphasize part of a diagram or encode additional information, the meaning of the diagram should not depend on the colors. Furthermore, objects can have different sizes and size is also meaningless in SBGN. For example, a transition node may be larger than a protein node. Also the meaning of a graph should be conserved upon scaling as far as possible.

4.2 Layout guidelines

4.2.1 Requirements

Requirements are rules which **must** be fulfilled by a layout to produce a valid SBGN Activity Flow Level 1 graph.

4.2.1.1 Node-node overlaps

Nodes are only allowed to overlap in two cases:

1. the overlapping nodes define a glyph (e.g., a *multimer* composed by stacking of two containers representing the monomers).
2. nodes overlapping compartments (e.g., a complex placed on the compartment border).

Otherwise, nodes are not allowed to overlap (Figure 4.1). This includes the touching of nodes. Touching is not allowed apart from the case where it has a specific meaning, e.g., two macromolecules touching each other within a complex because they form the complex. Submaps are not allowed to overlap.

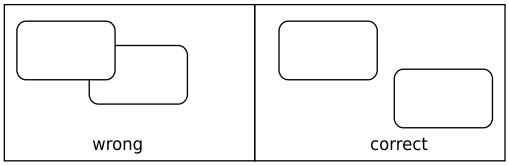

Figure 4.1: *Nodes must not overlap.*

4.2.1.2 Node-edge crossing

In case of node-edge crossing the edge must be drawn on the top of the node (Figure 4.2). See also recommendation 4.2.2.1 (crossing between edges and nodes should be avoided).

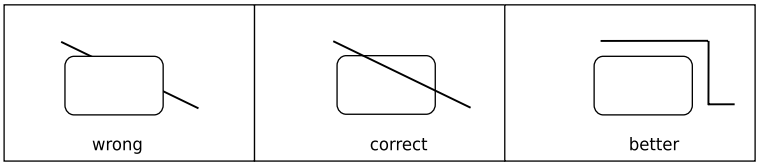

Figure 4.2: *If an edge crosses a node, the edge must be drawn on top of the node.*

4.2.1.3 Node border-edge overlaps

Edges are not allowed to overlap the border lines of nodes (Figure 4.3 on the following page).

4.2.1.4 Edge-edge overlaps

Edges are not allowed to overlap (Figure 4.4 on the next page). This includes touching of edges. Furthermore, an edge is neither allowed to cross itself nor to cross a boundary of node more than twice or other edges more than once.

4.2.1.5 Node orientation

Nodes have to be drawn horizontally or vertically, any other rotation of elements is not allowed (Figure 4.5 on the following page).

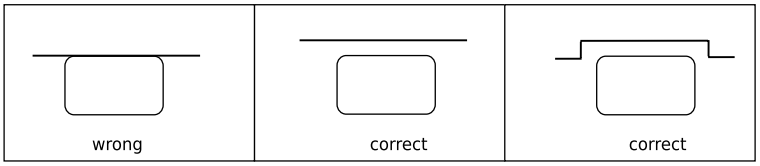

Figure 4.3: *Edges must not overlap node borders.*

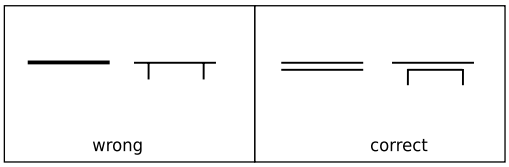

Figure 4.4: *Edges must not overlap.*

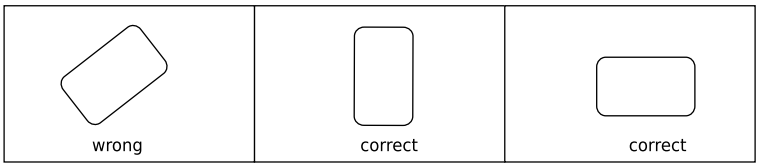

Figure 4.5: *The node orientation must be horizontally or vertically.*

4.2.1.6 Node-edge connection

The arcs linking the square glyph of a *transition* to the *consumption* and *production arcs* are attached to the center of opposite sides (Figure 4.6). The modulatory arcs are attached to the other two sides, but not necessarily all to the center, as several modifiers can affect the same *transition node*. A *transition* connected to *production arcs* on opposite sides is a reversible transition.

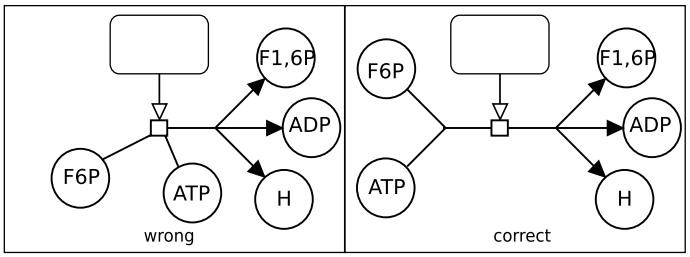

Figure 4.6: *Arcs between a* transition *and the* consumption *and* production arcs *must be attached to the center of opposite sides, modulatory arcs must be attached to the other two sides.*

4.2.1.7 Node labels

At least a part of the label (unbordered box containing a string of characters) has to be placed inside the node it belongs to. Node labels are not allowed to overlap nodes or other labels (this includes touching of other nodes or labels).

4.2.1.8 Edge labels

Edge labels are not allowed to overlap nodes. This includes touching of nodes.

4.2.1.9 Compartments

If a transition has all participants in the same compartment the transition node and all edges/arcs have to be in this compartment. If a transition has participants in at least two different compartments, the transition node has to be either in a compartment where the transition has at least one participant or in the empty space.

4.2.2 Recommendations

Recommendations are rules which should be followed if possible to produce layouts may be easier to understand.

