Diagramming Guidelines
For SysML and UML

Contents

1 Introduction .................................................................................................................... 2
2 Naming Conventions ..................................................................................................... 2
  2.1 Structural Diagrams ............................................................................................... 2
  2.2 Behavioural Diagrams ......................................................................................... 5
  2.3 Stereotypes ........................................................................................................... 6
3 Diagram Frame Labels ................................................................................................. 7
4 Additional Guidelines ................................................................................................. 10
  4.1 Block and Internal Block Diagrams – Showing Interfaces .................................... 10
  4.2 Block and Internal Block Diagrams – Showing Item Flows ................................... 11
  4.3 Activity Diagrams ............................................................................................... 13
  4.4 Default Tool Settings ......................................................................................... 14
    4.4.1 The Use of Colour ....................................................................................... 14
    4.4.2 Navigability ............................................................................................... 16
    4.4.3 Other Common Settings ............................................................................. 16
5 Model Structure .......................................................................................................... 16
6 Summary ..................................................................................................................... 18
7 References ................................................................................................................ 19
1 Introduction

Producing consistent SysML (and UML) diagrams that have a common look and feel is crucial to efficient and effective modelling. One of the easiest ways of helping to ensure consistency within SysML models is to set and follow effective naming and diagramming conventions that ensure a common look and feel across diagrams and help to ensure that diagrams are easy to read. Such guidelines should also save time by limiting the number of stylistic choices faced, allowing focus to be directed to the modelling rather than the drawing. This document defines a set of rules and guidelines to be followed when producing SysML diagrams.

Following this introduction, the Naming Conventions section defines rules to be applied to elements on SysML structural and behavioural diagrams together with stereotypes. This is followed by the Diagram Frame Labels section which defines rules for naming SysML diagrams. The Additional Guidelines section defines rules for specific diagram types and gives some guidelines on producing diagrams using SysML CASE tools. Finally, the Model Structure section discusses ways in which SysML models can be structured.

The guidelines presented here are those used by the authors and which form part of their SysML modelling Standard i.e. the adoption by the authors is not optional. Of course, for the reader these are only guidelines and can therefore be followed, changed or ignored. Nevertheless, it is recommended that a defined SysML modelling standard be produced and enforced as part of the reader’s systems engineering processes.

2 Naming Conventions

This section defines general naming guidelines that should be followed when producing SysML diagrams.

When modelling Standards etc. or producing models for customers, any naming conventions described in the Standard or used by the customer should be followed.

2.1 Structural Diagrams

Figure 1 illustrates the naming conventions to be followed when producing SysML structural diagrams.

The case of the text used in all elements indicates the convention to be adopted for that element. For example, an association should be named all in lower case whereas a property should be named in sentence case (i.e. initial word starts with a capital letter, all others with a lower case letter).
The naming conventions shown in Figure 1 are summarised in Table 1.

Figure 1 - Naming conventions - structural diagrams
### Diagram Item | Naming Conventions
---|---
**Block** | Name should have each word capitalised.  
Name should be singular.  
*Compartments* can be turned off to aid clarity. If necessary, add a *comment* to the diagram so that reader of the diagram knows that information has been omitted from the diagram.  
*Compartments* should be named.
**Block property** | *Property names* should be noun phrases.  
*Property names* should be in *sentence case*.  
*Property types* should have each word capitalised.  
The *direction of flow properties* must be stated.
**Operation** | *Operation names* are verb phrases; strong verbs should be used where possible.  
*Operation names* should be in lower case with spaces between words. NOTE: When modelling software, then *operation names* should be named in camel case, for example ‘checkTime’.  
*Parameter names* should be in sentence case, but with underscores replacing spaces, e.g. ‘Time_server’.  
*Parameter types* should have each word capitalised.  
*Operation return types* should have each word capitalised.
**Interface** | Name should have each word capitalised.
**Association** | Name *associations* using the active voice where possible.  
Name should be in lower case.  
Direction must be shown,  
Always show *multiplicity*, even when it is ‘1’.  
Role names should be in sentence case.
**Requirement** | Name should be in sentence case.
**Package** | *Package name* should have each word capitalised.
**Constraint** | Name should have each word capitalised.  
*Parameter names* should be in sentence case.  
*Parameter types* should have each word capitalised.  
If the constraint relationship is a Boolean test, then it should start with a lower-case letter. E.g. ‘if...’
**Note** | A note can contain *any* style of content.

| Table 1 - *Structural diagram* naming conventions |

With the naming conventions for *structural diagrams* defined, the next section discusses the naming guidelines for *behavioural diagrams*.  

