Systems Biology Graphical Notation: Activity Flow language Level 1

Version 1.3

Date: December 19, 2025

Editors:

Hasan Balci	*NIH, USA*
Adrien Rougny	*University of Luxembourg, Luxembourg*
Luiz Ladeira	*University of Liège, Belgium*
Huaiyu Mi	*University of Southern California, USA*
Falk Schreiber	*University of Konstanz, Germany*
Stuart Moodie	*Novo Nordisk A/S, Denmark*
Emek Demir	*OHSU, USA*
Robin Haw	*Ontario Institute for Cancer Research, Canada*
Anatoly Sorokin	*InSysBio UK LTD, UK*
Alice Villéger	*Freelance IT Consultant, UK*
Rupert Overall	*Humboldt University, Germany*
Tobias Czauderna	*University of Applied Sciences Mittweida, Germany*
Augustin Luna	*NIH, USA*

To discuss any aspect of SBGN, please send your messages to the mailing list sbgn-discuss@googlegroups.com. *To get subscribed to the mailing list or to contact us directly, please write to* sbgn-editors@googlegroups.com. *Bug reports and specific comments about the specification should be entered in the issue tracker at* https://github.com/sbgn/activity-flow/issues.

Contents

Chapter 1

Introduction

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

The goal of the Systems Biology Graphical Notation (SBGN) is to standardise the graphical/visual representation of biochemical and cellular processes. SBGN defines comprehensive sets of symbols with precise semantics, together with detailed syntactic rules defining their use. It also describes the manner in which such graphical information should be interpreted. SBGN is made up of three different and complementary languages [1]. This document defines the *Activity Flow* visual language of SBGN. Activity Flows are one of three views of a biological process offered by SBGN. It is the product of many hours of discussion and development by many individuals and groups.

1.1 What are the languages?

The **Process Description** language permits the description of all the processes taking place in a biological system. The **Entity Relationship** language permits the description of all the relations involving the entities of a biological system. The **Activity Flow** language permits the description of the flow of activity in a biological system.

1.2 Nomenclature

The three languages of SBGN should be referred to as:

- the Process Description language (the PD language).
- the Entity Relationship language (the ER language).
- the Activity Flow language (the AF language).

A specific representation of a biological system in one of the SBGN languages should be referred to as:

- a Process Description map (a PD map).
- an Entity Relationship map (an ER map).
- an Activity Flow map (an AF map).

The corpus of all SBGN representations should be referred to as:

- Process Descriptions.
- Entity Relationships.
- Activity Flows.

The capitalisation is important. PD, ER and AF are names of languages. As such they must be capitalised in English. This is not the case of the accompanying noun (language or map).

1.3 SBGN levels and versions

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

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

Capabilities and features that cannot be agreed upon and are judged insufficiently critical to require inclusion in a given level, are postponed to a higher level or version. In this way, the development of SBGN languages is envisioned to proceed in stages, with each higher 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@googlegroups.com`) 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.

1.5 Note on the typographical conventions and requirement levels

The concept represented by a glyph is written using a regular font, while a *glyph* means the SBGN visual representation of the concept. For instance, “a biological activity is encoded by the SBGN AF *biological activity*.”

Throughout this specification, we use two requirement levels, indicated by the keywords “must” and “should”:

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

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

1.6 Structure of this document

Chapter 2 provides a catalogue of the graphical symbols available for representing entities in Activity Flows. In Chapter 3 beginning on page 24, we describe the rules for combining these glyphs into a legal SBGN Activity Flow map, and in Chapter 4 beginning on page 27, we describe requirements and guidelines for the way that Activity Flow maps are visually organised.

Chapter 2

Activity Flow glyphs

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.

The diagram illustrates the regulation of peroxisome proliferator-activated receptor delta (PPAR delta, a nuclear hormone receptor) on brown fat metabolism, a redraw from Fig. 7E of Pan et al. [2]. The rectangle nodes represent *biological activities* - activities from biological materials. The type of material is indicated in the units of information that decorate the activity nodes (See Section 2.3.1). Each biological activity can influence, or be influenced by, other biological activities, and such relationships are represented in Activity Flow by lines with arrows and other decorations. It should be noted that the essence of Activity Flow is to show the flow of activities from one entity to another or within the same entity. For example, in the diagram, it shows that PPARδ positively influences the Twist-1 gene expression. The underlying mechanisms of how the influence occurs may not be known and are not captured in the diagram. If the mechanism is known, the details could be described in a Process Description and/or Entity Relationship map.

Table 2.1 summarises the different SBGN abstractions described in this chapter.

Component	**Abbrev.**	**Role**	**Examples**
Activity node	AN	A functional unit that can affect, or be affected by, another functional unit	Biological activity or phenotype
Biological activity	BA	Glyph to represent molecular activities	Biological activity
Compartment node	CN	An encapsulation of one or more other SBGN constructs	Compartment
Influence arc	IA	Links between different activities to indicate influences	Positive influence, Negative influence
Auxiliary units	AU	A decorating glyph added to the BA, compartment or submap to provide additional information about the node, such as the property where the activity originates, in the case of a BA	Unit of information or submap terminal
Logical operators	LO	Combines one or several inputs into one output	Boolean *and*, *or*, *not*, *delay*

Table 2.1: *Summary of Activity Flow components and their roles.*

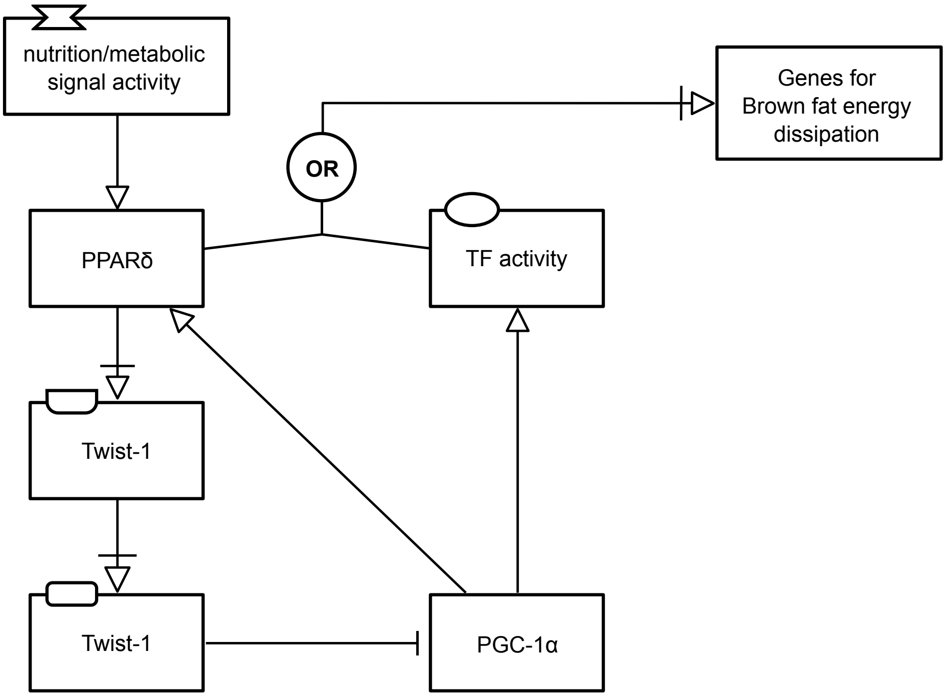

nutrition/metabolic signal activity
Genes for Brown fat energy dissipation
OR
PPARδ
TF activity
Twist-1
Twist-1
PGC-1α

Figure 2.1: *This example of Activity Flow is based on Figure 7E of Pan el. al. [2]. It depicts the effect of nutrients and metabolic signals on brown fat metabolism through PPARδ. The signal, shown as a perturbation, positively influences the nuclear hormone receptor PPARδ, which in turn stimulates the Twist 1 gene expression. Please note the different* units of information *on Twist-1 activity nodes that indicate the activity from different biological materials (gene and protein). The Twist-1 protein negatively influences the PGC-1α activity, which positively influences PPARδ and other unspecified transcription factor activity to stimulate the expression of genes for brown fat energy dissipation. Therefore, the Twist-1, induced by PPARδ, serves as a negative feedback regulator of PGC-1α in brown fat metabolism.*

2.2 Controlled vocabularies used in SBGN Activity Flow Level 1

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 2.4.1). An example is in the case of a *biological activity*, which can have a *unit of information* conveying the type of entity the activity is from.

The text that appears as the *unit of information* decorating a Compartment Node (CN) must, in most cases, be prefixed with a controlled vocabulary term indicating the type of information being expressed.

In the rest of this section, we describe the controlled vocabularies (CVs) used in SBGN Activity Flow Level 1. 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 term's meanings in a Figure legend or other text accompanying the map.

2.2.1 Unit of information material types

The material type of an AN can be visualised in the *unit of information* glyph to indicate its chemical structure. A list of common material types is shown in Table 2.2 on the next 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 *entity*). It is optional to use them in the text to label the *unit of information*.

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 artefact, but its use as messenger RNA is a role.

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

Table 2.2: *A sample of values from the* material types *controlled vocabulary (Section 2.2.1).*

2.2.2 Unit of information conceptual types

A *conceptual type* indicates the function within the context of a given Activity Flow. A list of common conceptual types is shown in Table 2.3, 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 *entity*). It is optional to use them in the text to label the *unit of information*.

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 2.3: *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 2.4 lists the particular values defined by SBGN Activity Flow Level 1. The values correspond to the Systems Biology Ontology branch with the identifier `SBO:0000002` (*quantitative parameter*).

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

Table 2.4: *A sample of values from the* physical characteristics *vocabulary (Section 2.2.3).*

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. In addition to activities of material entities, SBGN Activity Flow Level 1 represents activity from a conceptual entity: *phenotype*. Auxiliary units, such as *units of information*, can be used to indicate the material property of the activity source. Each activity is displayed only once in a particular compartment.

2.3.1 Glyph: *Biological activity*

SBGN Activity Flow Level 1 uses one glyph to represent molecular activities of all types of biological entities, collectively they are called *biological activity*. The nature of the molecule that the activity comes from, e.g., simple chemical or macromolecule, can be encoded in the *units of information* (Section 2.4.1).

It should be noted that the *biological activity* is not equivalent to a biological entity per se. A biological activity can come from one biological entity, a part of an entity, or a combination of them. It is up to the users to determine how to represent it in their diagram. For example, a protein kinase receptor such as an EGF receptor, has two activities, the binding activity that allows the extracellular part of the receptor to bind to the ligand, and the kinase activity that is capable of phosphorylating the downstream protein and initiating the intracellular signalling. The user can choose to use two nodes to represent each activity, or to use one node to represent the overall "EGF receptor activity" (Figure 2.3). More examples can be found in examples in Appendix A.

SBO Term:

SBO:0000412 ! biological activity

Incoming arcs:

Zero or more *influence* arcs (Section 2.6).

Outgoing arcs:

Zero or more *influence* arcs (Section 2.6), *logic arcs* (Section 2.8.1), or *equivalence arcs* (Section 2.10.2).

Container:

A *biological activity* is represented by a rectangle, as shown in Figure 2.2.

Label:

A *biological activity* 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 centre of the container. The label may spill outside of the container.

Auxiliary items:

A *biological activity* can carry a *unit of information* (Section 2.4.1), which can provide information such as the nature of the entity from which the activity originated. Specific glyphs are used to represent different types of entities (Section 2.4.1). The centre of the bounding box of a *unit of information* must lie on the border of the *biological activity* node. The label in the *unit of information*, which is optional, indicates the name of the molecule where the activity comes from, as shown in Figure 2.3.

