SBGN PD

Current status and
Draft L1 V2.0 Specification

Stuart Moodie, Eight Pillars
SBGN-10, LA, 16/8/14

PD Specification

- Level 1 Version 1: Sep 2008
  - Original specification
- Level 1 Version 1.1: Sep 2009
  - Fixed inconsistencies
  - Typos
  - Clarified “ontology” of glyphs
  - Nomenclature changes for consistency with other languages
- Level 1 Version 1.2: Oct 2010
- Level 1 Version 1.3: Feb 2011
  - Minor changes of clarity
  - Document errors
- Draft – RFC. Now.

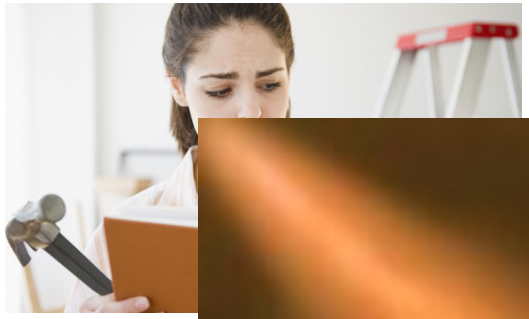
Purpose of SBGN

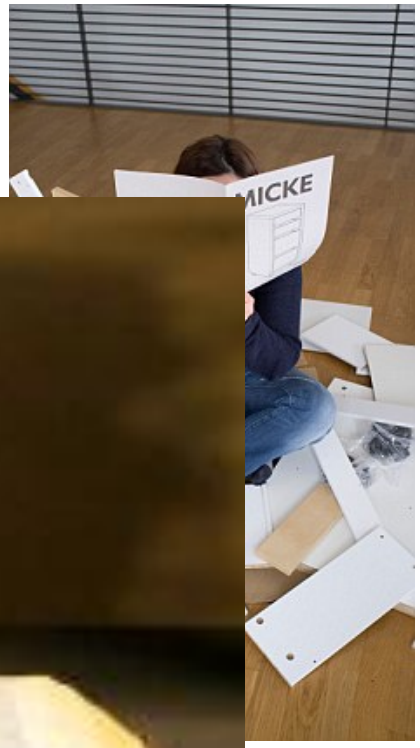
- NOT A Modelling **Language**
  - However, it has a **model** or underlying abstraction
- Unambiguous visual language!!!
  - Reader understands exactly what is meant by author **without** additional textual description, e.g. Caption
- Knowledge capture
- Information exchange
  - publications rather than DBs

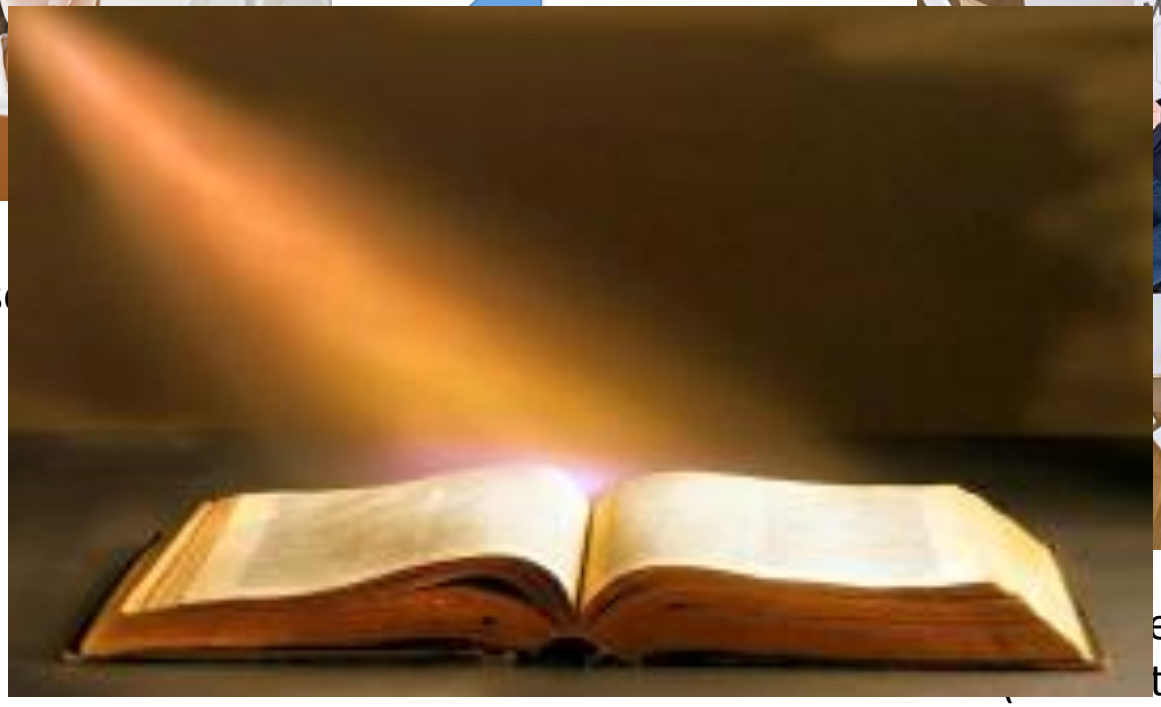
What's in Version 2.0

- New/Modified Glyphs
- Resolving semantic "issues"
- Improve organisation/clarity of the spec
- Enumerated rules *

Planned Specifications







Use

ecification

tive)

SBGN “Bible”? : Merge User Manual + Reference Specification

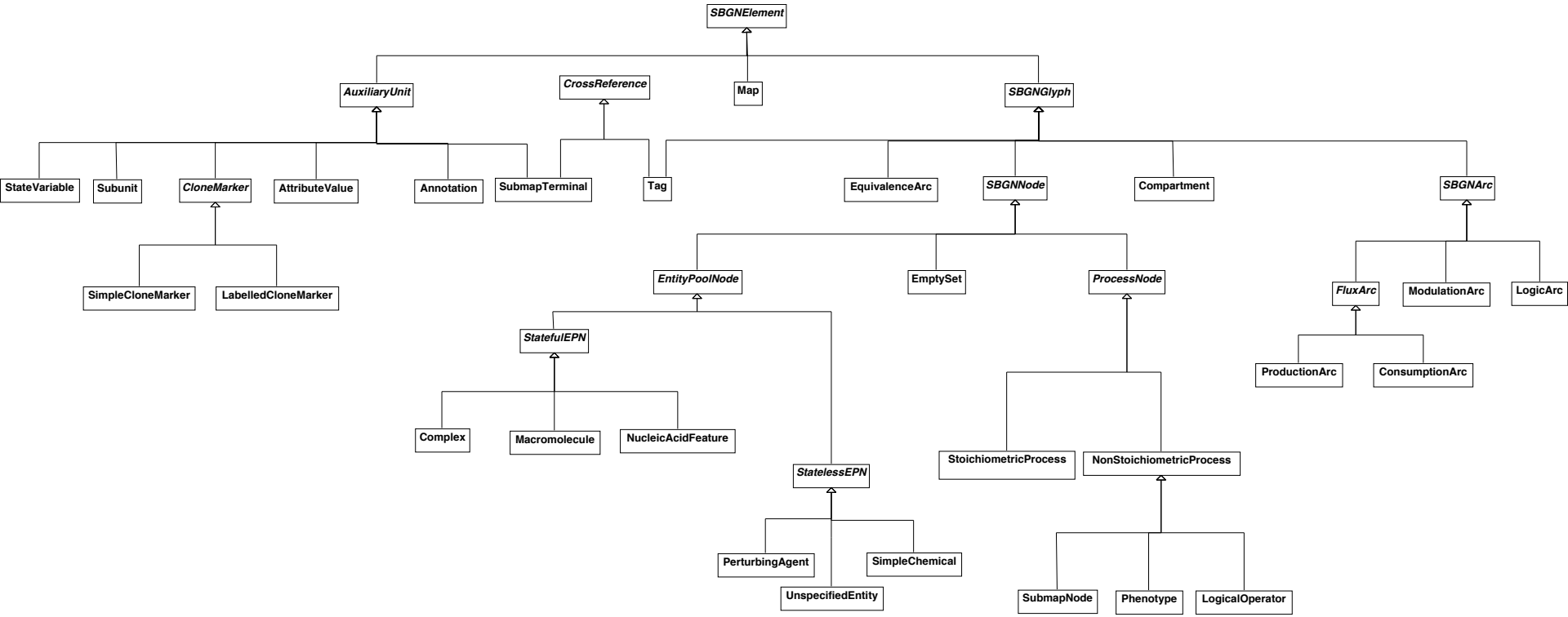
Level 1 Version 2.0: Spec

- Major rewrite
  - Based on SBML spec
  - Based on UML spec

- Aim
  - Remove ambiguities that exist in current spec
  - Remove inconsistencies
  - “What does the spec say?”

Contents

UML Model


SBGNElement
AuxiliaryUnit
CrossReference
Map
SBGNGlyph
StateVariable
Subunit
CloneMarker
AttributeValue
Annotation
SubmapTerminal
Tag
EquivalenceArc
SBGNNode
Compartment
SBGNArc
SimpleCloneMarker
LabelledCloneMarker
EntityPoolNode
EmptySet
ProcessNode
FluxArc
ModulationArc
LogicArc
StatefulEPN
ProductionArc
ConsumptionArc
Complex
Macromolecule
NucleicAcidFeature
StatelessEPN
StoichiometricProcess
NonStoichiometricProcess
PerturbingAgent
SimpleChemical
UnspecifiedEntity
SubmapNode
Phenotype
LogicalOperator

Attributes

name: string **(O)** The name that of the state variable. This is optional, but if defined cannot be an empty string or just white space characters. It should also start with an alpha-numeric character and end with a non-space character. It should not contain a '@' character[9].

⚠ See footnote.

Associations

owning_entity_type:EntityType (1) The EntityType that owns this definition.

Rule and Constraints

None.

Changes from Previous Version

⚠ See footnote.

Although not defined explicitly in the previous version, arguably this concept did exist in the language.

3.5.9 EntityPoolNode

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

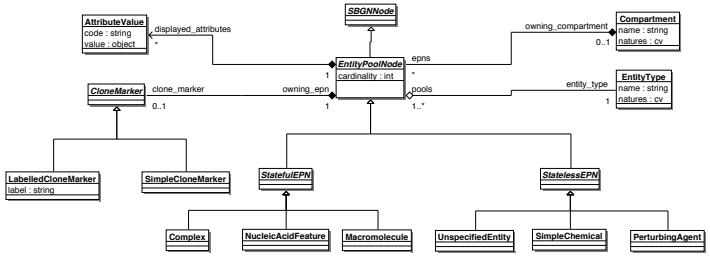


Figure 3.8: UML definition of the entity pool node and its descendant glyphs.