---

Scarecrow Consultants Limited ©2016
2.2 Behavioural Diagrams

Figure 2 illustrates the naming conventions to be followed when producing SysML behavioural diagrams.

The case of the text used in all elements indicates the convention to be adopted for that element. For example, a *message* should be named all in lower case whereas a *use case* should be named in sentence case (i.e. initial word starts with a capital letter, all others with a lower case letter).

![Diagram of naming conventions for behavioural diagrams](image)

NOTE: activities and states are 'soft boxes' i.e. rectangles with rounded corners and not 'sausages'.

e.g. Attribute name = value

Guard condition may contain references to properties, and so may start with initials caps.

Figure 2 - Naming conventions - behavioural diagrams
The naming conventions shown in Figure 2 are summarised in Table 2.

<table>
<thead>
<tr>
<th>Diagram Item</th>
<th>Naming Conventions</th>
</tr>
</thead>
</table>
| Life line    | *Instance name* should in sentence case.  
|              | *Type name* should have each word capitalised. |
| Message      | *Message name* should be in the same case as the *operation* or *event* that they correspond to (see Table 1).  
|              | *Parameter values* should be in lower case.  
|              | *Return types* should have each word capitalised. |
| Use Case     | Name should be in sentence case.  
|              | Name should begin with a strong verb. |
| Actor        | Name should have each word capitalised. |
| State        | Names should be in lower case.  
|              | Symbol is a rectangle with rounded corners (a “soft box”) and not a sausage with semi-circular end. |
| Activity     | Names should be in lower case.  
|              | Symbol is a rectangle with rounded corners (a “soft box”) and not a sausage with semi-circular end. |
| Guard conditions | *Guard conditions* may contain references to *properties* (see Figure 1 and Table 1) and so may start with initial capitals. Otherwise the condition should be in lower case. |

Table 2 - Behavioural diagram naming conventions

With the naming conventions for *behavioural diagrams* defined, the next section discusses the naming guidelines for *stereotypes*.

### 2.3 Stereotypes

Figure 3 illustrates the naming conventions to be followed when using *stereotypes*.
Diagramming Guidelines
For SysML and UML

Stereotypes are lower case EXCEPT for those which are abbreviations or acronyms such as «HTTP» OR SysML meta-types such as Association. Where multiple stereotypes are applied they should appear above each other.

![Diagram showing stereotypes and tags](image)

Tags defined on stereotypes should be lower case

The naming conventions shown in Figure 3 are summarised in Table 3

<table>
<thead>
<tr>
<th>Diagram Item</th>
<th>Naming Conventions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stereotype</td>
<td>Stereotypes are lower case except for those which are abbreviations or acronyms such as «HTTP» or SysML meta-types such as Association. An exception to this rule is when a stereotype is defined that corresponds to an Ontology Element. For example, if defining a stereotype for the Project Ontology Element, the stereotype would be «Project».</td>
</tr>
<tr>
<td></td>
<td>Multiple stereotypes applied to same item should appear above each other.</td>
</tr>
<tr>
<td>Tag</td>
<td>Tags defined on stereotypes are lower case</td>
</tr>
</tbody>
</table>

Table 3 - Stereotype naming conventions

With the naming conventions for stereotypes defined, the next section discusses the naming of diagrams.

3 Diagram Frame Labels

This section defines guidelines to be followed when labelling diagrams.

All SysML diagrams must have a diagram frame that contains the name of the diagram. Each diagram should be named in the following fashion:

```
<frame tag> [model element type] [model element name] [diagram name]
```

Each part is separated by a space and the frame tag is bolded. The model element type and diagram name parts of the name are in brackets. The frame tag and model element name are mandatory.