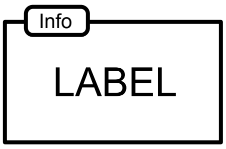


Figure 2.2: *The Activity Flow glyph for* biological activity.

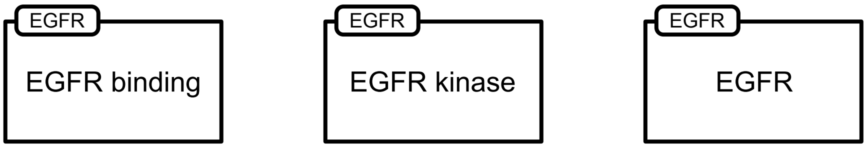


Figure 2.3: *An example of Activity Flow glyphs of EGFR activities. Since EGFR protein has both binding and kinase activities, each of those activities can be represented by different nodes, labelled as* EGFR binding *and* EGFR kinase*. One node can be used to represent the overall activity of* EGFR*. The label in the* unit of information *indicates the protein that the activities come from. In this example, all three activities come from the same EGFR protein.*

2.3.2 Glyph: *Phenotype*

A phenotype is a type of biological process. In SBGN, *phenotype* is used to show the observable or measurable outcome of the network. It is usually the end-point(s) of the network, i.e., it cannot be used as the start of an arc.

SBO Term:
SBO:0000358 ! phenotype

Incoming arcs:
Zero or more *influence* arcs (Section 2.6).

Outgoing arcs:
None.

Container:
A *phenotype* is represented by an elongated hexagon, as illustrated in Figure 2.4.

Label:
A *phenotype* 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 centre of the *phenotype* container. The label may spill outside of the container.

Auxiliary items:
None.

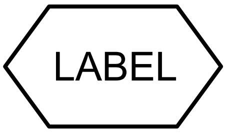


Figure 2.4: *The Activity Flow glyph for* phenotype.

2.4 Auxiliary units

2.4.1 Glyphs: *Units of information for Biological activity*

When representing biological activities, it is often useful to illustrate the nature of the entity where the activity is originated, e.g., whether the activity is from a macromolecule (protein or nucleic acid), or from a chemical compound. The SBGN Activity Flow Level 1 *unit of information* is used to add such information to a glyph. It represents the information in two ways. First, different symbols are used to represent the nature of the entity where the activity is from, e.g., macromolecule, nucleic acid feature, or complex. These symbols are identical to the *entity pool node* symbols in SBGN Process Description language. Second, names of the entity (gene names, protein names) are usually provided as labels in the *unit of information* container.

SBO Term:

Not applicable.

Container:

A unit of information is represented by containers of different shapes, depending on the nature of the entity where the biological activity is from. There are a total of six types of units of information, as shown in Figure 2.5. Below is a summary of the six glyphs.

A. Macromolecule – Macromolecules 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.). A *macromolecule unit of information* is represented by a rectangle with rounded corners, as illustrated in (A) of Figure 2.5. This container is used to decorate a biological activity that is originated from a macromolecule, such as a protein, a nucleic acid, or a complex sugar.

B. Nucleic acid feature – The nucleic acid feature construct in SBGN is meant to represent a fragment of a macromolecule carrying genetic information. A *nucleic acid feature unit of information* is represented by a rectangle whose bottom half has rounded corners, as shown in (B) of Figure 2.5.

C. Simple chemical – A simple chemical is a chemical compound that is not formed by the covalent linking of pseudo-identical residues. Examples of simple chemicals are an atom, a monoatomic ion, a salt, a radical, a solid metal, a crystal, etc. A *simple chemical unit of information* is represented by a “stadium” shape, that is two semicircles of the same radius joined by parallel line segments, as shown in (C) of Figure 2.5. If desired the parallel line segments can have zero length, and the shape is then identical to a circle. To avoid confusion with the unspecified entity (shown in D of (Figure 2.5), this form of the glyph must remain a circle and cannot be deformed into an ellipse.

D. Unspecified entity – An unspecified entity is used to represent the entity type that is unknown or simply not relevant to the purposes of the map. This arises, for example, when the existence of the entity has been inferred indirectly, or when the entity is merely a construct introduced for the needs of a map, without direct biological relevance. An *unspecified entity unit of information* is represented by an elliptic container, as shown in (D) of Figure 2.5. It is used to decorate a biological activity that is originated from an unspecified entity.

E. Complex – A complex represents a biochemical entity composed of other biochemical entities, whether macromolecules, simple chemicals, or other complexes. The resulting entity may have its own identity, properties and function in an SBGN map. A *complex unit of information* is represented by a rectangular shape with cut-corners (that is, an octagonal shape with sides of two different lengths) as shown in (E) of Figure 2.5. It is used to decorate a biological activity that is originated from a complex.

F. 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, glucose deprivation, stress. A *perturbation unit of information* is represented by a modified hexagon having two opposite concave faces, as illustrated in F of Figure 2.5. It is used to decorate a biological activity when it is originated from a perturbation.

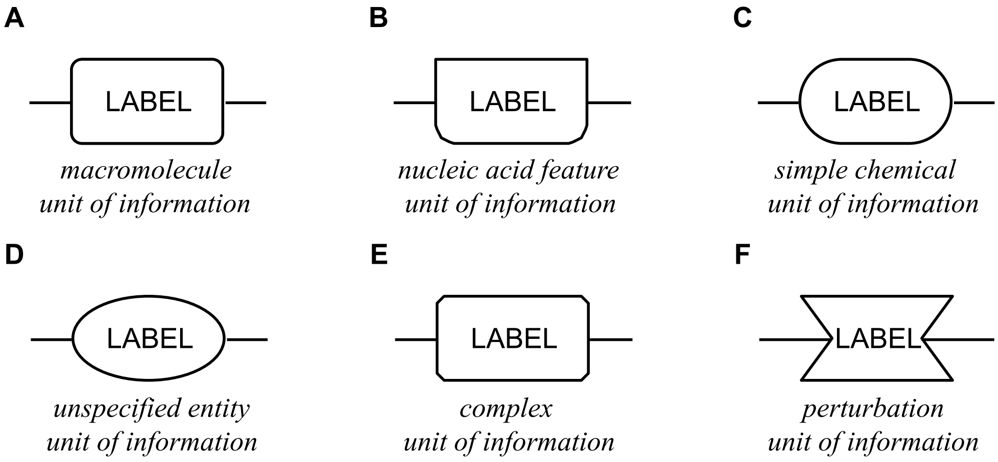


Figure 2.5: *The Activity Flow glyph for* unit of information.

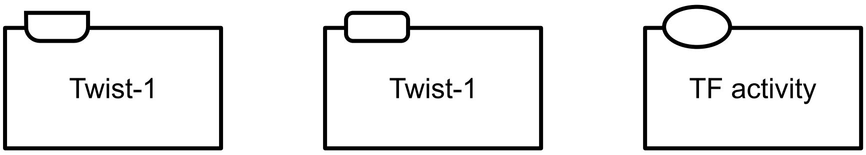


Figure 2.6: *Examples of* unit of information *used on* biological activity node *to indicate that the* Twist-1 activity *is from a* nucleic acid feature *or a* macromolecule, *or a* transcription factor activity *from an* unspecified *entity.*

Figure 2.6 shows examples taken from Figure 2.1, where *units of information* are used on *biological activities* to illustrate the nature of the entities from which the activities originate.

The long side of the glyphs above must be orientated parallel to the border of the biological activity being annotated by the unit of information. The centre of the bounding box of a unit of information must lie on the border of the *BA*.

Label:

A *unit of information* is not required to carry any label. If a label is desired, it must be 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 centre of the container. The label may spill outside of the container. The label defines the information carried by the *unit of information.*

Auxiliary items:

None.

2.4.2 Glyph: *Unit of information for Compartment*

A *unit of information* can be used to decorate a compartment to convey information about physical characteristics of the compartments (Section 2.2.3).

SBO Term:

Not applicable.

Container:

A *unit of information* for a compartment is represented by a rectangle as shown in Figure 2.7. The long side of the rectangle must be orientated parallel to the border of the *compartment* being annotated by the *unit of information*. The centre of the bounding box of a *unit of information* must lie on the border of the *compartment*.

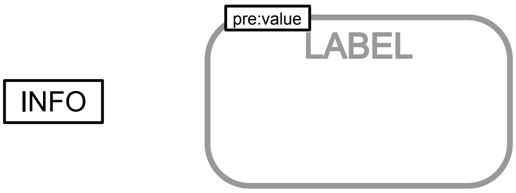


Figure 2.7: *The Activity Flow glyph for* unit of information, *shown plain on the left, and decorating a* compartment *(Section 2.5) on the right.*

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 centre of the container. The label may spill outside of the container. The label defines the information carried by the *unit of information*. The controlled vocabularies predefined in SBGN Activity Flow Level 1 for *unit of information*, regarding compartments, are described in Section 2.2.3.

Auxiliary items:

None.

2.4.3 Glyph: *Submap terminal*

A *submap terminal* is a decorator of the *submap* (Section 2.11). It is a named handle, or reference, to both an *AN* (Section 2.3) or *compartment* (Section 2.5) of the map, and a *tag* (Section 2.10.1) of the map the *submap* glyph refers to. Together with the *tag*, it allows linking glyphs of a map to their counterpart lying in a submap.

SBO Term:

Not applicable.

Incoming arcs:

One *equivalence arc* (Section 2.10.2).

Outgoing arcs:

None.

Container:

A *submap terminal* is represented by a rectangular shape fused to an empty arrowhead, as shown in Figure 2.8. The flat edge opposite to the arrowhead must be aligned to the edge of the *submap* glyph, and the incoming *equivalence arc* (Section 2.10.2) must be linked to its middle.

Label:

A *submap terminal* is identified by a label that is a string of characters that may be distributed on several lines to improve readability. The centre of the label must be placed on the centre of the container. The label may extend outside of the container.

Auxiliary items:

None.

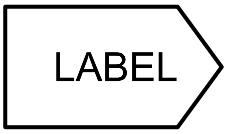


Figure 2.8: *The Activity Flow glyph for* submap terminal.

2.5 Glyph: Compartment

A compartment is a logical or physical structure where the function or activity is located. At the moment, an activity can only belong to one compartment. Therefore, the "same" biochemical activities located in two different compartments are in fact, two different activities, and should be represented separately.

SBO Term:
SBO:0000289 ! functional compartment

Incoming arcs:
None.

Outgoing arcs:
Zero or more *equivalence arcs* (Section 2.10.2).

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 ANs. 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 2.4.2. The centre of the bounding box of a *unit of information* must lie on the border of the compartment.

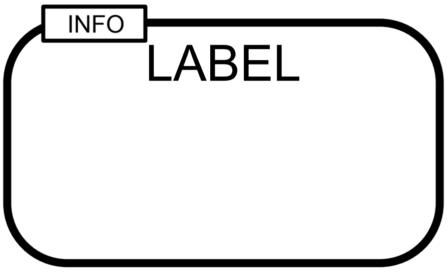


Figure 2.9: *The Activity Flow glyph for* compartment.