The EntityPoolNode (figure 3.8) is the definition of the entity pool and it shares an EntityType with other identical entities[10]. An instance of an entity pools is therefore distinguished from other pools with the same entity type by its cardinality, its owning_compartment and the values of its StateVariables (where appropriate). It must belong to a compartment or be associated with the map (c.f. section 3.5.26) and can contain a clone marker if it is cloned (see section 3.7)— note that not all EPNs can be cloned.

⚠ See footnote.

[9] No rule defined previously, but this would seem to make sense.

[10] Although this concept is discussed it is not explicitly defined previously.

Generalisation

- SBGNNode (see section 3.5.5)

Attributes

cardinality: int **(R)** The number of copies of the entity. Must be a positive non-zero integer.

Associations

owning_compartment:Compartment (0..1) The compartment that this EPN belongs too.
entity_type:EntityType (1) The type of this entity pool.
clone_marker:CloneMarker (0..1) The clone marker decorator. See section 3.5.37 for its use.
displayed_attributes:AttributeValue (*) One or more decorators used to display attribute values[11].

Unique Key

- owning_compartment
- entity_type
- cardinality

Rules and Constraints

- If cardinality > 1 then the descendant glyph must be displayed as a multimer.
- If the EPN is drawn directly on a *Map* then owning_compartment is not set. We interpret this as belonging to an invisible default compartment.
- natures can only use the material type (section 3.6.1), conceptual type (section 3.6.2) or physical characteristics (section 3.6.4) controlled vocabularies.
- The appropriate subclass of CloneMarker must be used to distinguish logically identical instances of this class.
- All StateVariableDefinitions associated with the EntityType must have an associated StateVariable.

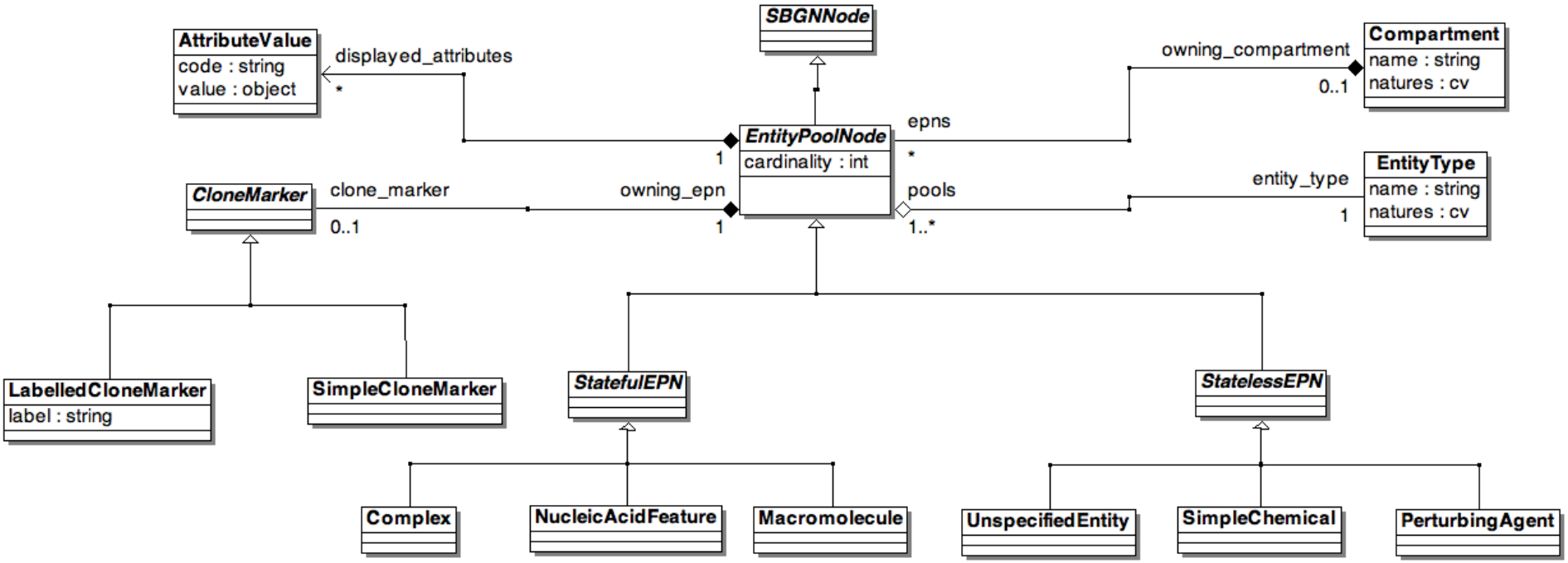
Changes from Previous Version

Not defined explicitly in the previous version, but the concept of the EPN and its semantics existed. The main change to previous semantics is that of the natures, which didn't formally exist before, but which now must contain a unique set of controlled vocabularies and is part of the logical key of the EntityPoolNode.

3.5.10 Empty Set

It is useful to have the ability to represent the creation of an entity or a state from an unspecified source, that is, from something that one does not need or wish to make precise. For instance, in a model where the production of a protein is represented, it may not be desirable to represent all

[11] This is an alternate way of using the Unit of Information to display information, but to constrain it so that it presents attributes of the EPN not general annotation. See the AttributeValue class for more information.


SBGNNode
AttributeValue
code : string
value : object
displayed_attributes
*
Compartment
name : string
natures : cv
owning_compartment
0..1
EntityPoolNode
cardinality : int
1
epns
*
CloneMarker
clone_marker
0..1
owning_epn
1
pools
1..*
entity_type
1
EntityType
name : string
natures : cv
LabelledCloneMarker
label : string
SimpleCloneMarker
StatefulEPN
StatelessEPN
Complex
NucleicAcidFeature
Macromolecule
UnspecifiedEntity
SimpleChemical
PerturbingAgent

Rules and Constraints

- State variables do not need to be logically unique, therefore two or more state variables with the same name are permitted.
- The LabelledCloneMarker must be used to indicate cloning for instances of StatefulEPN and its subclasses, with a must use the same

Changes from Previous Version

Not defined explicitly in the previous version.

Macromolecule

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

Generalisation

- StatefulEPN (see section 5.5)

Attributes

No additional attributes.

Associations

No additional associations.

Rules and Constraints

No additional rules and constraints.

Notation

There are two glyphs associated with Macromolecule. The first *Macromolecule monomer* is used when cardinality = 1 and the second *Macromolecule multimer* is used when cardinality > 1.

Glyph: ***Macromolecule monomer***
SBO Term: SBO:0000245 ! macromolecule
Container: A macromolecule is represented by a rectangular container with rounded corners, as illustrated in Figure 5.17.
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.

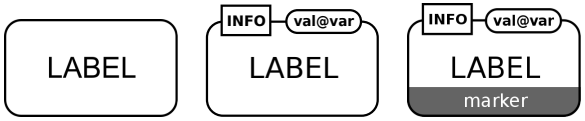


Figure 5.17: The Process Description glyph for *macromolecule*, shown plain and unadorned on the left, and with an additional state variable and a unit of information in the middle and the cloned form on the right.

Glyph: ***Macromolecule multimer***
SBO Term: SBO:0000420 ! multimer of macromolecules
Container: A *multimer* is represented by two identical containers shifted horizontally and vertically and stacked one on top of the other. Figure 5.18 illustrates the glyph.
Label: As monomer

Figure 5.18: The Process Description glyph for *macromolecule multimer*, shown plain and unadorned on the left, and with an additional state variable and a unit of information in the right and the cloned form on the right.

Usage Examples In this section, we provide examples of Entity Pool Node representations drawn using the SBGN Process Description Level 1 glyphs described above.

Figure 5.19 represents calcium/calmodulin kinase II, with phosphorylation on the sites threonine 286 and 306, as well as catalytic and autoinhibitory domains. Note the use of *units of information* and *state variables*.

Issues to take note of

3.5.7 EntityType

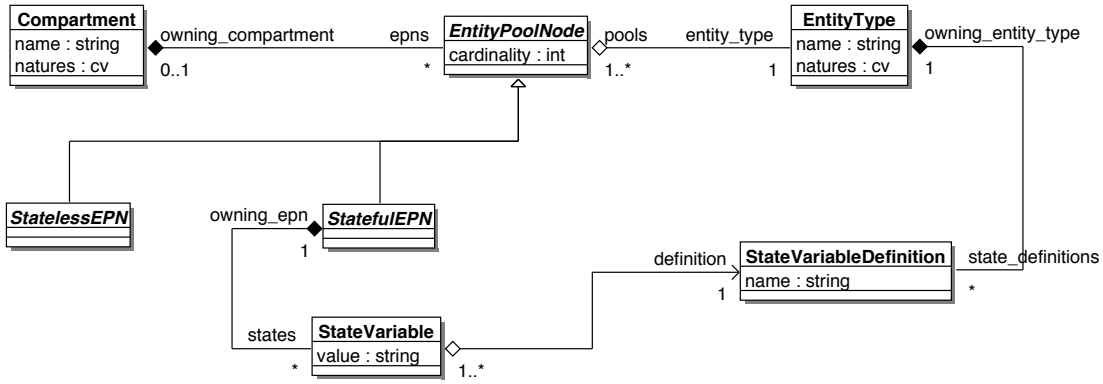


Figure 3.7: UML definition of the entity type and the state variable definition. The diagram shows how these classes interact with the entity pool, state variable and so influence EntityPoolNode logical identity.

⚠ See footnote.

The EntityType[5] defines the type of entity instantiated by one or more Entity Pools in a Process Description map. The EntityType has associated state variable definitions (see figure 3.7) and this enforces one of the core rules in the Process Description language that once a state is associated with an entity type it must be used by all entity pools of that type.

Generalisation

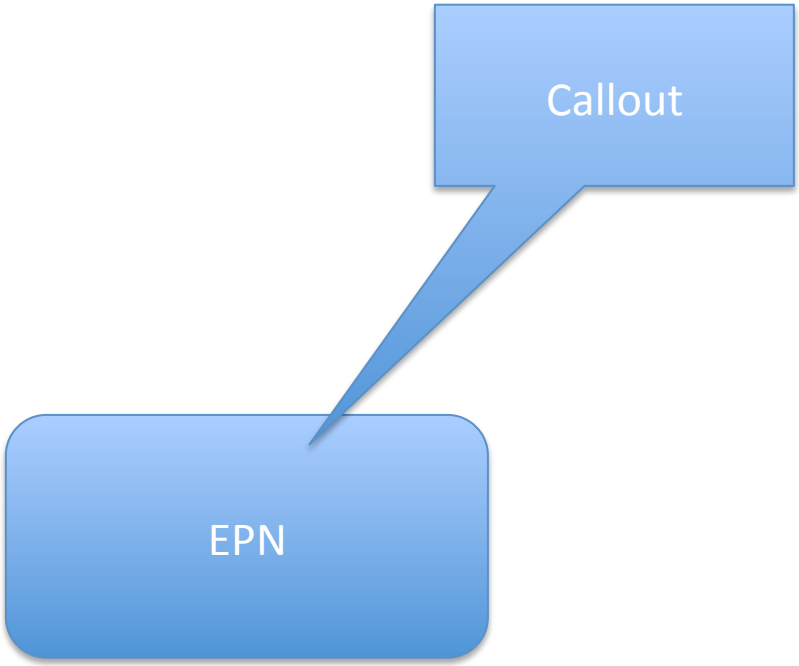
None.