The abbreviations shown in Table 4 should be used to indicate the type of diagram – known in SysML as the frame tag. If using a tool that automatically adds a diagram frame and that does not allow the frame tags to be changed, then the tag names used by the tool will be used.
Diagramming Guidelines

For SysML and UML

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Frame Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity diagram</td>
<td>act</td>
</tr>
<tr>
<td>Block definition diagram</td>
<td>bdd</td>
</tr>
<tr>
<td>Internal block diagram</td>
<td>ibd</td>
</tr>
<tr>
<td>Package diagram</td>
<td>pkg</td>
</tr>
<tr>
<td>Parametric diagram</td>
<td>par</td>
</tr>
<tr>
<td>Requirement diagram</td>
<td>req</td>
</tr>
<tr>
<td>State machine diagram</td>
<td>stm</td>
</tr>
<tr>
<td>Sequence diagram</td>
<td>sd</td>
</tr>
<tr>
<td>Use case diagram</td>
<td>uc</td>
</tr>
</tbody>
</table>

**Table 4 - Diagram frame labels**

The following shows the *model element type* associated with the different diagram kinds:

- activity diagram - activity
- block definition diagram - block, package, or constraint block
- internal block diagram - block or constraint block
- package diagram - package or model
- parametric diagram - block or constraint block
- requirement diagram - package or requirement
- sequence diagram - interaction
- state machine diagram - state machine
- use case diagram - package

The *model element type* indicates the *namespace* for the elements contained on the diagram.

The *model element name* identifies which model element type the diagram is describing.

The *diagram name* is used to give the diagram a unique name. This is particularly important when different diagrams of the same type are drawn for the same model element. The diagram name would differentiate between these diagrams since they would have the same diagram kind, model element type and model element name.

Some examples will help:

A block definition diagram is created inside a package named ‘System Structure’. The diagram shows the structural hierarchy. The diagram might be named:

```
bdd [Package] System Structure [Structural Hierarchy]
```

The first three parts of the name are determined by its type and “owner”.

Scarecrow Consultants Limited ©2016

:: 8 ::
Diagramming Guidelines
For SysML and UML

A second block definition diagram is created in the same package and shows the properties and operations of the System Elements. It could be named:

```
```

Note that the first three parts of the name are the same, since the diagram type is the same as is its owner. The [diagram name] element of the full name differentiates between them.

A parametric diagram is created for a block named “System”. The diagram shows power consumption. The diagram could be named:

```
| par [block] System [System Power Consumption] |
```

The following capitalisation rules should be applied:

- The frame tag should be lower case and emboldened
- The model element type should be lower case. (Note: the SysML specification is inconsistent on this, as are many SysML tools)
- The model element name will have the same capitalisation as the element it corresponds to in the model.
- The diagram name element should have each word capitalised.

When producing diagrams that are based on a Framework that defines a number of Views which have their own abbreviations, then:

- The View abbreviation should be in upper case. Lower case letters are allowed in order to distinguish Views that would otherwise have the same abbreviation. For example, in a model-based requirement engineering Framework a Stakeholder Scenario View might have the abbreviation SSV, whereas a System Scenario View might have the abbreviation SysSV.
- The View abbreviation should replace the standard SysML frame tag. If the SysML tool being used does not allow replacement of frame tags, then the View abbreviation should be added to the diagram name element, separated by a hyphen.
- If stereotypes have been defined to indicate the diagram usage, then the stereotype should be shown in the diagram frame.

For example, an Ontology View has been defined as part of an Architectural Framework metamodel. It has been given the abbreviation ONT and has the stereotype «ontology view» associated with it. The diagram is a usage of a SysML block definition diagram. An Ontology View is created in a package called MBSE Ontology and is intended to show a simplified Ontology. The diagram frame for this diagram would look like Figure 4.

```
| «ontology view» |
| ONT [package] MBSE Ontology [Simplified Ontology] |
```

*Figure 4 - Example diagram frame showing user-defined view abbreviation replacing frame tag*
Diagramming Guidelines  
For SysML and UML

In a tool that doesn’t allow replacement of frame tags, then the diagram frame would look like Figure 5.

«ontology view»  

bdd [package] MBSE Ontology [ONT - Simplified Ontology] 

Figure 5 - Example diagram frame showing user-defined view abbreviation added to diagram name

4 Additional Guidelines

This section contains additional guidelines that apply to particular diagram types.

4.1 Block and Internal Block Diagrams – Showing Interfaces

This section defines guidelines to be followed when producing block definition diagrams and internal block diagrams that show interfaces.