It is important to note that a compartment never contains another compartment. To allow more aesthetically pleasing and understandable diagrams, compartments are allowed to overlap each other visually, but it must be kept in mind that this does not mean one compartment contains part or entire of the other compartment. Figure 2.10 shows three semantically equivalent placements of compartments:

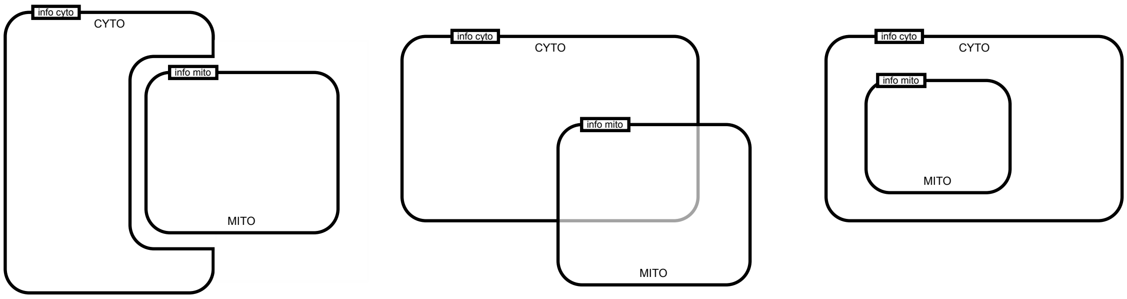


Figure 2.10: *Overlapped compartments are permitted, but the overlap does not imply containment.*

2.6 Influence arcs

Influence arcs represent direct or indirect influences of biological activities on other activities. SBGN Activity Flow Level 1 provides four influence arcs: *positive influence*, *negative influence*, *unknown influence* and *necessary stimulation*.

2.6.1 Glyph: *Positive influence*

A *positive influence* represents that an activity exerts a **positive** or **stimulating** effect on another activity.

SBO Term:

SBO:0000170 ! stimulation

Origin:

One *biological activity* (Section 2.3.1) or *logical operator* (Section 2.7).

Target:

One *biological activity* (Section 2.3.1) or *phenotype* (Section 2.3.2).

Symbol:

The target extremity of a *positive influence* carries an open arrow pointing to the target activity node (Figure 2.11).

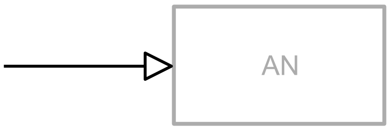


Figure 2.11: *The Activity Flow glyph for* positive influence.

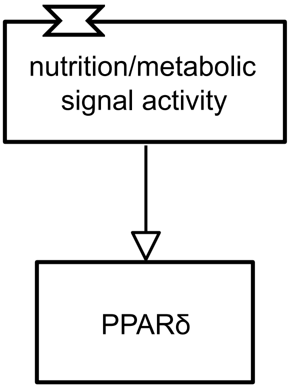


Figure 2.12: *An example, taken from Figure 2.1, of* positive influence *from a* perturbation *to the nuclear hormone receptor* PPARδ*.*

2.6.2 Glyph: *Negative influence*

A *negative influence* represents that an activity exerts a **negative** or **inhibiting** effect on another activity.

SBO Term:
SBO:0000169 ! inhibition

Origin:
One *biological activity* (Section 2.3.1) or *logical operator* (Section 2.7).

Target:
One *biological activity* (Section 2.3.1) or *phenotype* (Section 2.3.2).

Symbol:
The target extremity of a *negative influence* carries a bar perpendicular to the arc (Figure 2.13).

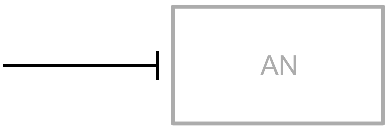


Figure 2.13: *The Activity Flow glyph for* negative influence*.*

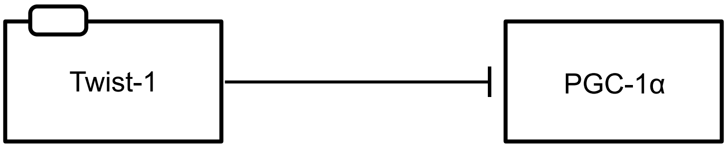


Figure 2.14: *An example, taken from Figure 2.1, of* negative influence *from* Twist-1 *protein activity to* PGC-1α *activity.*

2.6.3 Glyph: *Unknown influence*

An *unknown influence* represents that an activity affects another activity, but the exact nature of this effect is not specified or not known.

SBO Term:
SBO:0000168 ! control

Origin:
One *biological activity* (Section 2.3.1) or *logical operator* (Section 2.7).

Target:
One *biological activity* (Section 2.3.1) or *phenotype* (Section 2.3.2).

Symbol:
The target extremity of an *unknown influence* carries an open diamond (Figure 2.15).

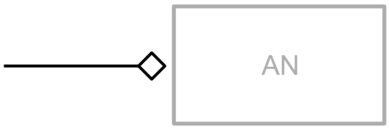


Figure 2.15: *The Activity Flow glyph for* unknown influence*.*

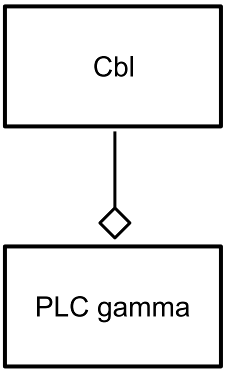


Figure 2.16: *An example, taken from Figure A.2, of* unknown influence *from the adaptor protein* Cbl *to the signalling enzyme* PLC gamma*.*

2.6.4 Glyph: *Necessary stimulation*

A *necessary stimulation* represents that the target activity can only take place if the source activity takes place, regardless of other influences on the target.

SBO Term:
SBO:0000171 ! necessary stimulation

Origin:
One *biological activity* (Section 2.3.1) or *logical operator* (Section 2.7).

Target:
One *biological activity* (Section 2.3.1) or *phenotype* (Section 2.3.2).

Symbol:
The target extremity of a *necessary stimulation* carries a perpendicular bar followed by an open arrow pointing to the target activity node (Figure 2.17). The bar must be at least as long as the base of the arrowhead.

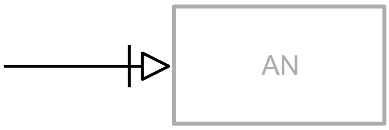


Figure 2.17: *The Activity Flow glyph for* necessary stimulation*.*

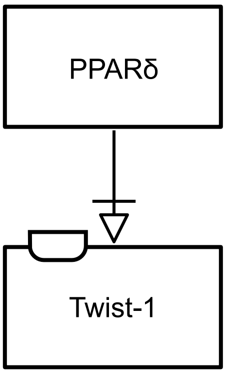


Figure 2.18: *An example, taken from Figure 2.1, of* necessary stimulation *where nuclear hormone receptor* PPARδ *transcription factor activity is necessary for the stimulation of the* Twist-1 *gene expression.*

2.7 Logical operator nodes

The logical operators *and*, *or* and *not* perform a Boolean operation to give a binary output. An Activity Flow map can represent a dynamic model in which activity nodes are considered to have states. Activity above a certain threshold is treated as True and activity below this threshold is treated as False. The *delay* glyph is currently categorised as a logical operator because it uses *logic arcs* for its connections, although it does not perform a logical operation.

2.7.1 Glyph: *And*

The output of an *and* glyph is True if all its inputs are True, and False otherwise.

SBO Term:
SBO:0000173 ! and.

Incoming arcs:
One or more *logic arcs* (Section 2.8.1).

Outgoing arcs:
One *logic arc* (Section 2.8.1) or one of the influence arcs (Section 2.6).

Container:
An *and* operator is represented by a circular shape containing the word "AND". The shape is linked to two ports, that are small arcs attached to the centres of opposite sides of the shape, as shown in Figure 2.19.

Label:
None.

Auxiliary items:
None.

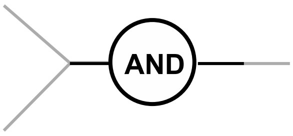


Figure 2.19: *The Activity Flow glyph for* and. *Only two inputs are represented, but more would be allowed.*

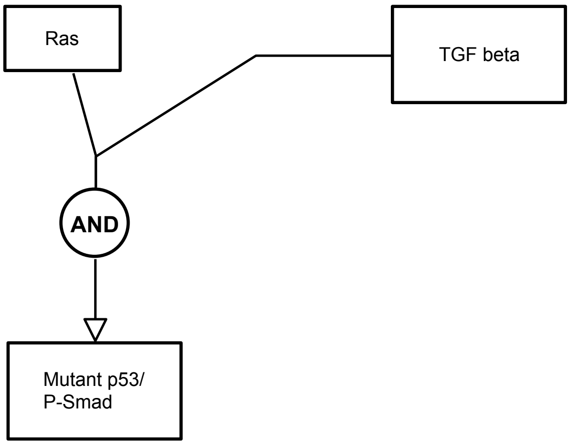


Figure 2.20: *An example, taken from Figure A.1, of the* and *logical operator, where the activity from* "Mutant p53/P-Smad" *is positively influenced only when both* Ras *and* TGF beta *activities are present.*

2.7.2 Glyph: *Or*

The output of an *or* glyph is True if at least one of its inputs is True, and False otherwise.

SBO Term:
SBO:0000174 ! or.

Incoming arcs:
One or more *logic arcs* (Section 2.8.1).

Outgoing arcs:
One *logic arc* (Section 2.8.1) or one of the influence arcs (Section 2.6).

Container:
An *or* operator is represented by a circular shape containing the word "OR". The shape is linked to two ports, that are small arcs attached to the centres of opposite sides of the shape, as shown in Figure 2.21.

Label:
None.

Auxiliary items:
None.

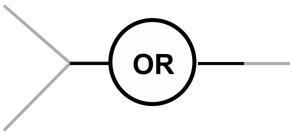


Figure 2.21: *The Activity Flow glyph for* or. *Only two inputs are represented, but more would be allowed.*

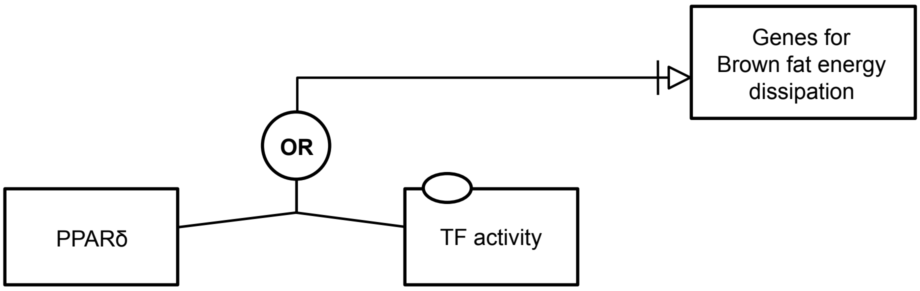


Figure 2.22: *An example, taken from Figure 2.1, of the* or *logical operator, where the activity from* "Genes for brown fat energy dissipation" *is positively influenced by either the* PPAR delta *activity or an* unspecified transcription factor *activity.*

2.7.3 Glyph: *Not*

The output of a *not* glyph is True if its input is False, and False otherwise.

SBO Term:

SBO:0000238 ! not.

Incoming arcs:

One *logic arc* (Section 2.8.1).

Outgoing arcs:

One *logic arc* (Section 2.8.1) or one of the influence arcs (Section 2.6).

Container:

A *not* operator is represented by a circular shape containing the word "NOT". The shape is linked to two ports, that are small arcs attached to the centres of opposite sides of the shape, as shown in Figure 2.23.

Label:

None.

Auxiliary items:

None.

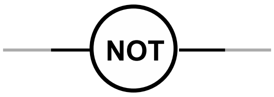


Figure 2.23: *The Activity Flow glyph for* not.