[5]The concept of an entity pool's type has been there implicitly in previous versions and comes up in discussions. This class aims to formalise that concept and the rules associated with it and enable us to formalise rules associated with EPNs.

Highlights of what to look out for

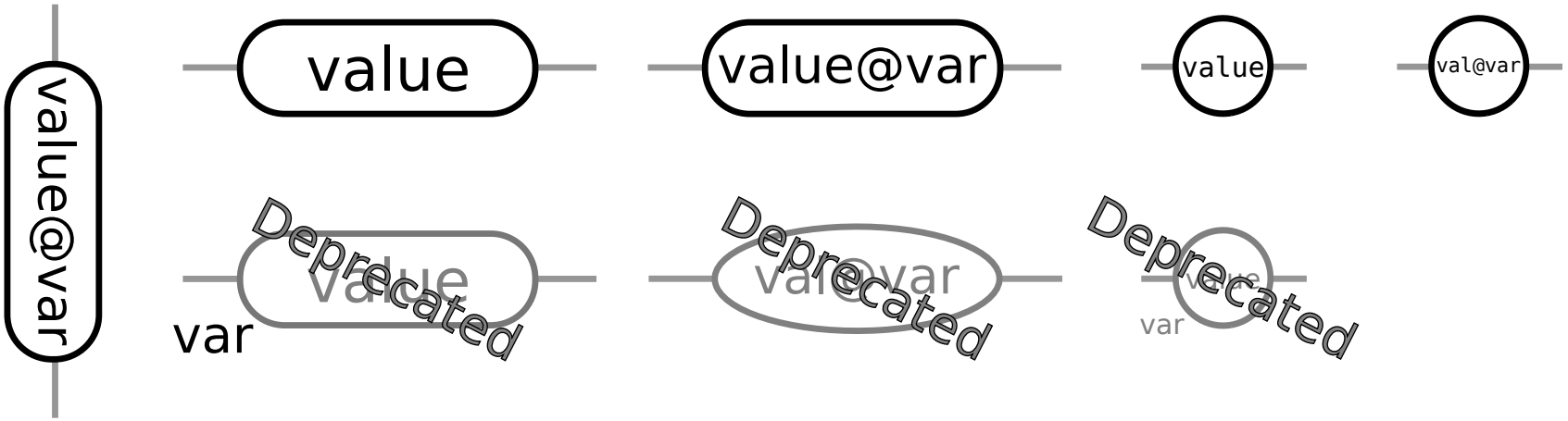
WHAT'S NEW IN 2?

New Glyph: Annotation



Modified Glyphs

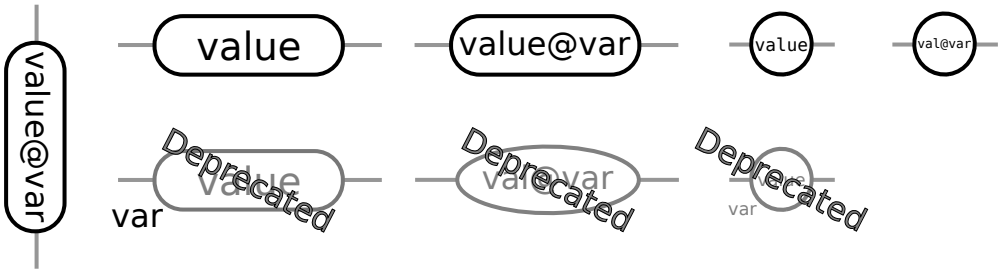
State Variable: Stadium symbol or circle



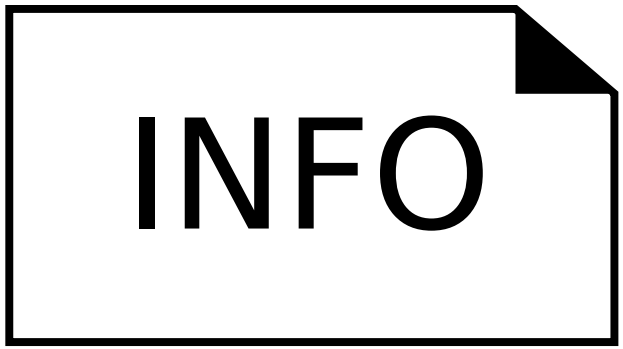
Simple Chemical: Stadium symbol or circle