The guidelines shown in Figure 6 are summarised in Table 5.
Diagramming Guidelines  
For SysML and UML

<table>
<thead>
<tr>
<th>Diagram Item</th>
<th>Guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provided interface</td>
<td>Place on the left of the block or part. If this is not possible, try to place them on the top of the block or part. Where possible, place the interface label above the ball symbol. When a required and provided interface are connected on a diagram (as for Interface X in Figure 6), use a single label.</td>
</tr>
<tr>
<td>Required interface</td>
<td>Place on the right of the block or part. If this is not possible, try to place them on the bottom of the block or part. Where possible, place the interface label above the socket symbol. When a required and provided interface are connected on a diagram (as for Interface X in Figure 6), use a single label.</td>
</tr>
<tr>
<td>Port</td>
<td>Port name should be in sentence case. Port type should have each word capitalised (corresponding to the model element that types the port). Port names and/or types can be omitted for clarity.</td>
</tr>
</tbody>
</table>

Table 5 - Interface naming conventions

With the guidelines for interfaces defined, the next section discusses the guidelines for the related subject of item flows.

4.2 Block and Internal Block Diagrams – Showing Item Flows

This section defines guidelines to be followed when producing block definition diagrams and internal block diagrams that show item flows.
Diagramming Guidelines
For SysML and UML

**Figure 7 - Block and internal block diagrams – showing item flows**

The guidelines shown in Figure 7 are summarised in Table 6.

<table>
<thead>
<tr>
<th>Diagram Item</th>
<th>Guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port with incoming flows</td>
<td>Place on the left of the block or part. If this is not possible, try to place them on the top of the block.</td>
</tr>
<tr>
<td>Port with outgoing flows</td>
<td>Place on the right of the block or part. If this is not possible, try to place them on the bottom of the block.</td>
</tr>
<tr>
<td>Port with conjugated flows</td>
<td>Place on the left of the block or part. If this is not possible, try to place them on the top of the block.</td>
</tr>
<tr>
<td>Port with two-way, non-conjugated flows</td>
<td>Place on the right of the block or part. If this is not possible, try to place them on the bottom of the block.</td>
</tr>
<tr>
<td>Port</td>
<td>Port name should be in sentence case.</td>
</tr>
<tr>
<td></td>
<td>Port type should have each word capitalised (corresponding to the model element that types the port).</td>
</tr>
<tr>
<td></td>
<td>Port types can be omitted only for single-direction ports.</td>
</tr>
<tr>
<td>Item flow</td>
<td>Item flow name should be in sentence case.</td>
</tr>
<tr>
<td></td>
<td>Item flow type should be in upper case.</td>
</tr>
<tr>
<td></td>
<td>Item flow name and type should be placed above the connector along which it flows.</td>
</tr>
</tbody>
</table>

Table 6 - Item flow naming conventions

With the guidelines for *item flows* defined, the next section discusses the guidelines for *activity diagrams*. 
4.3 **Activity Diagrams**

This section defines guidelines to be followed when producing *activity diagrams*.

<table>
<thead>
<tr>
<th>Diagram Item</th>
<th>Guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>Check that all <strong>activities</strong> (or <strong>states</strong> on a <strong>state machine diagram</strong>) have an input. Missing inputs suggest a missing edge or <strong>initial node</strong>. Check that all <strong>activities</strong> (or <strong>states</strong> on a <strong>state machine diagram</strong>) have an output. Missing outputs suggest a missing edge or <strong>final node</strong>. <strong>Activities</strong> should have only one input. Use a <strong>merge</strong> if necessary.</td>
</tr>
<tr>
<td>Fork</td>
<td><strong>Forks</strong> should have only one input. Use a <strong>merge</strong> if necessary.</td>
</tr>
</tbody>
</table>

The guidelines shown in Figure 8 are summarised in Table 7.

With the guidelines for activity diagrams defined, the next section discusses issues around tool settings.
4.4 **Default Tool Settings**

All SysML tools have default settings that control the appearance of diagrams in many ways, such as colour, navigability etc. This section discusses some of these issues.

4.4.1 **The Use of Colour**

The use of colour can be used to add extra information to a diagram or to make diagrams clearer. However, colour should not be used without careful consideration. As a general rule, all diagram elements should be drawn with black text on a white background. Where colour is used, then the diagram must include a key that explains the colour scheme used.