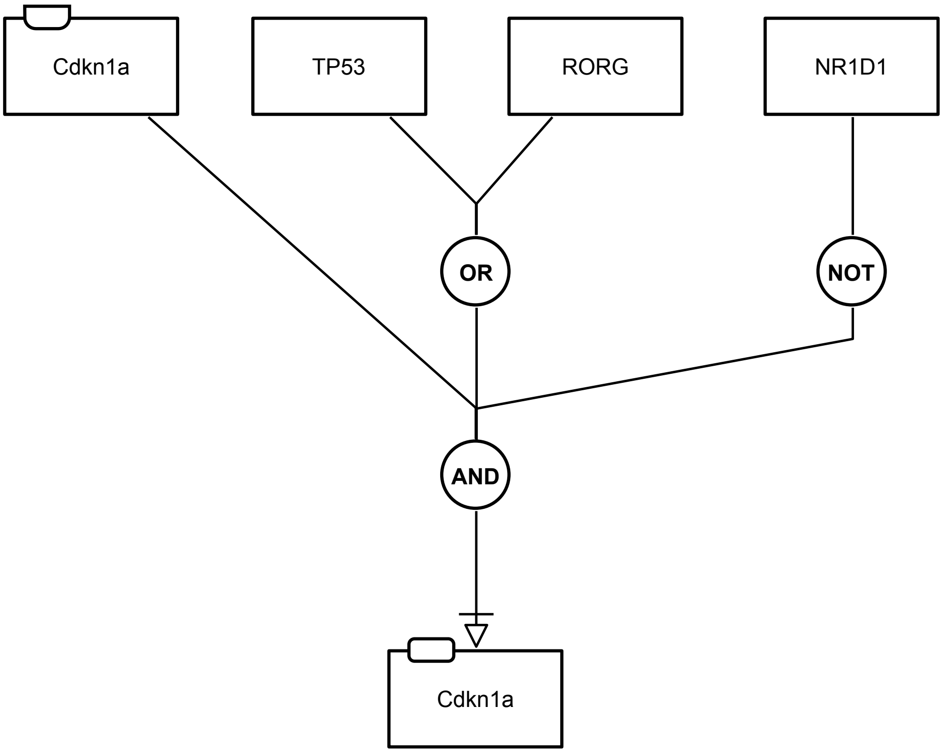


Figure 2.24: *An example of the* not *logical operator, where the activity of* NR1D1 *has a negative influence on the activity of* Cdkn1a *taking place in the regulation of the circadian clock.*

2.7.4 Glyph: *Delay*

The *delay* glyph represents that the input, connected via a logic arc from a *BA* or the output of another logical operator, does not produce its influence immediately on the target activity.

SBO Term:

SBO:0000225 ! delay.

Incoming arcs:

One *logic arc* (Section 2.8.1).

Outgoing arcs:

One *logic arc* (Section 2.8.1) or one of the influence arcs (Section 2.6).

Container:

A *delay* operator is represented by a circular shape containing the symbol “τ“ (letter “tau” of the Greek alphabet). The shape is linked to two ports, that are small arcs attached to the centres of opposite sides of the shape, as shown in Figure 2.25.

Label:

None.

Auxiliary items:

None.

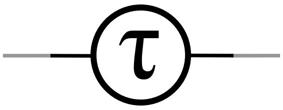


Figure 2.25: *The Activity Flow glyph for* delay*.*

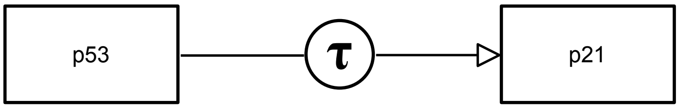


Figure 2.26: *An example of the* delay *logical operator, where the activity of* p21 *is positively influenced after a delay by the activity of* p53*, modelling the time-dependent transcriptional activation of p21 by p53.*

2.8 Logic arc

2.8.1 Glyph: *Logic arc*

Logic arc is used to represent the fact that an activity influences the outcome of a logic operator.

SBO Term:
SBO:0000398 ! logical relationship.

Origin:
One *biological activity* (Section 2.3.1) or *logical operator* (Section 2.7).

Target:
One *logical operator* (Section 2.7).

Symbol:
No symbol is used to represent the end of a logic arc.

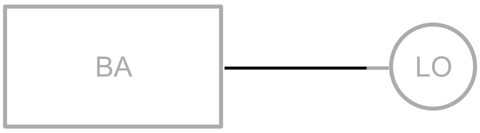


Figure 2.27: *The Activity Flow glyph for* logic arc*.*

2.9 Annotating nodes and arcs

2.9.1 Glyph: *Annotation*

SBGN Activity Flow Level 1 defines a glyph to add additional information to a map, that does not modify the semantics of the graph. This glyph can be used to add free text, or links to external information.

SBO Term:
SBO:0000550 ! annotation

Incoming arcs:
None.

Outgoing arcs:

None.

Container:

An *annotation* is represented by a rectangular container with a folded corner, as illustrated in Figure 2.28. This container is linked to the annotated element with a callout. The link ends up on the border of the annotated element.

Label:

An *annotation* contains information 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 centre of the container. The label may spill outside of the container.

Auxiliary items:

None.

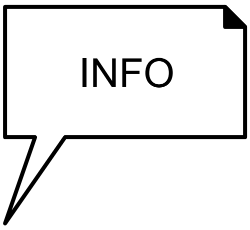


Figure 2.28: *The Activity Flow glyph for* annotation.

2.10 Referring to other nodes

2.10.1 Glyph: *Tag*

A *tag* is a named handle, or reference, to another *BA* (Section 2.3.1) or *compartment* (Section 2.5) of the map. Together with the *submap terminal* (Section 2.4.3), it allows linking glyphs of a map to their counterpart in a submap.

SBO Term:

Not applicable.

Incoming arcs:

One *equivalence arc* (Section 2.10.2).

Outgoing arcs:

None.

Container:

A *tag* is represented by a rectangular shape fused to an empty arrowhead, as shown in Figure 2.29. The incoming *equivalence arc* (Section 2.10.2) must be linked to the extremity of the arrowhead.

Label:

A *tag* is identified by a label that is a string of characters that may be distributed on several lines to improve readability. The centre of the label must be placed on the centre of the shape. The label may extend outside of the shape.

Auxiliary items:

None.

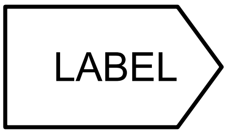


Figure 2.29: *The Activity Flow glyph for* tag.

2.10.2 Glyph: *Equivalence arc*

An *equivalence arc* is used to represent the fact that all activities or compartments marked by a *tag* are equivalent. In an Activity Flow map, it is used to show that an AN in a submap and another AN in the main map are equivalent.

SBO Term:
Not applicable.

Origin:
One *activity node* (Section 2.3) or *compartment*.

Target:
One *tag* (Section 2.10.1) or *submap terminal* (Section 2.4.3).

Symbol:
No symbol is used to represent the end of an *equivalence arc*.

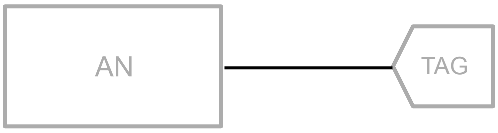


Figure 2.30: *The Activity Flow glyph for* Equivalence arc.

2.11 Glyph: Submap

A *submap* is used to encapsulate processes (including all types of nodes and edges) within one glyph. The submap hides its content from the users and displays only input *submap terminals* (or ports), linked to *ANs*. 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:
SBO:0000395 ! encapsulating process

Incoming arcs:
None.

Outgoing arcs:
None.

Container:
The *submap* is represented as a rectangle box to remind the viewer that it is fundamentally a biological activity node.

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 must be attached to the centre of the container box.

Auxiliary items:

A *submap* carries labelled *submap terminals*. When the submap is represented in its folded form, those *submap terminals* are linked to external *ANs*. In the unfolded view, which exposes the internal structure of the submap, a set of *tags* points to the corresponding internal *ANs*.

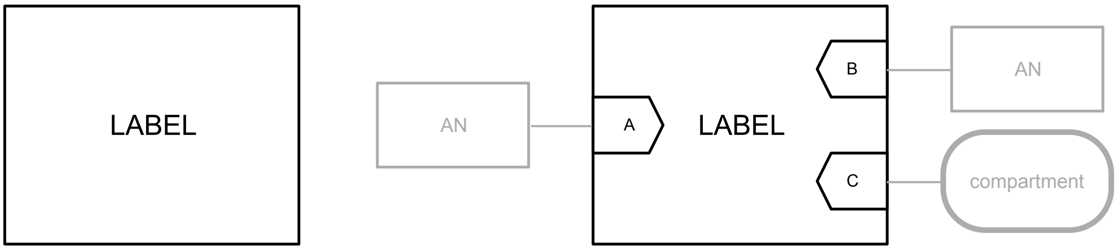


Figure 2.31: *The Activity Flow glyph for* submap, *shows plain and unadorned on the left, and with three* submap terminals *on the right folded submap. (Lower part) content of the submap.*

Figure 2.32 represents a *submap* of inhibitory G-protein coupled receptor signalling. The *submap* carries five *submap terminals*, three linked to biological ANs, and two linked to *compartments*. Note that the *submap terminals* do not define a "direction", such as input or output. The flux of the activities is determined by the context.

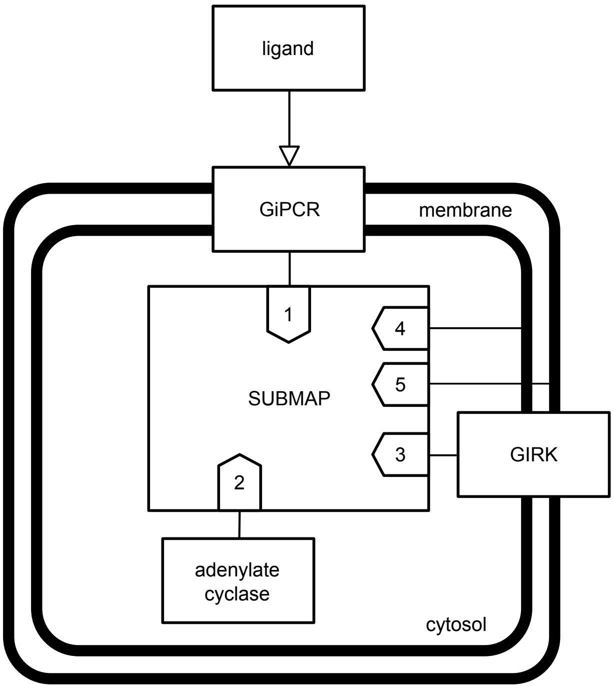


Figure 2.32: *Example of a submap with contents elided.*

The diagram in Figure 2.33 represents an unfolded version of a submap. Here, anything outside the submap has disappeared (e.g., ligand in Figure 2.32), and the internal *tags* are not linked to the corresponding external *submap terminals*.

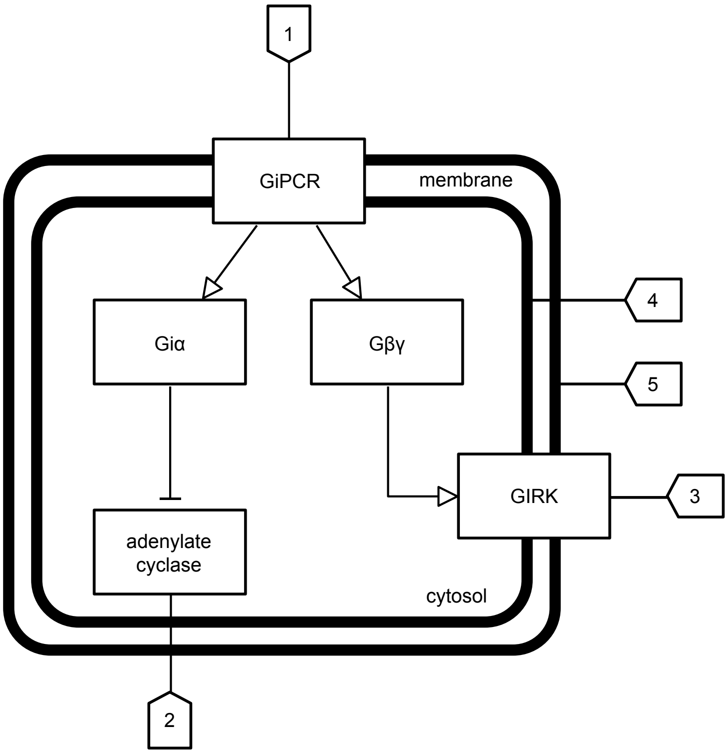


Figure 2.33: *Example of an unfolded submap. The unfolded submap corresponds to the folded submap of Figure 2.32.*

Chapter 3

Activity Flow language grammar

3.1 Overview

In this chapter, we describe how the glyphs of SBGN AF can be combined to make a valid SBGN AF map. To do this, we must at the very least define which glyphs can be connected to each other and which glyphs can contain others. In addition to connectivity and containment rules, we must also specify glyph specific rules that impose further constraints based on the biological roles represented. These rules help preserve the clarity, consistency, and interpretability of SBGN AF diagrams across different tools and users.