Glyphs

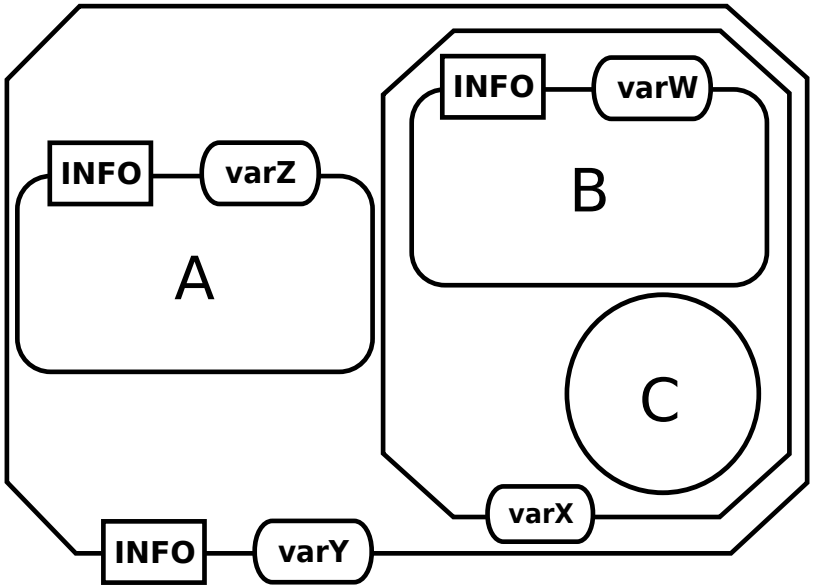
State Glyph Change



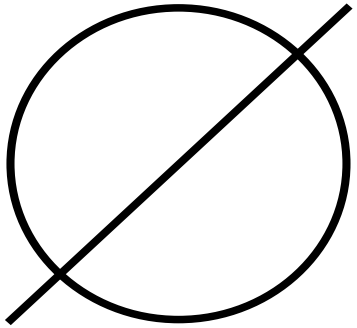
New Annotation Glyph



Complex Subunits are Decorators

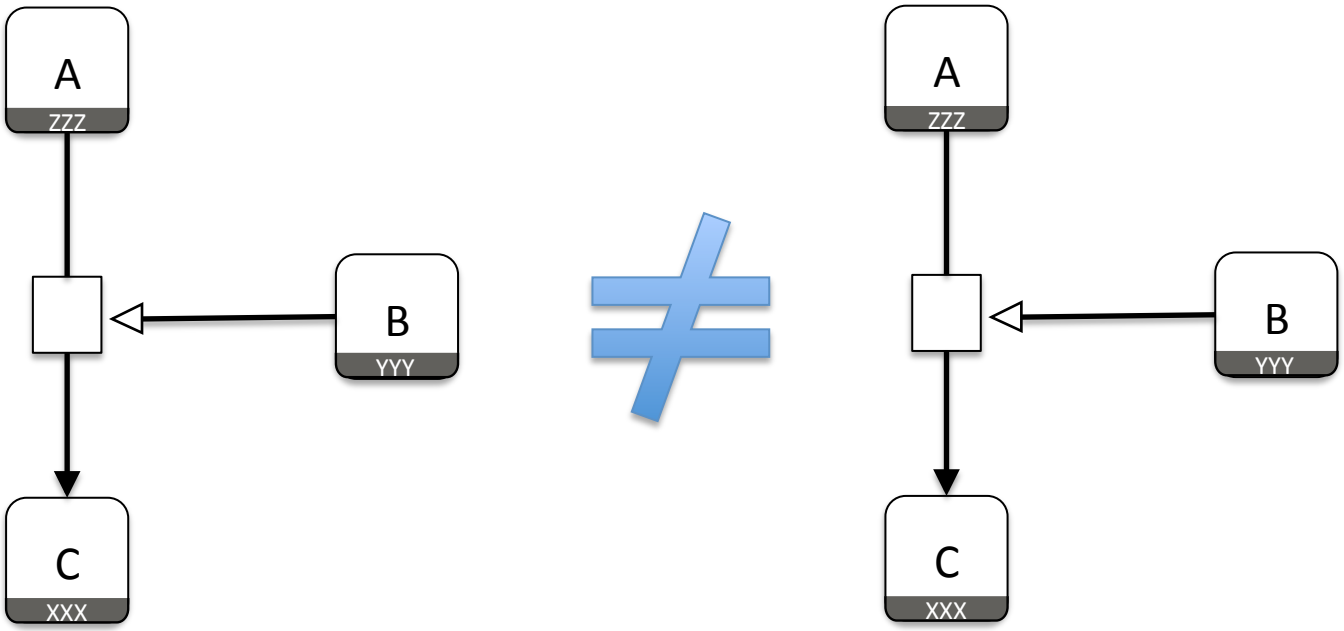


Empty Set



Clarified Process Semantics

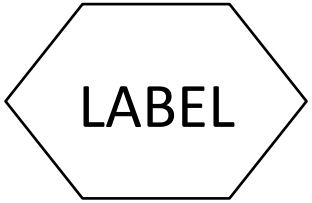
Process Duplication



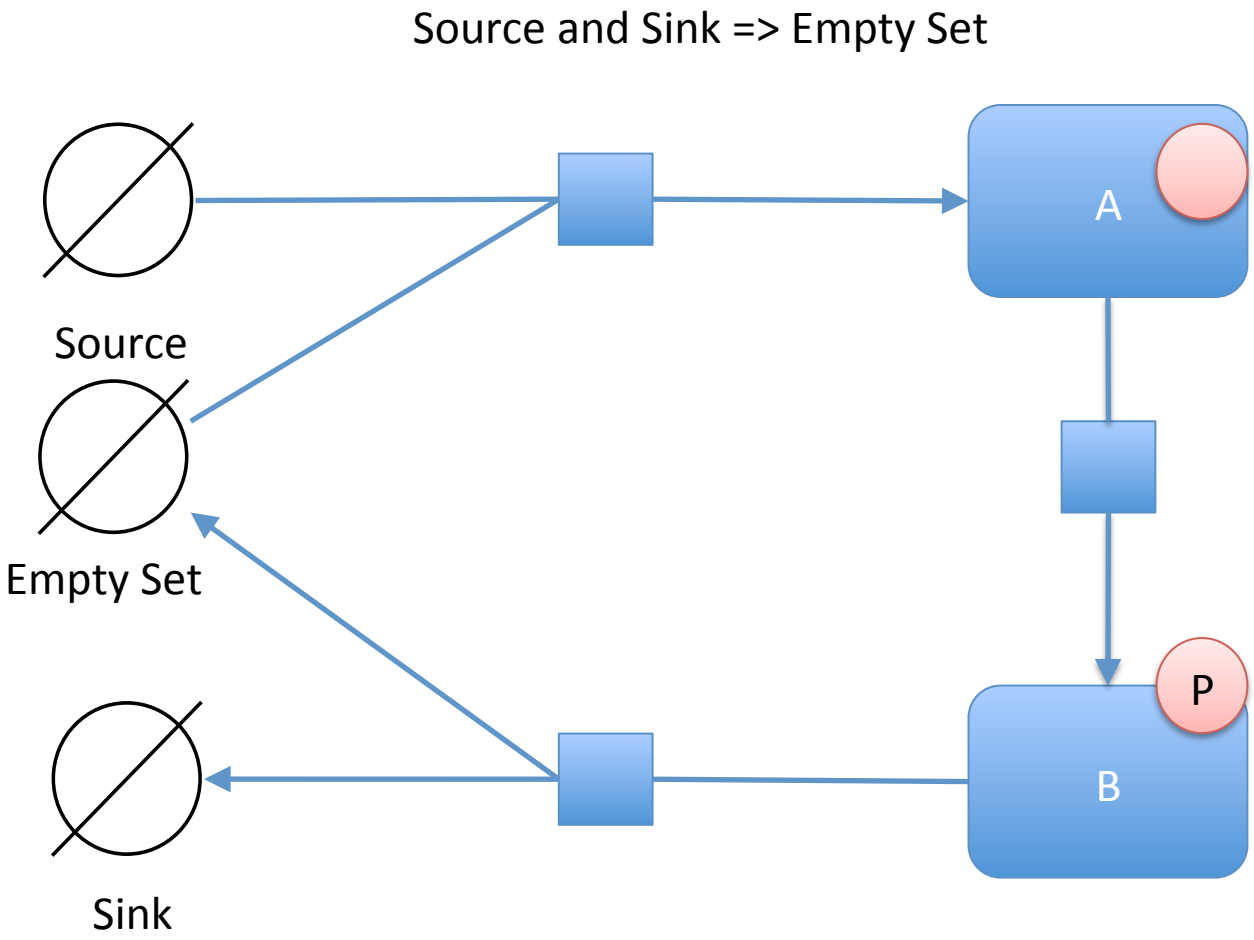
Fixed Phenotype Inconsistency

Phenotype is not an EPN

Phenotype cannot be duplicated



Changed Glyph and Semantics!

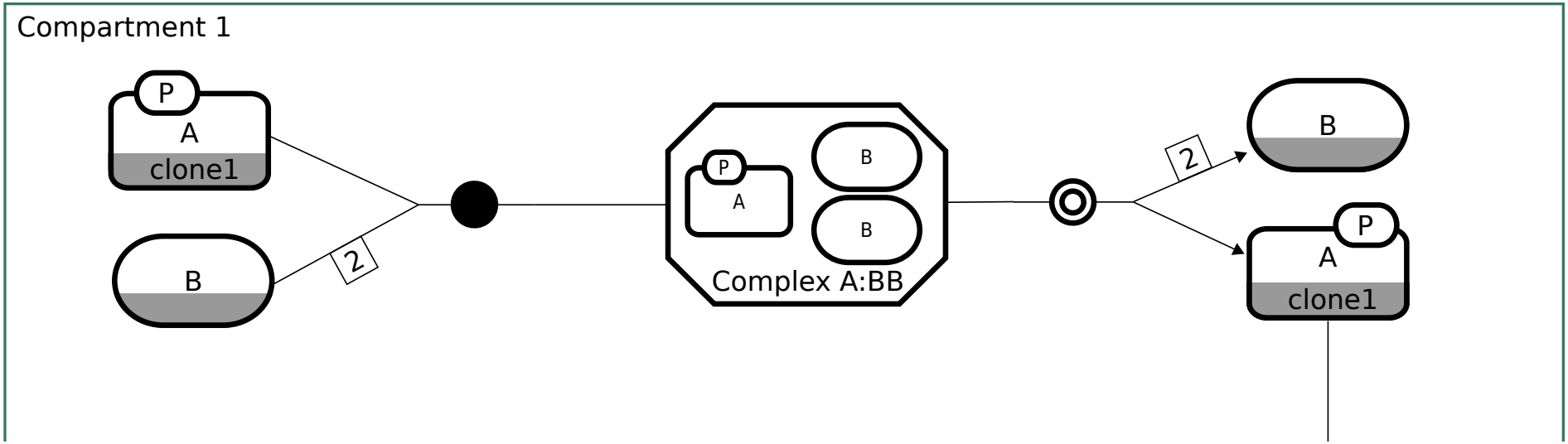


- No longer an EPN
- Rule of degree 1 dropped.

Uniqueness and Semantics

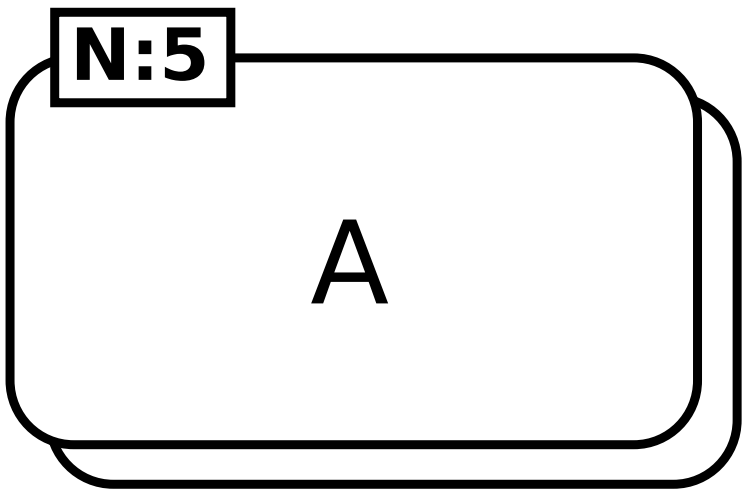
WHEN IS THE SAME THING?

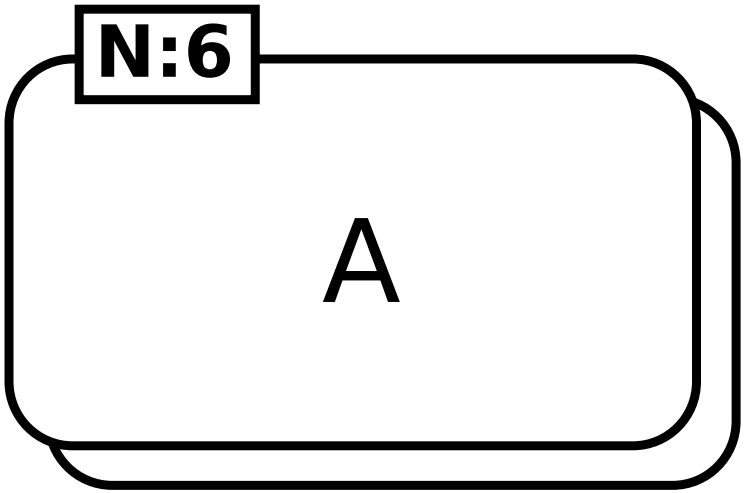
Why it's important.



- Distinct glyphs represent A: **Instance Identity**
- Both "A" glyphs represent the same thing: **Logically Unique**