4.2.2.1 Node-edge crossing

Crossings between edges and nodes should be avoided. Some crossings may be unavoidable, e.g., the crossing between an edge and a compartment border or an edge and a complex (if the edge connects an element inside the complex with something outside). See also requirement 4.2.1.2 (in case of node-edge crossings the edge must be drawn on the top of the node).

4.2.2.2 Labels

Labels should be horizontal. Node labels should be placed completely inside the node if possible.3 Edge labels should be placed close to the edge and avoid overlapping the edge as well as other edge labels.

4.2.2.3 Avoid edge crossings

The amount of crossings between edges should be minimized.

4.2.2.4 *Branching of* association *and* dissociation

The branching points of *association* and *dissociation* nodes should be placed closed to the symbol of the transition, if possible at a distance comparable than, or smaller to, the diameter of the symbol defining the transition (Figure 4.7).

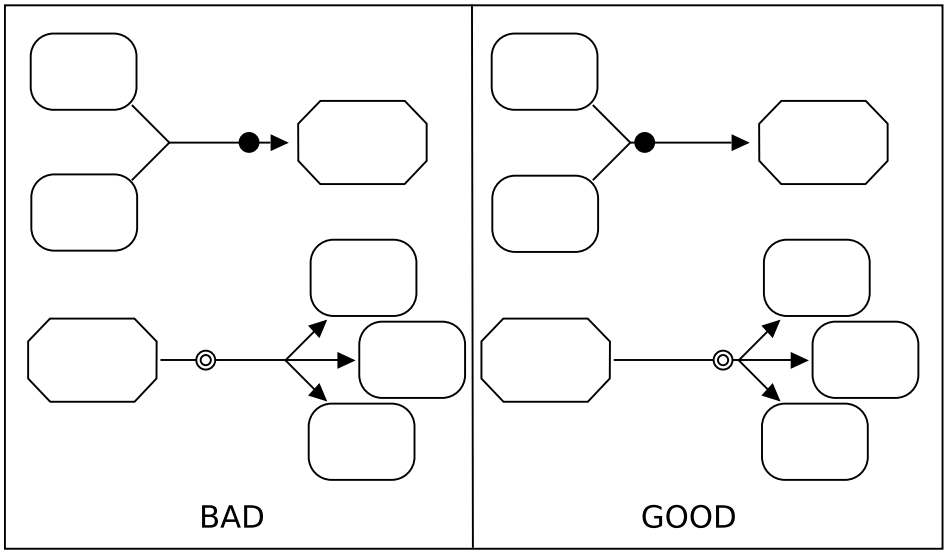

Figure 4.7: *Branching points should be close to association and dissociation symbols.*

4.2.2.5 Units of information

Units of information should not hide the structure of the corresponding node and should not overlap other elements (Figure 4.8 on the next page).

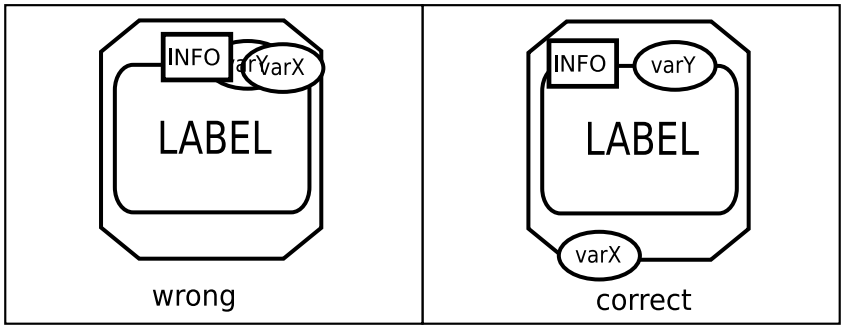

Figure 4.8: *Units of information should not overlap with any other element.*

4.2.3 Additional suggestions

Here is a list of additional layout suggestions which may help in producing aesthetically more pleasing layouts which may be easier to understand.

- Angle of edge crossings: If edge crossings are not avoidable edges should cross with an angle close to 90 degrees.
- Placement of substrates and products of a transition: Substrate and product nodes should be placed on different sides of the transition node.
- Drawing area and width/height ratio: The drawing should be compact and the ratio between the width and the height of the drawing should be close to 1.
- Edge length: Long edges should be avoided.
- Number of edge bends: Edges should be drawn with as few bends as possible.
- Similar and symmetric parts: Similar parts of a diagram should be drawn in a similar way, and symmetric parts should be drawn symmetrically.
- Proximity information: Related elements (e.g., nodes connected by a transition or all elements within a compartment) should be drawn close together.
- Directional information: Subsequent processes (e.g., a sequence of reactions) should be drawn in one direction (e.g., from top to bottom or from left to right).
- Compartments: Different compartments should have different background shade or color.

Bibliography

[1] Gerhard Michal. On representation of metabolic pathways. *BioSystems*, 47:1–7, 1998.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] E. Demir, O. Babur, U Dogrusoz., A. Gursoy, G. Nisanci, R. Cetin-Atalay, and M. Ozturk. Patika: an integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics*, 18(7):996–1003, 2002.

[7] E. Demir, O. Babur, U. Dogrusoz, A. Gursoy, A. Ayaz, G. Gulesir, G. Nisanci, and R. Cetin-Atalay. An ontology for collaborative construction and analysis of cellular pathways. *Bioinformatics*, 20(3):349–356, 2004.

[8] 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.

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

[10] 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.

[11] 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*, 3:36.1–11, 2006.

[12] 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.

[13] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, New Jersey, 1998.

[14] M. Kaufmann and D. Wagner. *Drawing Graphs: Methods and Models*, volume 2025 of *Lecture Notes in Computer Science Tutorial*. Springer, 2001.