In this section we start off by describing the concepts of the Activity Flow notation. Next, a detailed description of the connectivity and containment is provided followed by glyph specific rules.

3.2 Concepts

The SBGN Activity Flow is more than a collection of symbols. It is a visual language that uses specific abstractions to describe the biological activities that make up a model, a signalling pathway or a metabolic network. This abstraction is the semantics of SBGN, and to describe it requires more than a definition of the symbols and syntax of the language.

The Activity Flow in SBGN describes biological activities involving biological entities. A *biological activity* can influence, or be influenced by, other *biological activities*, and such relationships are represented in Activity Flow by lines with arrows and other decorations. So, the essence of Activity Flow is to show the flow of activities from one entity to another or within the same entity. The underlying mechanisms of how the influence occurs may not be known and are not captured in the diagram. If the mechanism is known, the details should be described in annotation or captured in other SBGN languages, such as Process Description and/or Entity Relationship.

3.3 Connectivity and containment

3.3.1 Node connectivity

The syntax of the SBGN Activity Flow language is defined in the form of an incidence matrix. An incidence matrix has arcs as rows and nodes as columns. Each element of the matrix represents the role of an arc in connection to a node, as described below.

- Source (S) means that the arc can begin at that node.

- Target (T) indicates that the arc can end at that node.

- Numbers in parentheses represent the maximum number of arcs of a particular type to have this specific connection role with the node.

- Empty cells mean the arc is not able to connect to the node.

Arc\Node	*biological activity*	*phenotype*	*tag*	*submap terminal*	*and*	*or*	*not*	*delay*
positive influence	S & T	T			S(1)	S(1)	S(1)	S(1)
negative influence	S & T	T			S(1)	S(1)	S(1)	S(1)
unknown influence	S & T	T			S(1)	S(1)	S(1)	S(1)
necessary stimulation	S & T	T			S(1)	S(1)	S(1)	S(1)
logic arc	S				S(1) T	S(1) T	S(1) T(1)	S(1) T(1)
equivalence arc	S	S	T	T				

3.3.2 Containment definition

By containment we mean that a glyph can be drawn inside the other glyph. This does not necessarily mean that the glyph "belongs" to the containing node, although in some cases it does. In this section the concept of "belonging" is referred to as ownership. There are two glyphs that allow containment: *compartment* and *submap*. The next table describes relationship between AF elements of SBGN and the compartment. A + means that the element is able to be contained within a node. A − means containment is not allowed.

elements\Containers	*compartment*	*submap*
biological activity	+	−
phenotype	+	−
tag	+	−
submap terminal	+[1]	+
compartment	+	−
submap	+	−
positive influence	+	−
negative influence	+	−
unknown influence	+	−
logic arc	+	−
equivalence arc	+	−
and	+	−
or	+	−
not	+	−
delay	+	−

3.4 Glyph specific rules

3.4.1 Activity node

1. Each *biological activity* must have at most one *unit of information.*

2. *Phenotype* must not be the origin of an influence arc.

3.4.2 Activity node and Compartment

Each AN can only appear once in a particular compartment. An AN can only *belong* to one compartment. However, an AN 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.

[1] Compartments may indirectly contain submap terminals through the submaps they contain.

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

3.4.3 Influence

It is implied, but not defined explicitly that an activity has a rate at which the input AN affects the output AN.

1. The effect of a positive influence is to increase the basal activity.
2. The effect of a negative influence is to decrease the basal activity.
3. The effect of an unknown influence on the basal activity is unknown.
4. An activity can be targeted by at most one necessary stimulation. Combinations of more than one necessary stimulation must be explicitly expressed using the Boolean AND or OR operators.

3.4.4 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 reintroduce a submap into the main map by eliminating the *submap terminals* and merging the equivalent nodes in both maps.

3.4.4.1 Rules for mapping to submaps

An AN in the main map can be mapped to one in the submap using a *tag* in the submap and *submap terminals* (see Section 2.11) in the main map. For a mapping between map and submap to exists the following must be true:

1. The identifiers in the *tags* and *submap terminals* must be identical.
2. The ANs must be identical.

3.4.4.2 Requirement to define a mapping

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

Chapter 4

Layout Guidelines for an Activity Flow Maps

4.1 Introduction

The previous chapters describe the appearance and meaning of SBGN Activity Flow Level 1 components. Objects are *activity nodes*, *container nodes*, *logical operators*, *submaps* as well as *connecting arcs*. The components of an Activity Flow map 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 maps, may reveal new insights. The arrangement of components in a map is called a *layout*.

SBGN Activity Flows 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 rules, some of which are requirements and some are recommendations, for the layout of activity flow maps. 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 map, and apply to both manually drawn maps as well as maps 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 [3] and Kaufmann and Wagner [4].

Please note that the colour of objects does not carry any meaning in SBGN. Although one can use colours to emphasise part of a diagram or encode additional information, the meaning of the diagram should not depend on the colours. Furthermore, objects can have different sizes and size is also meaningless in SBGN. For example, one biological activity node may be larger than another 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 map.

4.2.1.1 Node-node overlaps

Nodes are only allowed to overlap in three cases:

1. The overlapping nodes define a glyph (e.g., *auxiliary unit*).
2. Nodes overlapping compartments (e.g., a *biological activity* placed on a compartment).
3. Compartment overlapping compartment (e.g., Figure 2.10). However, it should be noted that it does not have the implication of containment of one compartment to the other.

Otherwise, nodes are not allowed to overlap (Figure 4.1). This includes the touching of nodes, which is also not allowed. Also, submaps are not allowed to overlap. Similarly, Figure 4.2 illustrates the problem using an incorrect map where a compartment hides an AN.

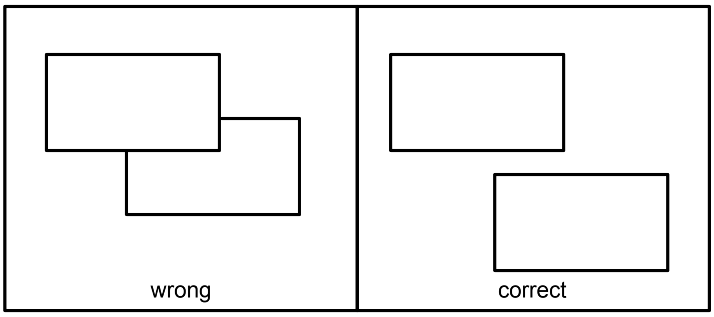


Figure 4.1: *Nodes must not overlap.*

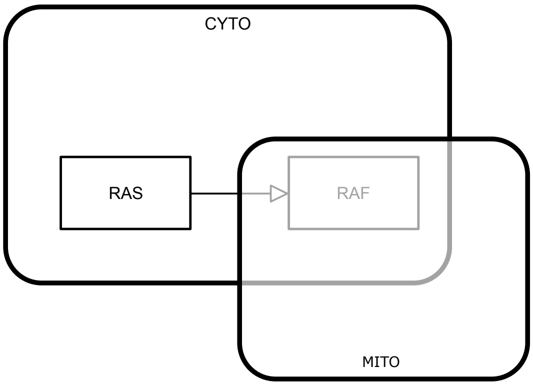


Figure 4.2: *Example of an* ***incorrect*** *diagram. Overlapped compartments must not obscure other objects.*

4.2.1.2 Node-edge crossing

In general, such crossing should be avoided. In case this cannot be avoided, the edge must be drawn on the top of the node (Figure 4.3). See also recommendation 4.2.2.2.

4.2.1.3 Node border-edge overlaps

Edges must not to overlap the border lines of nodes (Figure 4.4).

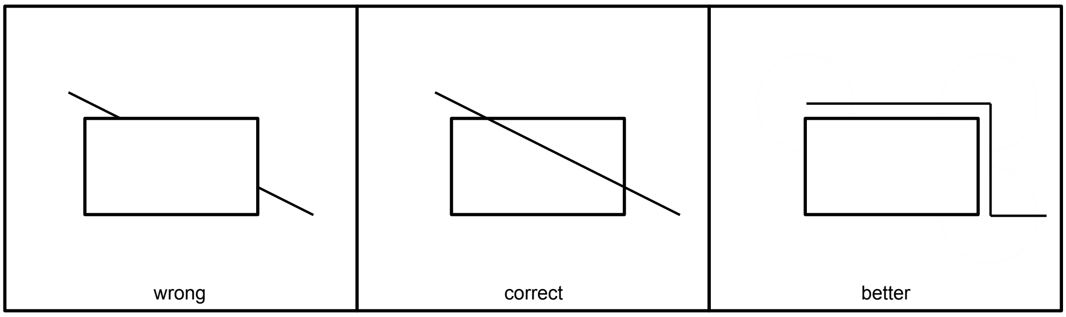


Figure 4.3: *If an edge crosses a node, the edge must be drawn on top of the node.*

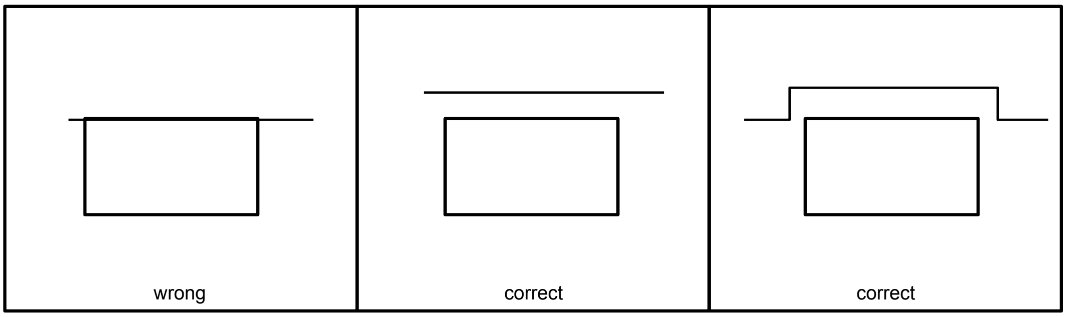


Figure 4.4: *Edges must not overlap node borders.*

4.2.1.4 Edge-edge overlaps

Edges must not overlap (Figure 4.5). This includes touching of edges. Furthermore, an edge is neither allowed to cross itself nor to cross a boundary of a node more than twice or other edges more than once.

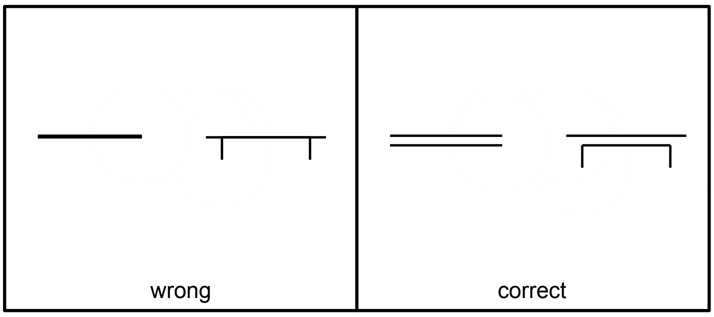


Figure 4.5: *Edges must not overlap.*

4.2.1.5 Node orientation

Nodes must be drawn horizontally or vertically; any other rotation of elements is not allowed (Figure 4.6).