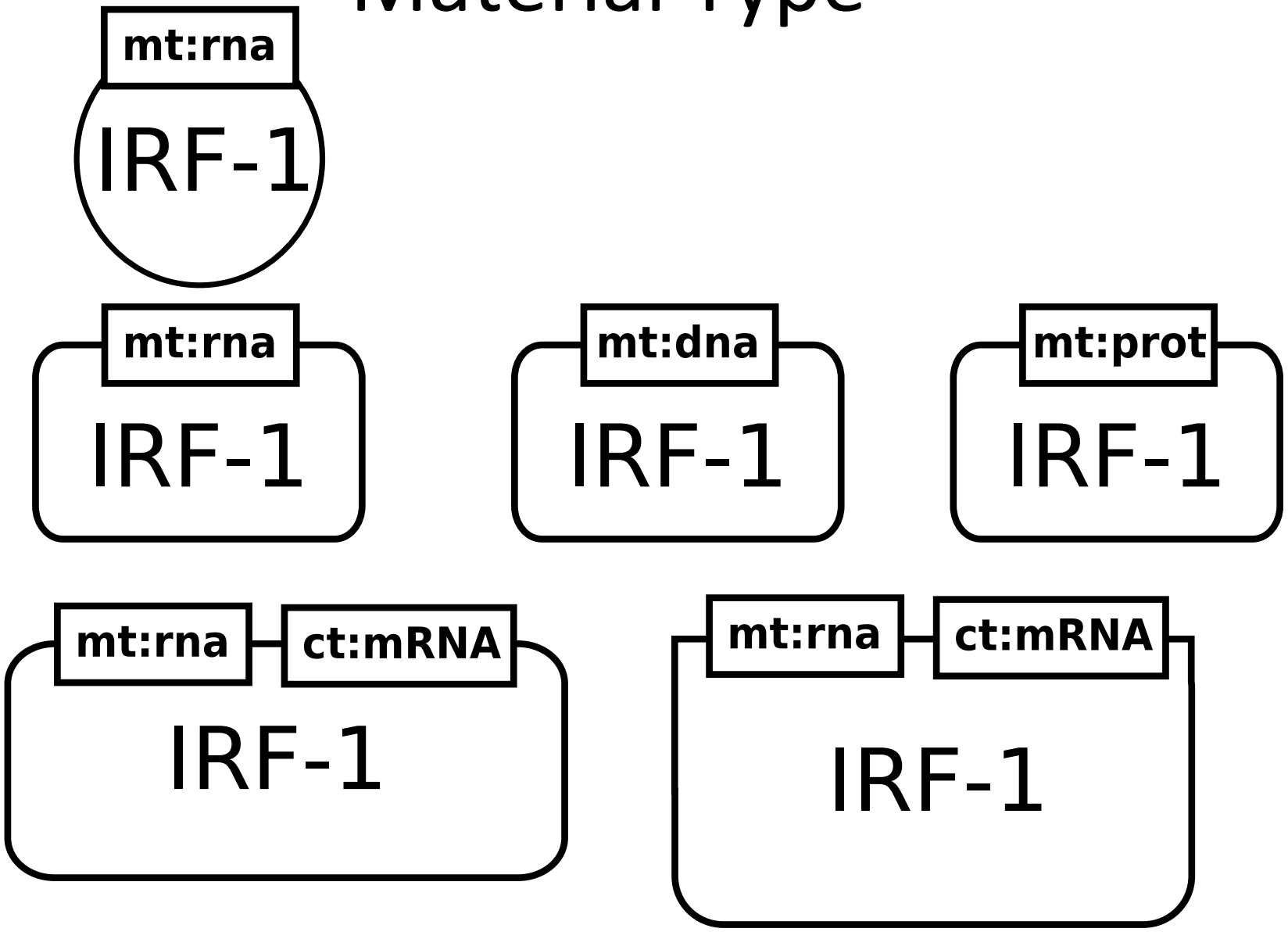
Cardinality





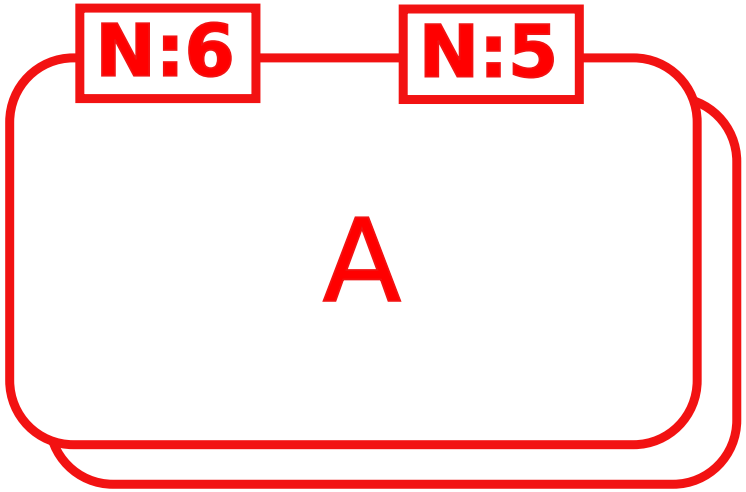
N == 1

Material Type



Unit of Information

- Glyph has multiple uses:
  - Decorator for cardinality
  - Decorator for material type
  - Place for annotation



Entity Type

3.5.7 EntityType

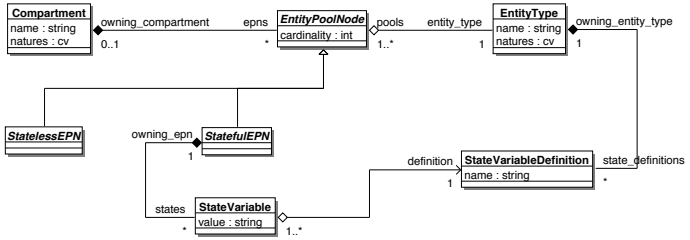


Figure 3.7: UML definition of the entity type and the state variable definition. The diagram shows how these classes interact with the entity pool, state variable and so influence EntityPoolNode logical identity.

⚠ See footnote.

The EntityType[5] defines the type of entity instantiated by one or more Entity Pools in a Process Description map. The EntityType has associated state variable definitions (see figure 3.7) and this enforces one of the core rules in the Process Description language that once a state is associated with an entity type it must be used by all entity pools of that type.

⚠ See footnote.

Generalisation

None.

[5]The concept of an entity pool's type has been there implicitly in previous versions and comes up in discussions. This class aims to formalise that concept and the rules associated with it and enable us to formalise rules associated with EPNs.

Attributes

name: string **(R)** The name that identifies the entity in the Process Description map. EPNs with the same label should be from the same entity. The string cannot be empty and must start and end with a non-space character. Any Unicode character is acceptable[6].

natures: cv(*) The nature of the entity pool node as defined by a controlled vocabulary. Zero, one or more values may be set, but each one must belong to a different controlled vocabulary (see section 3.6)[7].

Associations

state_definitions:StateVariableDefintion (*) The state definitions associated with this type.

pools:EntityPoolNode (1..*) The entity pool nodes that used this type.

Unique Key

- name
- natures

Rule and Constraints

- All instances of EntityPoolNode associated with a particular EntityType must be of the same class.
- If an instance of EntityType contains one or more instances of StateVariableDefinition then the EntityPoolNodes associated with it must be subclasses of StatefulEPN.

Notation

Although there is no direct graphical representation of this class, other classes that have a composite aggregation with it will need to represented the natures of the type graphically. Because this attribute is of class AttributeValue each nature is shown as a separate *Unit of Information*.

Changes from Previous Version

Although not defined explicitly in the previous version, this concept and the associated rules did exist in the language.

Entity Pool Node

3.5.9 EntityPoolNode

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

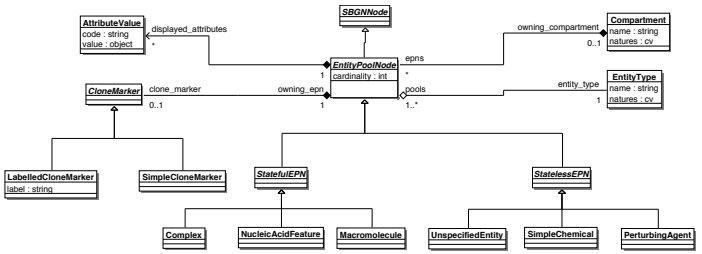


Figure 3.8: UML definition of the entity pool node and its descendant glyphs.

⚠ See footnote.

The EntityPoolNode (figure 3.8) is the definition of the entity pool and it shares an EntityType with other identical entities[10]. An instance of an entity pools is therefore distinguished from other pools with the same entity type by its cardinality, its owning_compartment and the values of its StateVariables (where appropriate). It must belong to a compartment or be associated with the map (c.f. section 3.5.26) and can contain a clone marker if it is cloned (see section 3.7)— note that not all EPNs can be cloned.

[9] No rule defined previously, but this would seem to make sense.
[10] Although this concept is discussed it is not explicitly defined previously.

Generalisation

- SBGNNode (see section 3.5.5)

Attributes

cardinality: int **(R)** The number of copies of the entity. Must be a positive non-zero integer.

Associations

owning_compartment:Compartment (0..1) The compartment that this EPN belongs too.
entity_type:EntityType (1) The type of this entity pool.
clone_marker:CloneMarker (0..1) The clone marker decorator. See section 3.5.37 for its use.
displayed_attributes:AttributeValue (*) One or more decorators used to display attribute values[11].

Unique Key

- owning_compartment
- entity_type
- cardinality

Rules and Constraints

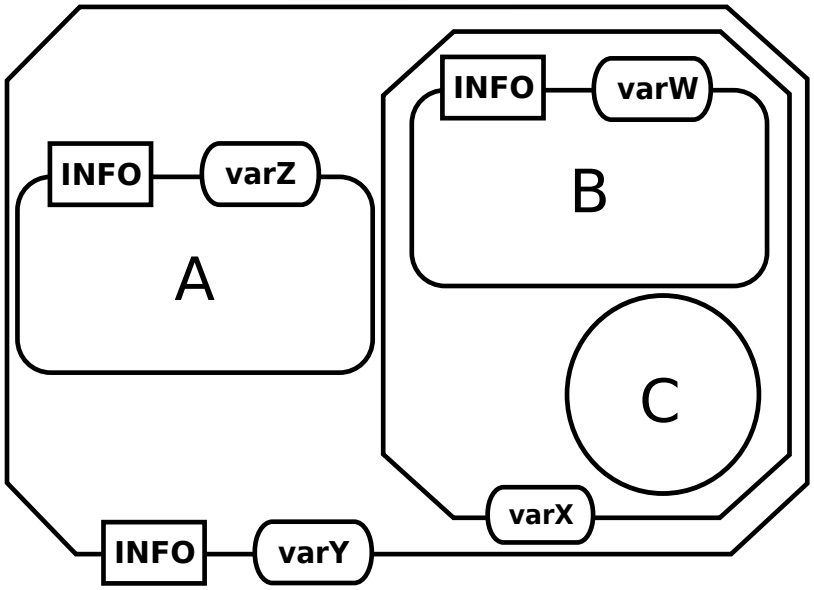
- If cardinality > 1 then the descendant glyph must be displayed as a multimer.
- If the EPN is drawn directly on a *Map* then owning_compartment is not set. We interpret this as belonging to an invisible default compartment.
- natures can only use the material type (section 3.6.1), conceptual type (section 3.6.2) or physical characteristics (section 3.6.4) controlled vocabularies.
- The appropriate subclass of CloneMarker must be used to distinguish logically identical instances of this class.
- All StateVariableDefinitions associated with the EntityType must have an associated StateVariable.

Changes from Previous Version

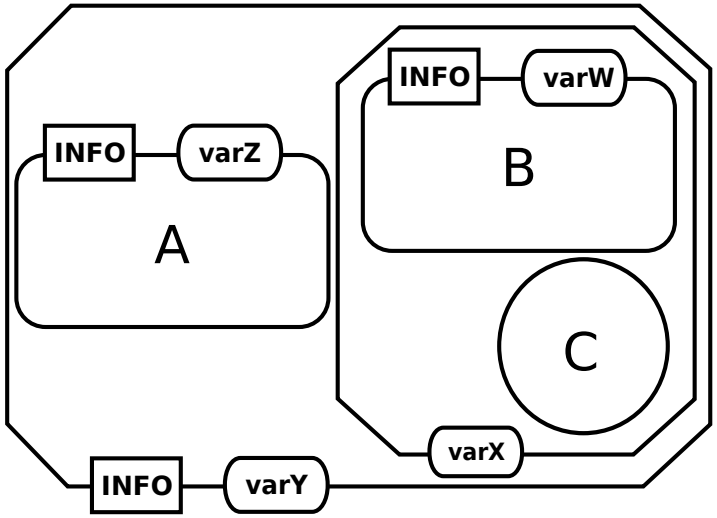
Not defined explicitly in the previous version, but the concept of the EPN and its semantics existed. The main change to previous semantics is that of the natures, which didn't formally exist before, but which now must contain a unique set of controlled vocabularies and is part of the logical key of the EntityPoolNode.