4.2.1.6 Node labels

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

4.2.1.7 Edge labels

Edge labels must not overlap nodes. This includes touching of nodes.

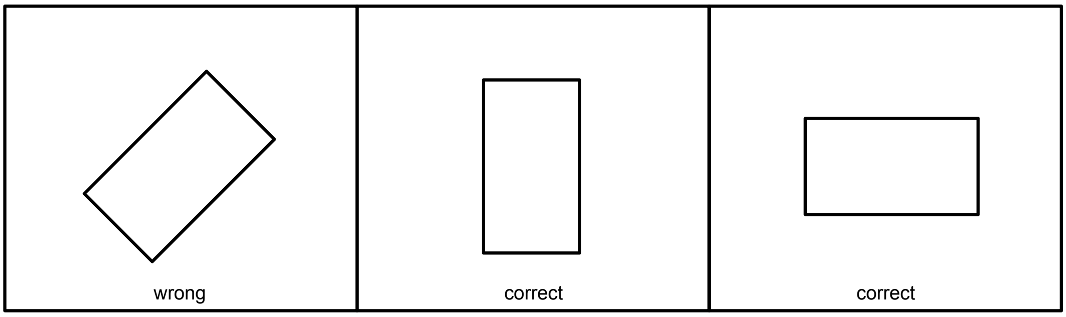


Figure 4.6: *The node orientation must be horizontally or vertically.*

4.2.1.8 *Compartments*

If a network has all participants (nodes and edges) in the same compartment, all the activity nodes and edges/arcs must be drawn in this compartment. Edges/arcs are allowed to cross the compartment boundaries when the input and output ANs are in two different compartments.

4.2.2 Recommendations

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

4.2.2.1 *Logical operators*

The incoming *logic arc(s)* (Section 2.8.1) should be linked to the extremity of the leftmost or uppermost port of logical operator glyphs (Section 2.7), while the outgoing *logic arc* (Section 2.8.1) or influence arc (Section 2.6) should be linked to the extremity of the rightmost or bottommost port for improved readability.

4.2.2.2 *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. See also requirement 4.2.1.2 (in case of node-edge crossings, the edge must be drawn on the top of the node).

4.2.2.3 *Labels*

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

4.2.2.4 *Avoid edge crossings*

The number of crossings between edges should be minimised.

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.

- Drawing area and width/height ratio: The drawing should be compact.

- Edge length: Long edges should be avoided if possible.

- Number of edge bends: Edges should be drawn with as few bends as possible.

- Similar and symmetric parts: Similar parts of a map should be drawn in a similar way, and symmetric parts should be drawn symmetrically.
- Proximity information: Related elements (e.g., nodes connected by edges within a compartment) should be drawn close together.
- Directional information: Subsequent activities (e.g., a sequence of activities) should be drawn in one direction (e.g., from top to bottom or from left to right).
- Compartments: Different compartments should have different background shades or colour.

Chapter 5

Acknowledgments

Here we acknowledge those people and organisations that assisted in the development of this and previous releases of the SBGN Activity Flow language specification. First, we specifically acknowledge those who contributed directly to each revision of the specification document, followed by a comprehensive acknowledgement of contributors that attended workshops and forum meetings or in some other way provided input to the standard. Finally, we acknowledge the bodies that provided financial support for the development of the standard.

We aim for this list to be rather complete. We are very sorry if we forgot someone and will be grateful if you notify us of any omission.

5.1 Level 1 Release 1.0

The specification was written by Huaiyu Mi, Falk Schreiber, Nicolas Le Novère, Stuart Moodie, Anatoly Sorokin, Michael Hucka, Emek Demir, Yukiko Matsuoka, Katja Wegner and Hiroaki Kitano. In addition, the specification benefited much from the help of Frank Bergmann, Sarala Dissanayake and Paul Thomas.

5.2 Level 1 Release 1.1

The specification of Activity Flow Level 1 Release 1.1 was modified by Huaiyu Mi, with significant contributions from Falk Schreiber and Stuart Moodie. Other contributors include (in alphabetic order) Tobias Czauderna, Emek Demir, Nicolas Le Novère, Yukiko Matsuoka, Anatoly Sorokin, Alice Villéger and Katja Wegner.

5.3 Level 1 Release 1.2

The specification of Activity Flow Level 1 Release 1.2 was modified by Huaiyu Mi, with significant contributions from Falk Schreiber, Stuart Moodie, Tobias Czauderna and Anatoly Sorokin. Other contributors include (in alphabetic order) Michael Blinov, Emek Demir, Robin Haw, Anushya Muruganujan, Nicolas Le Novère, Augustin Luna, Paul Thomas and Katja Wegner.

5.4 Level 1 Release 1.3

The specification of Activity Flow Level 1 Release 1.3 was modified by (in alphabetic order) Hasan Balci, Tobias Czauderna, Luiz Ladeira, Augustin Luna, Rupert Overall and Adrien Rougny.

5.5 Comprehensive list of acknowledgements

Here is a more comprehensive list of people who have been actively involved in SBGN development, either by their help designing the languages, their comments on the specification, help

with development infrastructure or any other useful input. We aim for this list to be rather complete. We are very sorry if we forgot someone and will be grateful if you notify us of any omission.

Mirit Aladjem, Hasan Balci, Frank Bergmann, Michael Blinov, Bernard de Bono, Sarah Boyd, Laurence Calzone, Melanie Courtot, David Croft, Tobias Czauderna, Emek Demir, Johannes W. Dietrich, Ugur Dogrusoz, Andreas Dräger, Damien Fleury, Tom Freeman, Akira Funahashi, Ralph Gauges, Peter Ghazal, Samik Ghosh, Igor Goryanin, Anja Hartmann, Robin Haw, Michael Hucka, Matthias Jeschke, Mathias John, Akiya Jouraku, Astrid Junker, Hideya Kawaji, Douglas Kell, Sohyoung Kim, Hiroaki Kitano, Christian Klukas, Kurt Kohn, Fedor Kolpakov, Matthias König, Luiz Ladeira, Nicolas Le Novère, Lu Li, Augustin Luna, Yukiko Matsuoka, Carsten Maus, Alexander Mazein, Huaiyu Mi, Stuart Moodie, Ulrike Münzner, Anushya Muruganujan, Rupert Overall, Michael Pedersen, Jacqueline Quinn, Adrien Rougny, Stefan Rybacki, Sven Sahle, Chris Sander, Herbert Sauro, Esther Schmidt, Falk Schreiber, Jacky Snoep, Anatoly Sorokin, Jessica Stephens, Linda Taddeo, Carolyn Talcott, Lin Uhrmacher, Martijn van Iersel, Alice Villéger, Steven Watterson, Katja Wegner (Wengler), Sarala Wimalaratne (Dissanayake), Guanming Wu, Röbbe Wünschiers, Fengkai Zhang.

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

5.6 Financial Support

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.org/english/), the Beckman Institute at the California Institute of Technology (https://beckmaninstitute.caltech.edu/), Ontario Genomics Institute (OGI, http://www.ontariogenomics.ca/), Ontario Institute for Cancer Research (OICR, http://oicr.on.ca/), National Science Foundation (NSF, http://www.nsf.gov/), USC Norris Comprehensive Cancer Center (http://uscnorriscancer.usc.edu/), Martin Luther University Halle-Wittenberg (http://www.uni-halle.de/), Monash University (http://www.monash.edu.au/), IPK Gatersleben (http://www.ipk-gatersleben.de/en/), University of Rostock (http://www.uni-rostock.de/) and German Federal Ministry of Research and Education (http://www.bmbf.de/) 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. This research was also supported in part by the Intramural Research Program of the National Institutes of Health (NIH) (ZIALM240126). The contributions of the NIH author(s) are considered Works of the United States Government. The findings and conclusions presented in this paper are those of the author(s) and do not necessarily reflect the views of the NIH or the U.S. Department of Health and Human Services.

Appendix A

Examples

The following diagrams present examples of SBGN Activity Flow diagrams representing biological activities and their influences among each other in pathway networks. They by no means exhaust the possibilities of SBGN Activity Flow Level 1.

Figure A.1 presents an example of a signalling pathway involving the regulation of TGFβ-induced metastasis. The pathway was described in a report titled ”A Mutant-p53/Smad Complex Opposes p63 to Empower TGFβ-Induced Metastasis” in the 2009 April issue of Cell [5]. The figure shows the usage of *biological activity nodes*, *phenotype*, *positive influence* arc, *negative influence* arc, *necessary stimulation* arc, and *logic operator*.

Figure A.2 presents a more complicated example of signalling pathway involving the intracellular signalling through the epidermal growth factor receptor (EGFR). This example is a redraw of the Epidermal Growth Factor Receptor Pathway described in the Signal Transduction Knowledge Environment (`https://www.science.org/do/10.1126/resource.2375846/`

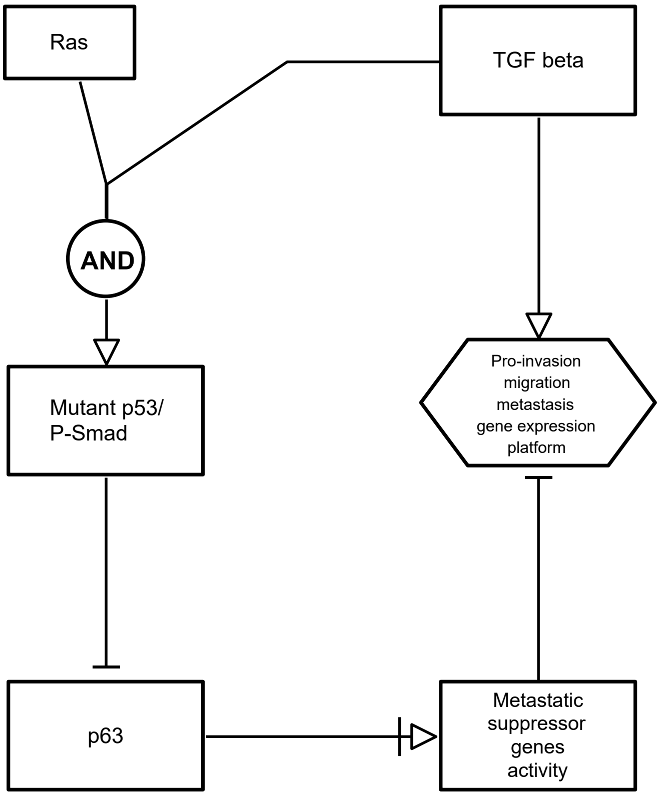


Figure A.1: *Regulation of TGFβ-induced metastasis.*

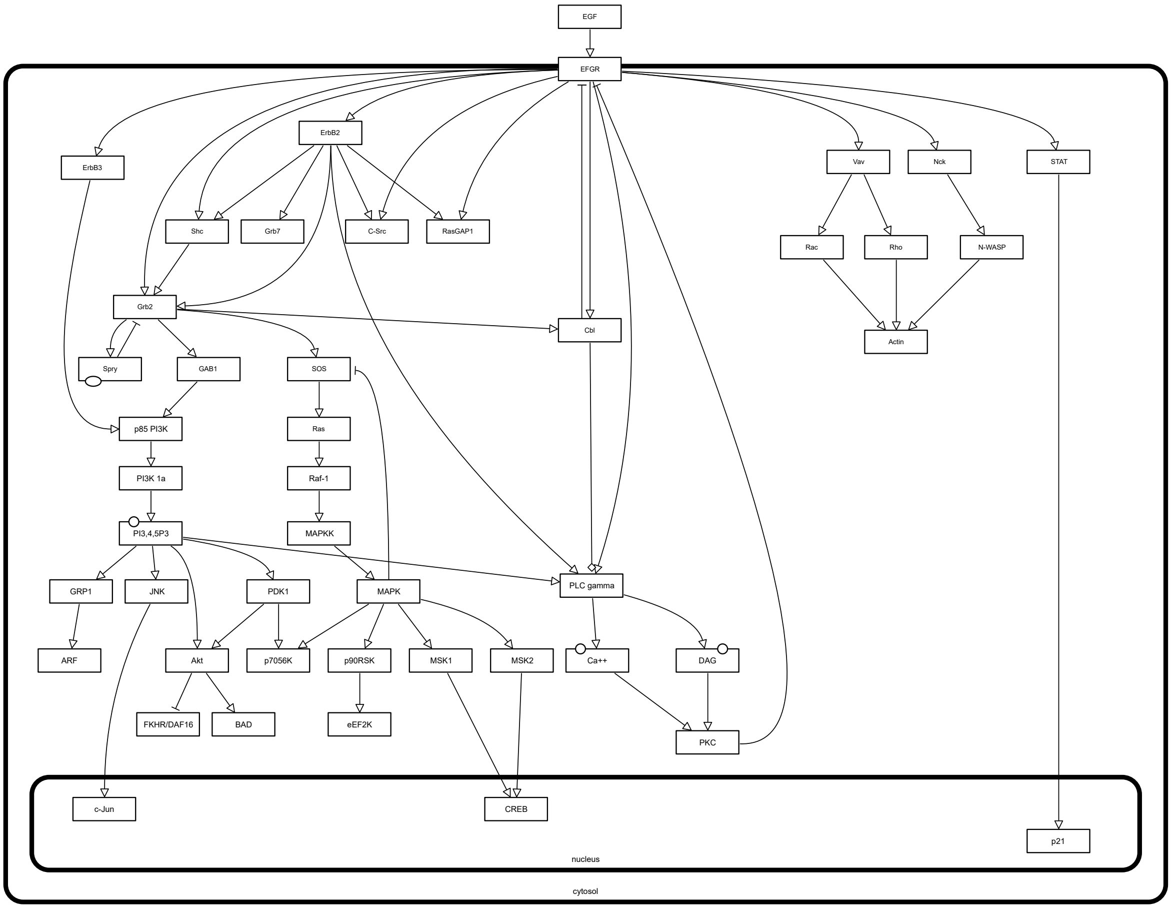


Figure A.2: *Epidermal Growth Factor Receptor Pathway.*

full/stke-1714066887583.pdf).

Figure A.3 shows the transforming growth factor beta (TGFβ) signalling pathway. The map is a redraw of the TRG-beta Signaling Pathway described in the PANTHER Pathway System (http://www.pantherdb.org/pathway/pathwayDiagram.jsp?catAccession=P00052) and is based on reviews by Massague [6] and Derynck [7].

Figure A.4 presents the simplest view of action potential propagation mediated by the voltage-gated sodium channels. There are two views of how voltage-gated ion channels are involved. The diagram on the left side shows that the *increase in membrane potential* activates *voltage-gated sodium channel activity*, which in turn triggers membrane *depolarisation*. The membrane depolarisation activates *voltage-gated potassium channel* activity, which results in repolarising the membrane potential. The diagram on the right side provides more detail in the mechanism. It shows that the *increase in membrane potential* first activates the *gating activity* of the channel, which in turn activates the *conductance activity* leading to the membrane *depolarisation*. In this case, both *gating activity* and *conductance activity* come from the sodium channel, which is indicated as the *unit of information*. In addition, this example also shows the advantage of using Activity Flow maps, because certain activities, such as gating and conductance, come from a few amino acids in particular three-dimensional structures that are not able to be illustrated in either Process Description or Entity Relationship maps.

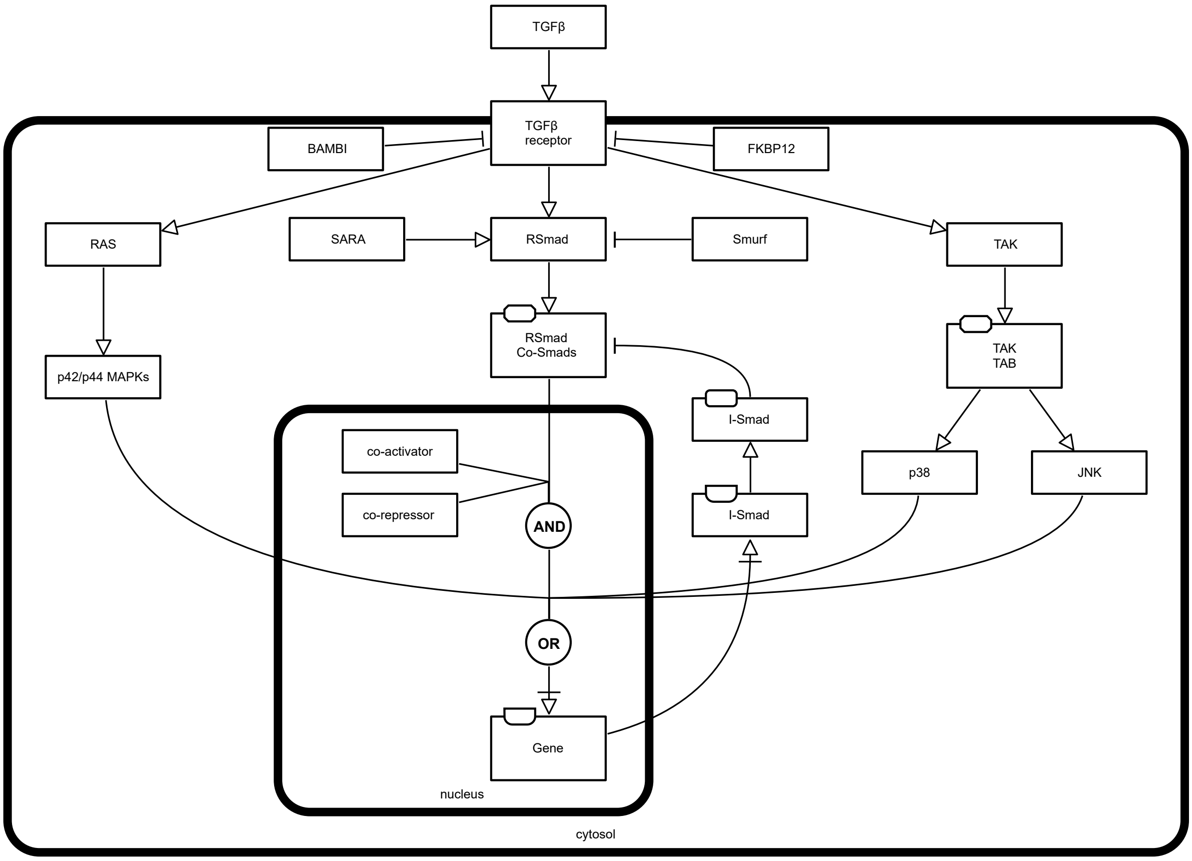


Figure A.3: *Transforming Growth Factor beta signalling pathway.*

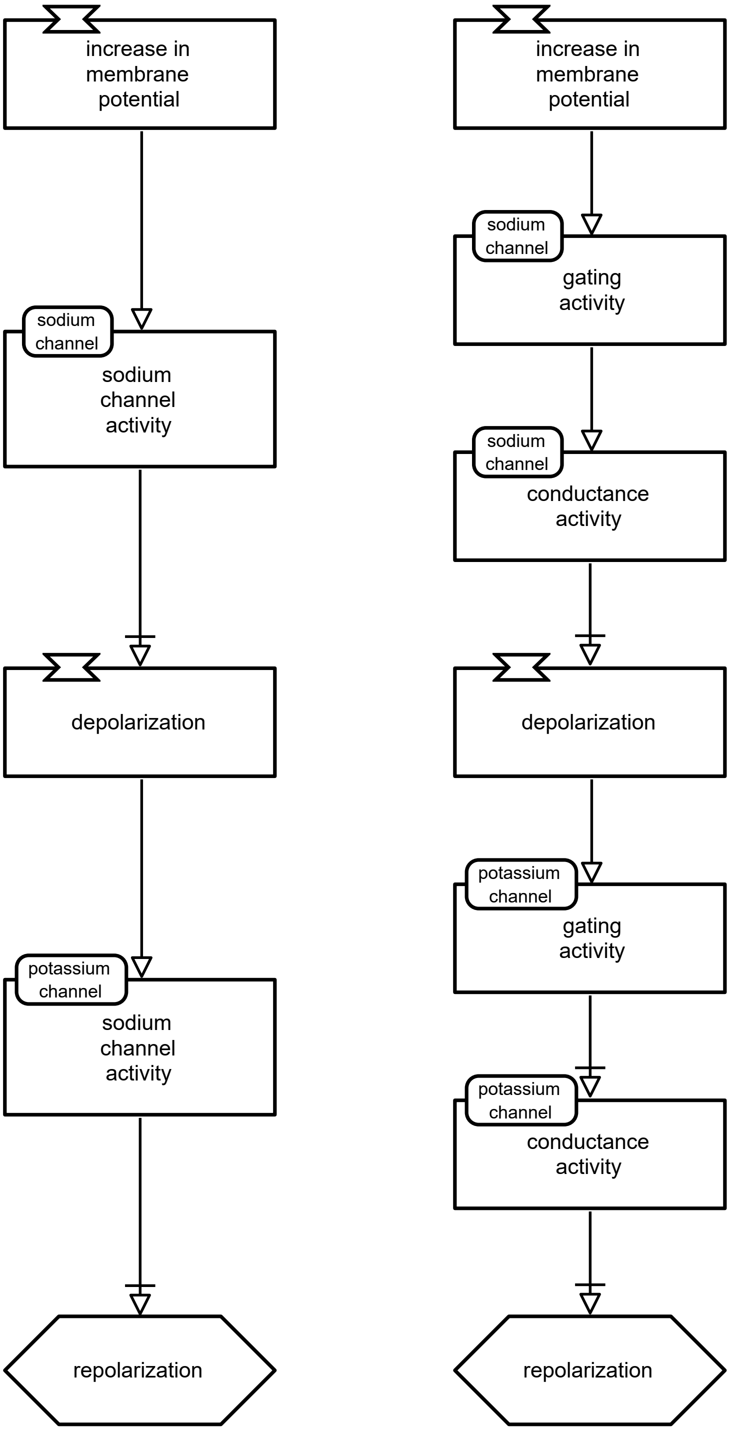
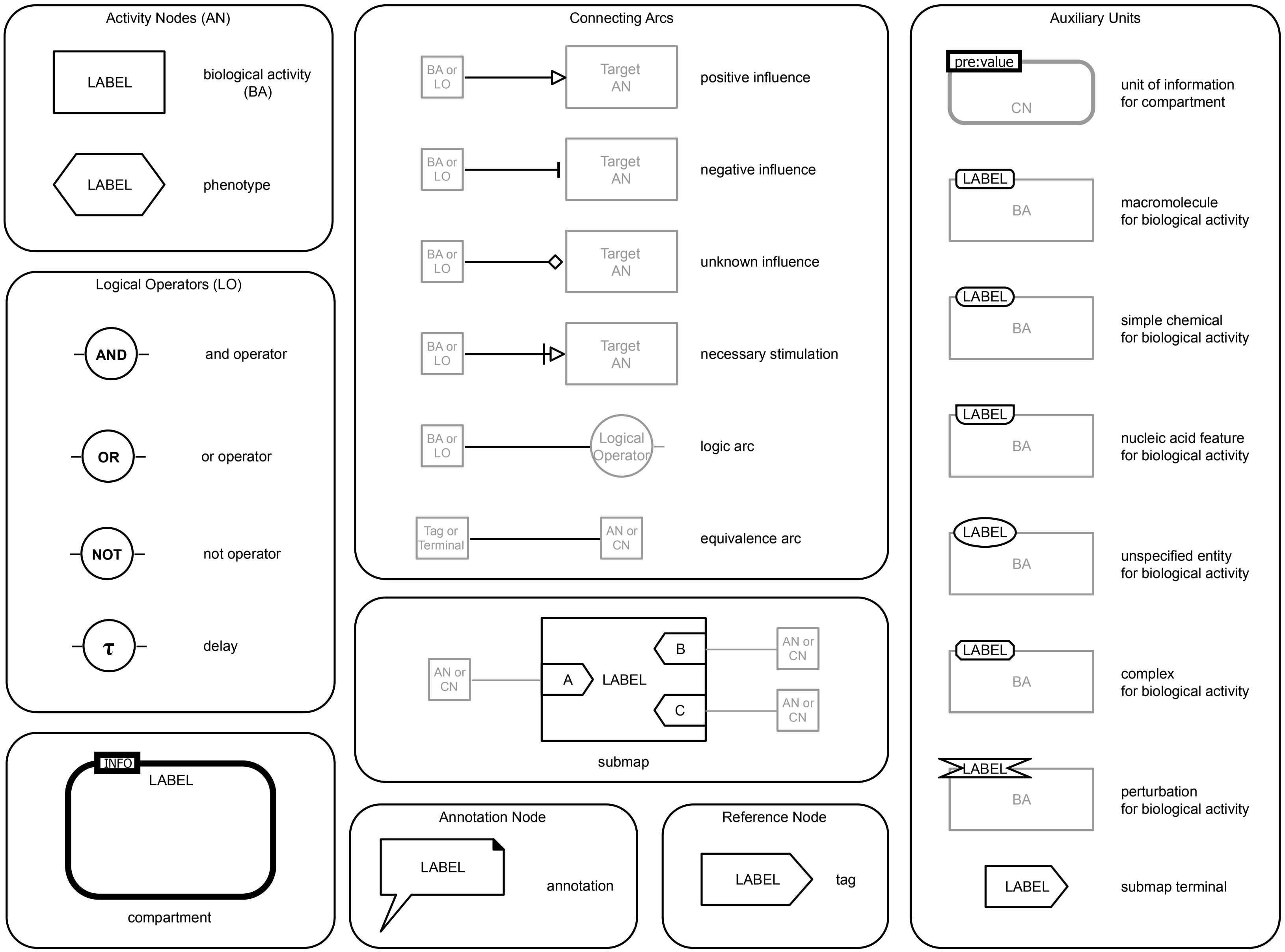