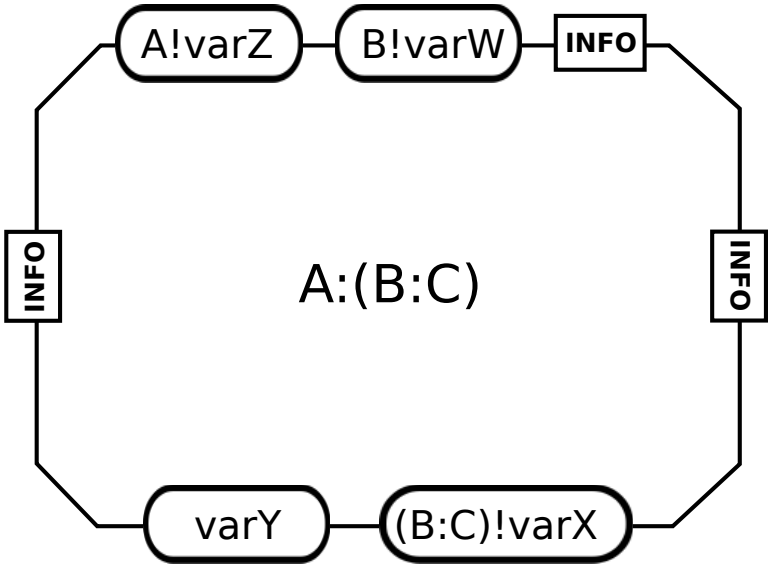
Subunits are not EPNs

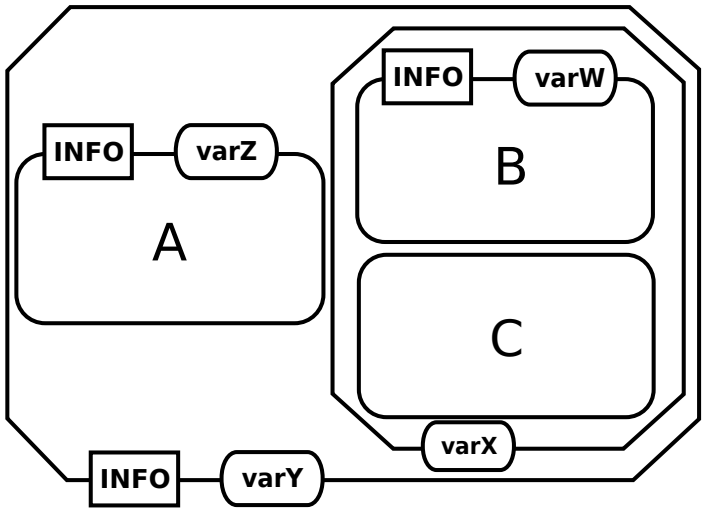
Complex Subunits are Decorators

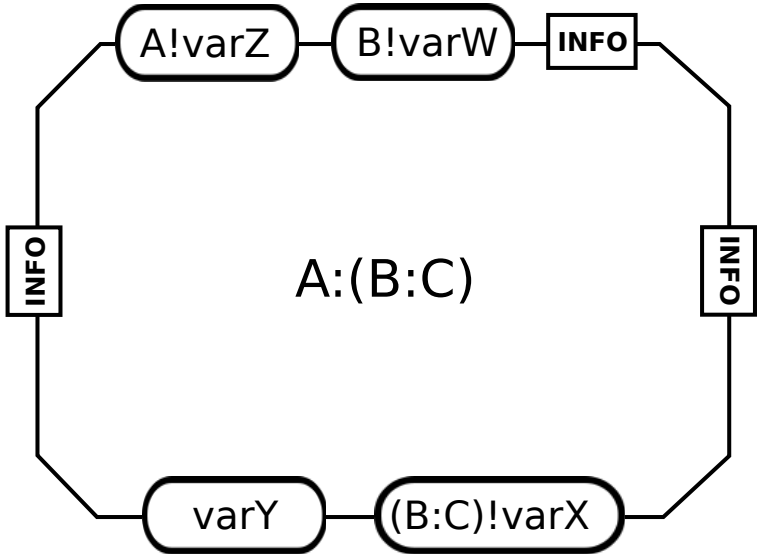


Complex naming issues

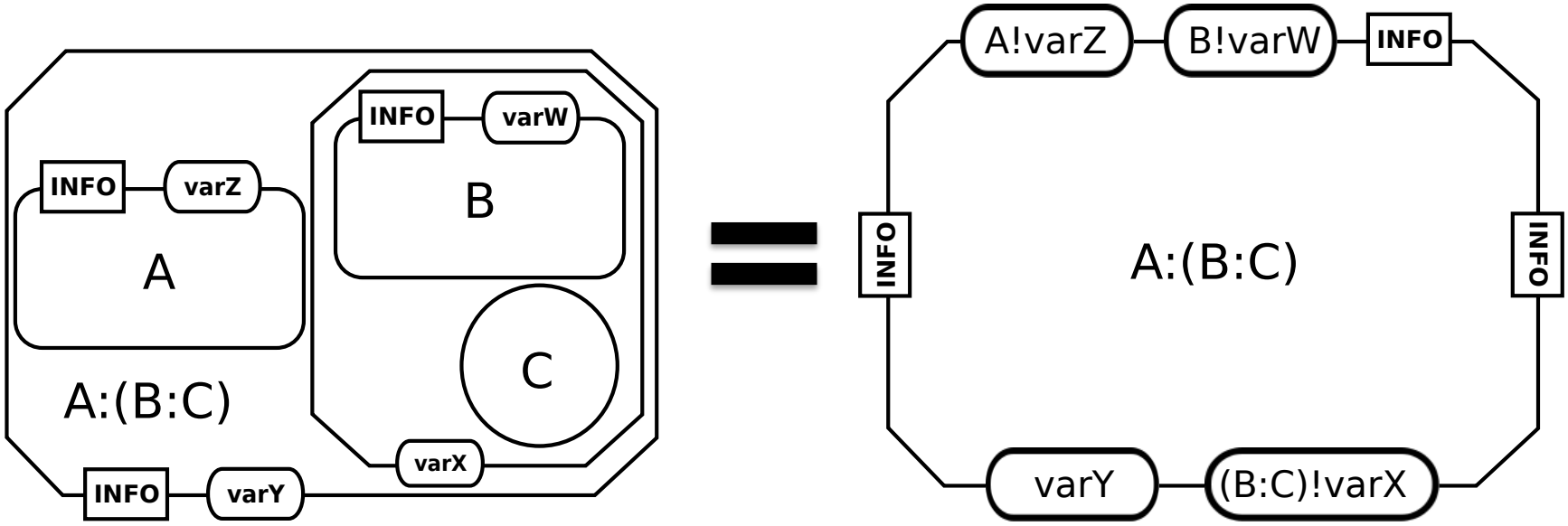






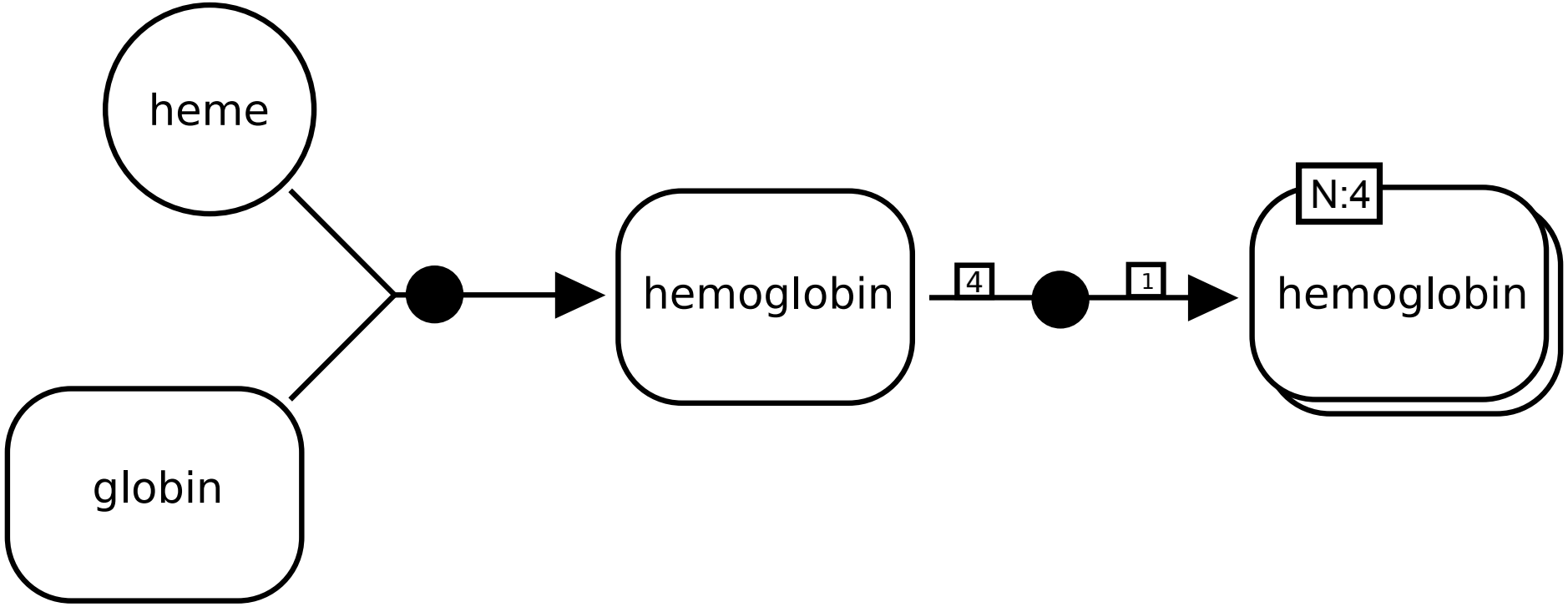


Complex Identity



Complexes must be named.

Stoichiometry



- If not specified then stoichiometry of 1
- Stoichiometry must be a positive integer

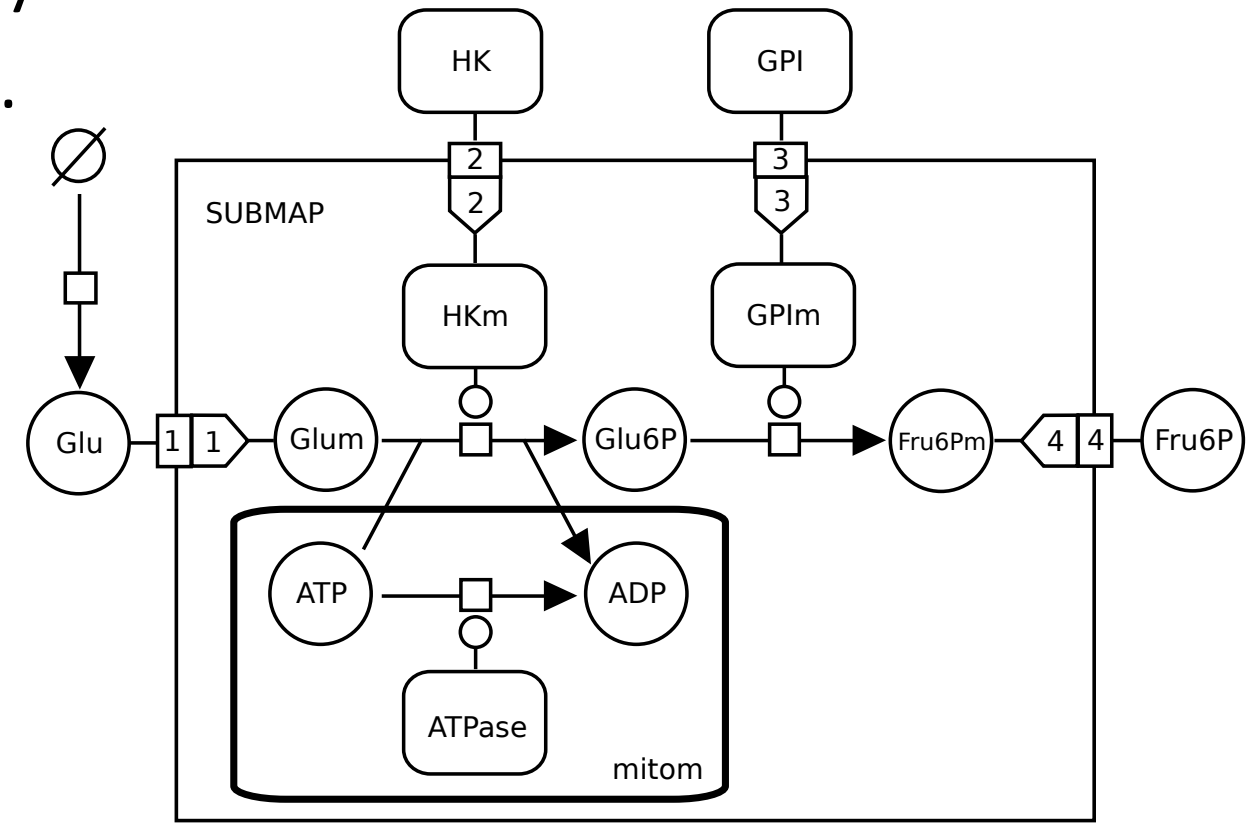
SUBMAPS

Complicated rules

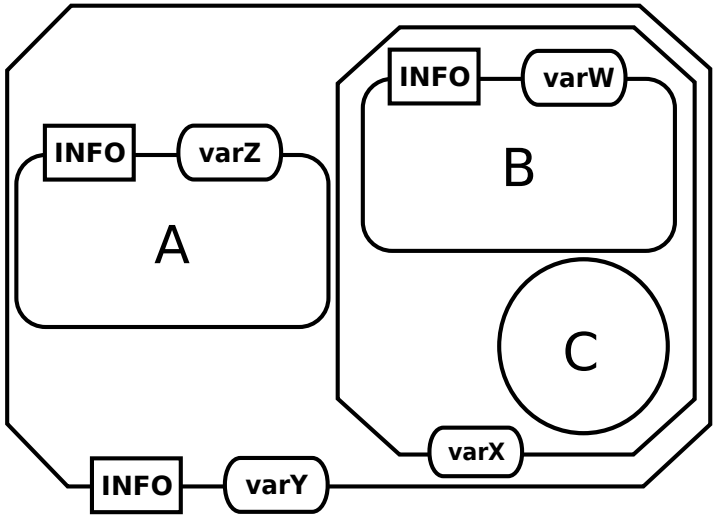
- Original idea: separate maps.
  - Complicated rules:
    - Share the same name space.
    - No current way to name maps.
  - Generally not implements
  - Problematic for SBGN-ML?

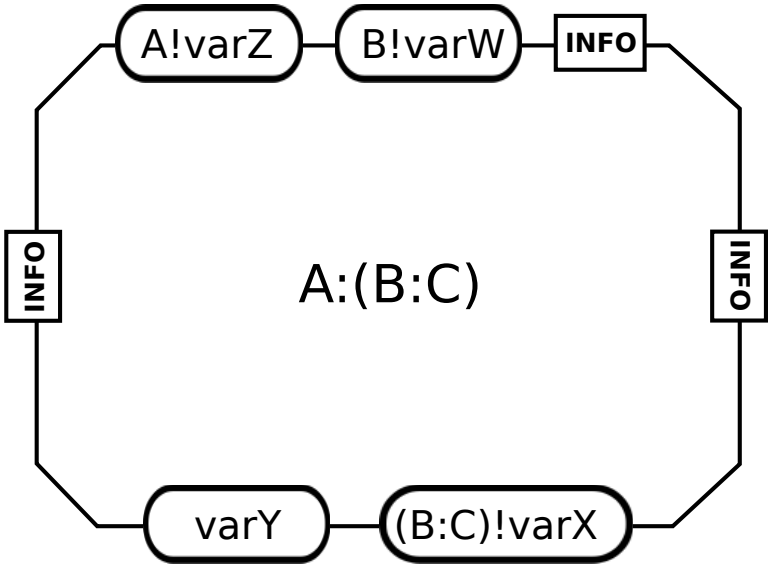
Simplify Rules

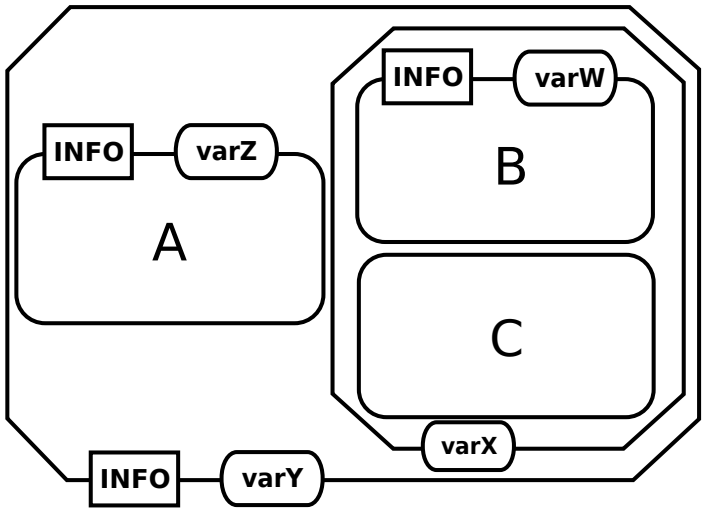
- Submaps are not on separate maps. All glyphs belong to the SAME map.
- Use it as a mechanism to collapse and expand existing maps by software.
- Cannot overlap.