Figure A.4: *Two views of the role voltage-gated sodium channel plays in action potential generation illustrated by SBGN Activity Flow Level 1.*

Appendix B

Reference card

Print this summary of SBGN Activity Flow symbols for a quick reference.


Activity Nodes (AN)
LABEL
biological activity (BA)
LABEL
phenotype
Logical Operators (LO)
AND
and operator
OR
or operator
NOT
not operator
τ
delay
INFO
LABEL
compartment
Connecting Arcs
BA or LO
Target AN
positive influence
BA or LO
Target AN
negative influence
BA or LO
Target AN
unknown influence
BA or LO
Target AN
necessary stimulation
BA or LO
Logical Operator
logic arc
Tag or Terminal
AN or CN
equivalence arc
AN or CN
A
LABEL
B
C
AN or CN
AN or CN
submap
Annotation Node
LABEL
annotation
Reference Node
LABEL
tag
Auxiliary Units
pre:value
CN
unit of information for compartment
LABEL
BA
macromolecule for biological activity
LABEL
BA
simple chemical for biological activity
LABEL
BA
nucleic acid feature for biological activity
LABEL
BA
unspecified entity for biological activity
LABEL
BA
complex for biological activity
LABEL
BA
perturbation for biological activity
LABEL
submap terminal

Appendix C

Revision History

C.1 Version 1.0 to Version 1.1

There are three major changes in glyphs in Version 1.1 of the SBGN Activity Flow Level 1 specification.

1. Add *unit of information* to the compartment. This is done in accordance with SBGN Process Description Level 1 specification.

2. Remove *perturbation* from the activity node, and add it to the *unit of information* to decorate a *biological activity* node. The change was based on a survey conducted in August 2011.

3. Add *process* as an activity node. This is rather controversial at the beginning. The issue was to make a process both an origin and a target of arcs. It was not possible in AF V. 1.0 because a *phenotype* cannot be the origin of an arc. The details of this issue and subsequent survey (August 2011) are described on the SBGN website. The survey did not yield a concrete solution. Through an additional discussion on the SBGN discussion list, this seems to be the best solution for the purpose without violating semantic rules.

Version 1.1 was never officially released due to some disagreements in the change, especially in the use of process. It served as a working draft only.

C.2 Version 1.1 to Version 1.2

Below are the changes incorporated into Version 1.2 of the SBGN Activity Flow Level 1 specification.

Description	Tracker ID
Add *Unit of information for Compartment*, section 2.4.2	
Add *Unit of information for Compartment* to the ref card (Appendix B)	
Remove *perturbation* from Activity node section (2.3)	
Add *perturbation* to the unit of information section (2.4.1)	
Modify *perturbation* glyph in figures 2.1, 2.5, 2.17, A.4 and refcard accordingly (Appendix B).	
Add *process* to the Activity node section (2.3)	
Modify figure A4 by adding the *process* glyph.	
Modify the description for *biological activity* in section 2.3.1 to clarify that it is for molecular activity, so to differentiate it from the activities from *process*.	

continued on next page

continued from previous page

Description	Tracker ID
Modify the description for *phenotype* in section 2.3.3.	
Modify tables in section 3.3.1 and 3.3.2 - Remove *perturbation* - Add *process*	
Add *phenotype* as input for *equivalent arc* in section 3.3.1	
Remove the sentence ”If an activity has participants in at least two different compartments, the activity node has to be either in a compartment where the activity has at least one participant or in the empty space.” in section 4.2.1.8	
In section 4.2.1.8, change the first sentence ”If an activity has all participants ...” to ”If a network has all participants....”	
In section 4.2.1.8, add a sentence ”Edges/arcs are allowed to cross the compartment boundaries when the input and output ANs are in two different compartments.”	
Add *delay* in the line of Logical operators in Table 2.1	
Correct figure legend for Figure 2.10. The example on the right side shows an incorrect representation.	
In section 2.2, controlled vocabularies, clarify that the terms are to describe the unit of information of the ANs	
Rewrite figure legends with more details throughout section 2	
Modify Figure A3 in Appendix A (TGF beta signaling pathway) by changing the *Gene transcription* phenotype node to a biological activity node. However, it should be discussed whether phenotype can be an input or not.	2992312
Update SBO term (SBO:0000255 to SBO:0000002) in section 2.2.3	3068940

C.3 Version 1.2 to Version 1.3

Below are the changes incorporated into Version 1.3 of the SBGN Activity Flow Level 1 specification.

Description	Tracker ID
Added Section 1.5 about the typographical conventions and requirement levels and moved requirement levels in Section 4.1 to Section 1.5 to make these levels valid throughout the document	#5
Updated the document to accurately reflect the requirement levels (must vs should)	#6
Moved first paragraph of Chapter 2 that explains the structure of the document to Section 1.6 of Chapter 1	#7
Added sections for *tag* and *submap terminal* glyphs (Section 2.10.1 and 2.4.3, respectively)	#8 #9
Reorganised the document structure to make it compatible with PD Spec L1V2.1	#10 #23
Reviewed all figures and the reference card and made improvements and corrections.	#11 #12 #16
Changed section titles “Syntax” → “Connectivity and containment” and “Semantic rules” → “Glyph specific rules” to remove the syntactic and semantic distinction in the rules	#10

continued on next page

continued from previous page

Description	Tracker ID
Updated containment table in Section 3.3.2 to indicate containment of compartments by other compartments, also added submap column and submap terminal row	#15
Fixed the broken hyperlinks throughout the document	#18
Added new row for Biological activity and LO abbreviation to Logical operators row in Section 2.1	#22
Updated the format used in sections for logical operator glyphs to match the PD Spec L1V1.2. Updated logical operator definitions and added examples to *and*, *not* and *delay* glyphs	#13 #35 #36
Added example for glyph *unknown influence* in Section 2.6.3	#12
Updated node connectivity table in Section 3.3.1 to fix some connections	#14
Changed notation to use *Origin* and *Target* in glyph explanations and used *Source (S)* and *Target (T)* in node connectivity table	#19
Added missing subsections to Section 2.3.1, 2.3.2, 2.5, 2.9.1 and 2.11	#24
Changed Section 2.4.1 title from "Glyph: Unit of information for Biological activity" to "Glyphs: Units of information for Biological activity". Also changed naming of units of information (e.g., unit of information of a macromolecule to macromolecule unit of information) together with Figure 2.5	#29
Refined definitions of *positive influence*, *negative influence* and *necessary stimulation* glyphs in Section 2.6.1, 2.6.2 and 2.6.4	#31 #33 #34
Section 2.5 "Container nodes" is replaced by subsection 2.5.1 "Glyph: Compartment" and now Section 2.5 is "Glyph: Compartment".	#30
Renamed all occurrences of "modulations" to "influences" including section title 2.8 Modulation arcs (in L1V1.2) → 2.6 Influence arcs (in L1V1.3)	#27
Fixed typos and modified visual style of some tables/figures throughout the document	

Bibliography

[1] Nicolas Le Novère, Michael Hucka, Huaiyu Mi, Stuart Moodie, Falk Schreiber, Anatoly Sorokin, Emek Demir, Katja Wegner, Mirit I. Aladjem, Sarala M. Wimalaratne, Frank T. Bergman, Ralph Gauges, Peter Ghazal, Hideya Kawaji, Lu Li, Yukiko Matsuoka, Alice Villéger, Sarah E. Boyd, Laurence Calzone, Melanie Courtot, Ugur Dogrusoz, Tom C. Freeman, Akira Funahashi, Samik Ghosh, Akiya Jouraku, Sohyoung Kim, Fedor Kolpakov, Augustin Luna, Sven Sahle, Esther Schmidt, Steven Watterson, Guanming Wu, Igor Goryanin, Douglas B. Kell, Chris Sander, Herbert Sauro, Jacky L. Snoep, Kurt Kohn, and Hiroaki Kitano. The systems biology graphical notation. *Nat Biotechnol*, 27(8):735–41, 8 2009.

[2] D. Pan, M. Fujimoto, A. Lopes, and Y. Wang. Twist-1 is a PPARδ-inducible, negative-feedback regulator of PGC-1α in brown fat metabolism. *Cell*, 137(1):73–86, 2009.

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

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

[5] M. Adorno, M. Cordenonsi, M. Montagner, S. Dupont, C. Wong, B. Hann, A. Solari, S. Bobisse, M. B. Rondina, V. Guzzardo, A. R. Parenti, A. Rosato, S. Bicciato, A. Balmain, and S. Piccolo. A mutant-p53/smad complex opposes p63 to empower TGFbeta-induced metastasis. *Cell*, 137(1):87–98, 2009.

[6] J. Massague. TGF-beta signal transduction. *Annual Review Biochemistry*, 67:753–791, 1998.

[7] R. Derynck, R. J. Akhurst, and A. Balmain. TGF-beta signaling in tumor suppression and cancer progression. *Nature Genetics*, 29(2):117–129, 2001.