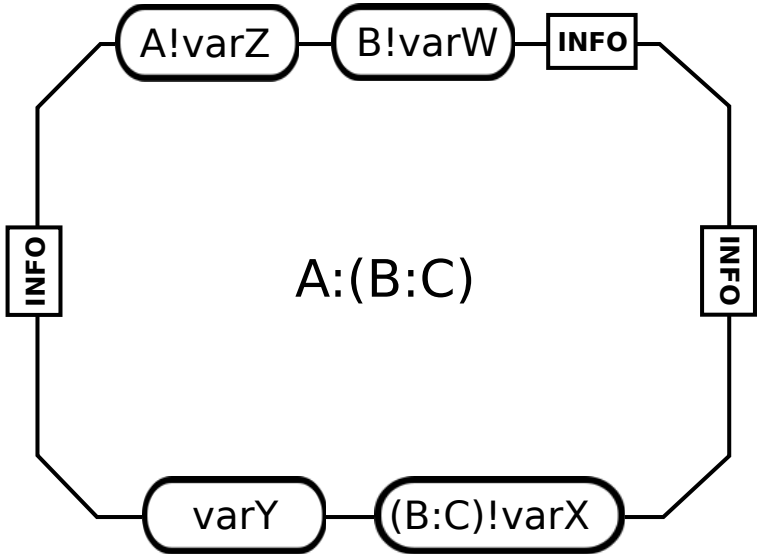


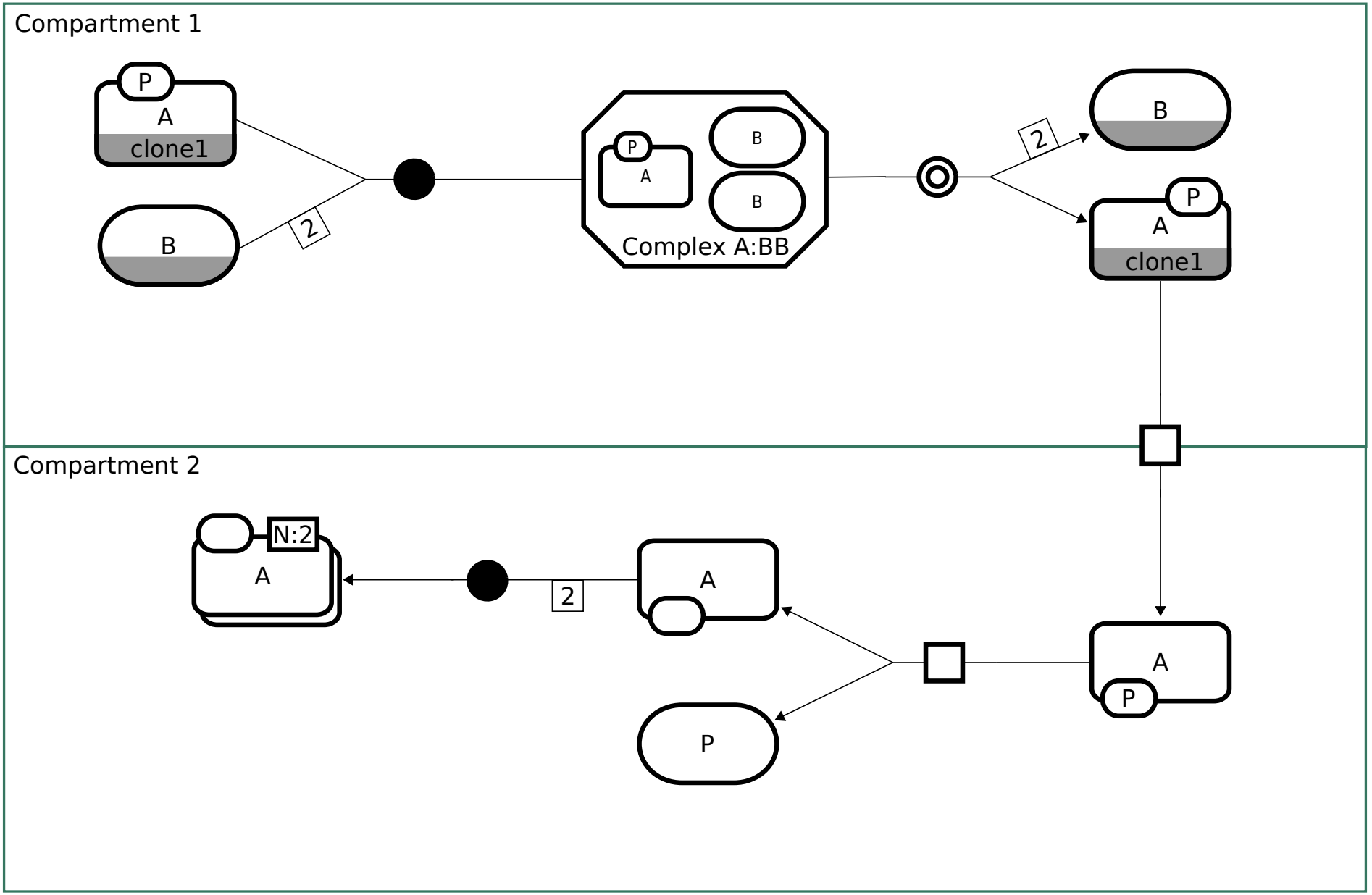
Complex Identity

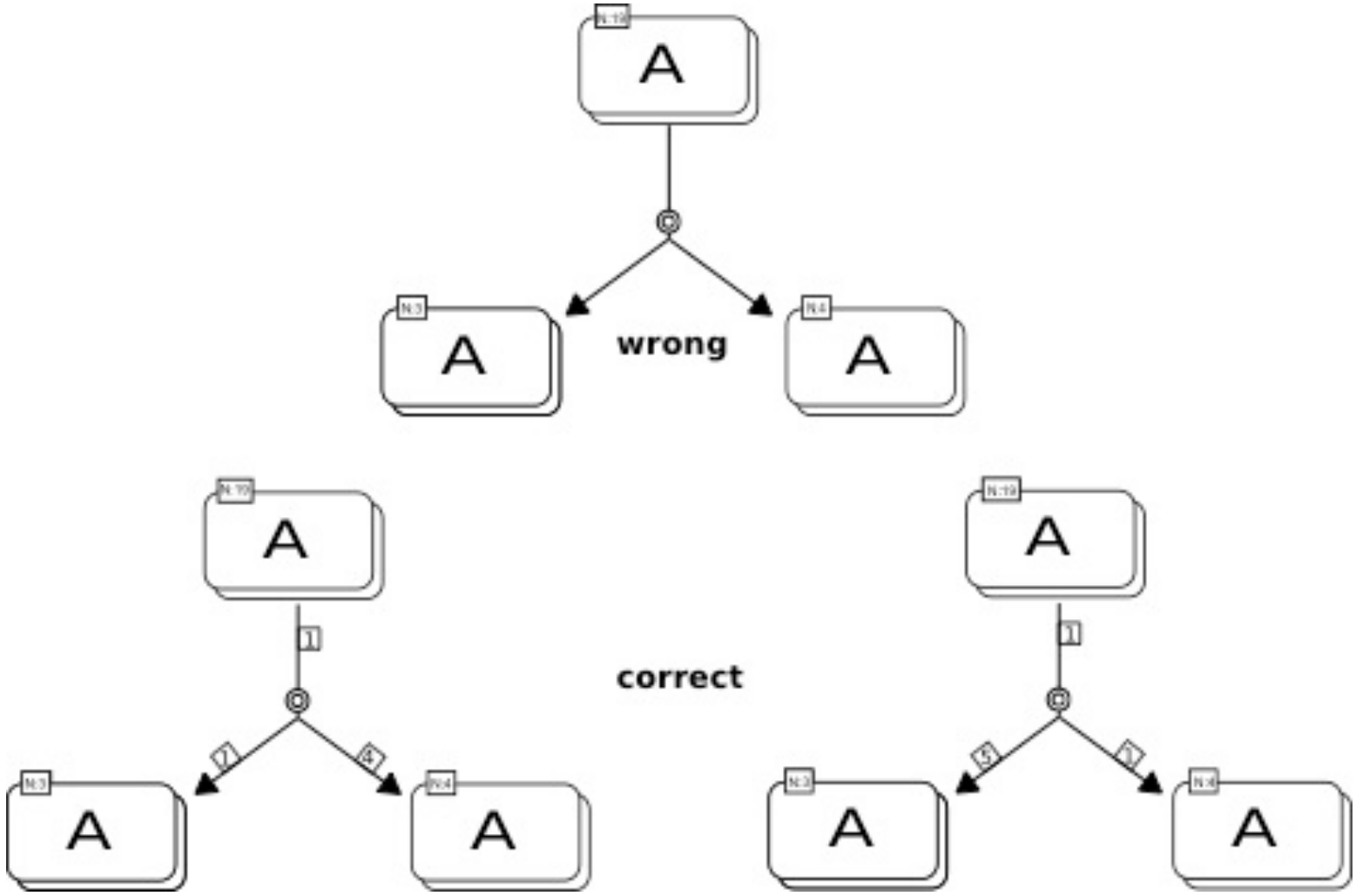




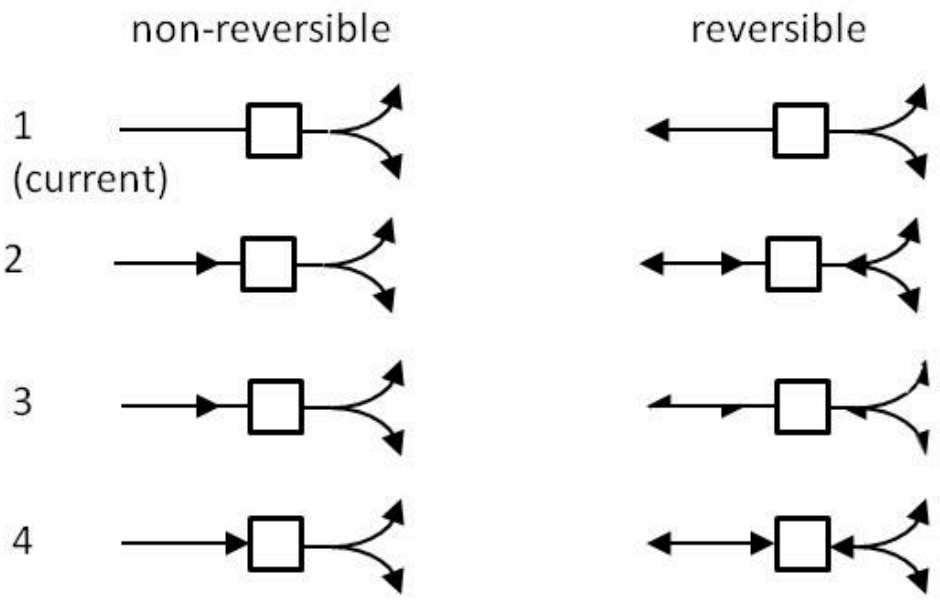





Compartment 1
P
A
clone1
B
2
P
A
B
B
Complex A:BB
2
B
P
A
clone1
Compartment 2
N:2
A
2
A
P
A
P


A
A
A
wrong
A
A
A
correct
A
A
A

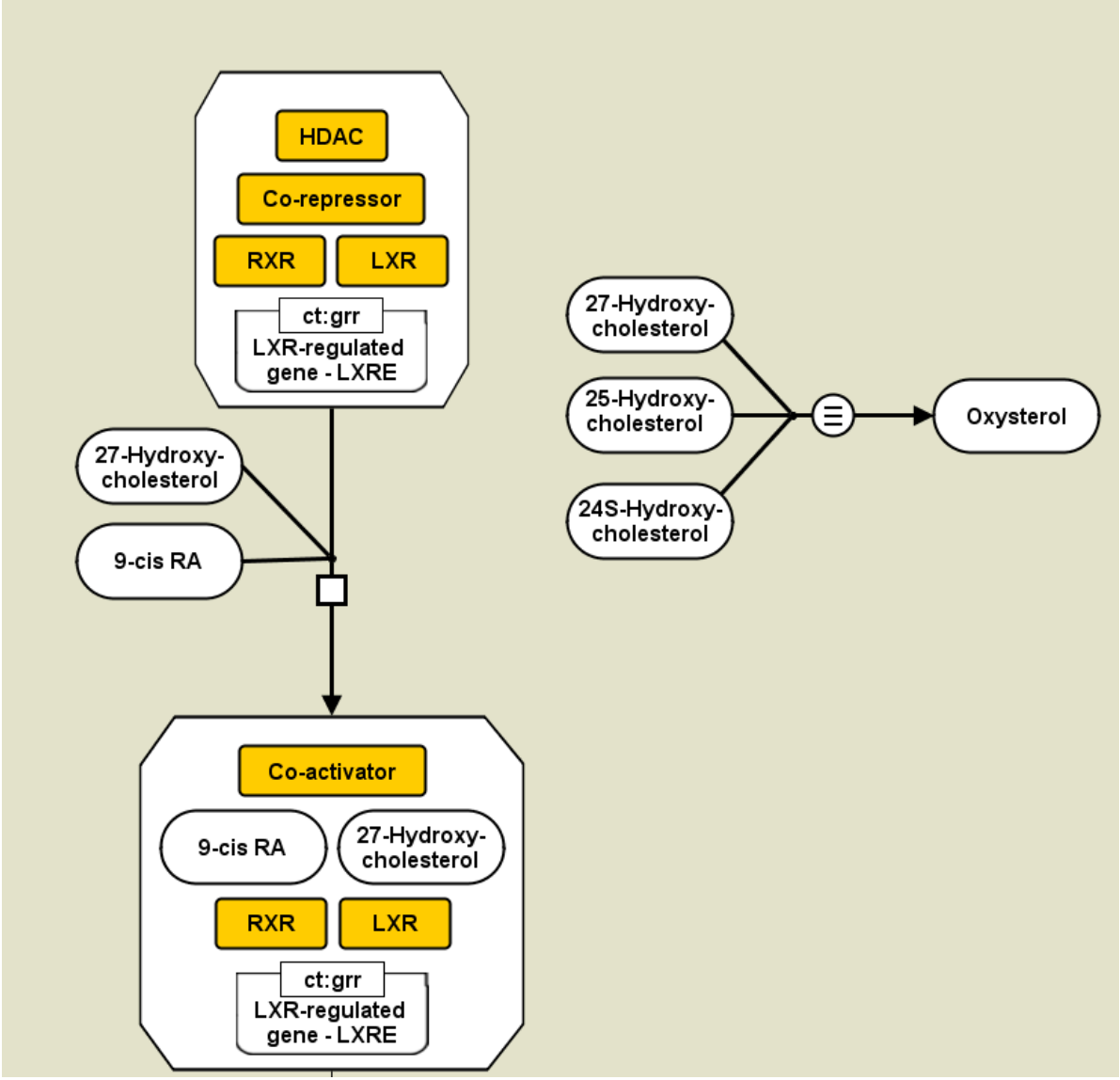
For Discussion: Reversible Arcs



Future Changes

- Generics
  - Proposal to be written
- Level 2 PD
  - Simplify rules
    - Drop clone marker
    - Single process type
    - Drop material type for identification
  - Domain?
  - Translocation within Compartment?

Generics


HDAC
Co-repressor
RXR
LXR
ct:grr
LXR-regulated gene - LXRE
27-Hydroxy-cholesterol
9-cis RA
27-Hydroxy-cholesterol
25-Hydroxy-cholesterol
24S-Hydroxy-cholesterol
Oxysterol
Co-activator
9-cis RA
27-Hydroxy-cholesterol
RXR
LXR
ct:grr
LXR-regulated gene - LXRE

Generics: outcomes

- New glyph?
- Looks like it works
- Experimental proposal?
- Tool support?

Version Numbering

- Level – Major release
- Major – Major changes, new feature, non-backwards compatible.
- Minor – Features added changed, backwards compatible
- Patch – Fixes errors no changes to semantics or features

- Do we want to change this?
  - Add major revision? L1 Version 2.0.0
  - Now is the time to do it.

Road Map for PD

- Level 1 Version 2.0
  - Draft for Review: Now
  - Revised draft for review End Aug
  - Release End Sep 2014

- Level 2
  - Not sure.

Acknowledgements

- SBGN Editors
  - Tobias Czauderna
  - Huaiyu Mi
  - Falk Schreiber
  - Anatoly Sorokin
- Former Editors
  - Emek Demir
  - Nicolas Le Novère
  - Alice Viléger
- SBGN Community