In addition, some tools allow diagrams to be produced that have 3D effects, gradient fills, shadows and curved lines that can be applied to diagram elements. These should not be used, and should be turned off in the tools options. As an example, the following three diagrams are taken from a SysML tool.

Figure 9 - Example **block definition diagram** showing inappropriate use of shading and other graphical effects

Figure 9 shows the diagram as produced by the tool with the default graphical options turned on (but with the colour used for the blocks replaced with grey). Clearly this diagram is not fit for purpose. The use of curved connectors makes the diagram almost unreadable and the 3D effects and shadowing do nothing to add to either the clarity or meaning of the diagram.

The same diagram, but without the use of curved lines, 3D effects and shadows is shown in Figure 10.
Diagramming Guidelines
For SysML and UML

Figure 10 - Example block definition diagram showing use of colour

The diagram, like the original in Figure 9, makes use of colour, with the blocks being filled with a pink background (here rendered as a grey background to the blocks). If colour is required, then thought should be given to its choice and purpose. For example:

- How well will the colour reproduce when printed in black and white or greyscale?
- Are any fonts or other symbols that lie on top of the colour readable?
- Will the chosen colours be problematic to those who are colour-blind?
- What meaning is attributed to the colours used?
- Have such meanings been made clear in a key?

Figure 11 - Example block definition diagram rendered in black and white

Using colour is problematical at best. If it must be used, use it sparingly. The SysML specification does NOT define colours for SysML elements. The best approach is to use simple black and white, as shown in Figure 11.
4.4.2 Navigability

In each of the figures in Section 4.4.1 the compositions from ‘Block1’ all have an open arrowhead at the part end of the relationship. These arrowheads are used to show the navigability of the relationship. However, this is a concept that is mainly needed when modelling software (the notation is part of SysML through its inheritance from the UML) and is used less often in more general systems modelling.

The arrowheads are shown on these diagrams because, in the tool used to produce these diagrams, the display of navigability is turned on by default. This requires the user to change this default setting, if the tool allows this, or if not permitted, requires the user to edit the setting for each association when added to the diagram.

4.4.3 Other Common Settings

Other common diagramming settings that need to be considered include the display of role names on associations, whether whole-part relationships should default to composition or aggregation, whether association names should be displayed by default, whether compositions, aggregations and generalisation relationships should be displayed in a tree layout, what colours should be used for diagram elements such as blocks, requirements, use cases etc.

A SysML tool should allow such settings to be changed once for a model and not force the modeller to change the settings for every diagram. Even more desirable is the ability to define these settings for all models created with the tool. This allows standard settings to be rolled-out across an entire organisation. Unfortunately, not all tools allow changes to default settings to be made.

5 Model Structure

When creating a SysML model it is important, in order to aid navigability and ease of use, that the model is well structured. However, it is impossible here to define a structure that is suitable for all projects; any structure adopted must be set up so as to meet the needs of the project for which the model is being created. The authors have created models that have been structured in many different ways. Some examples of structuring adopted by the authors on projects include by:

- Life Cycle Stage
- Engineering process or activity
- System and sub-system
  - Structure
  - Behaviour
- Team
- Architecture framework
- Modelling framework

Sometimes model structure is a combination of these. For example, a model might first be structured by Life Cycle Stage, then within each Stage further structured by System. Figure 12 shows part of a model of a Standard (ISO15288:2008 – see [ISO15288:2008]) that is structured according to the ‘seven views’ Process modelling Framework described in Chapter 7 of [Holt & Perry 2013]. Note the use of additional packages to contain aspects of the model such as styles (symbol colours etc.), stereotypes and scripts (the tool in which this model was produced allows the user to enhance functionality through user-defined scripts).
Another example is given in Figure 13. Here, the model is structured largely into a structural and behavioural split influenced by engineering activity. For example, the ‘Coffin Escape Schematic’, ‘Requirements’, ‘Stakeholders’ and ‘Scenarios’ packages contain the parts of the model concerned with Requirements, whereas the ‘System’, ‘Constraints’, ‘Processes’ and ‘Units and Types’ packages contain the parts of the model concerned with design, defining System structure and behaviour.

Some SysML tools have a very useful facility that allows the model to be navigated both by the package structure (as in Figure 12 and Figure 13) and by model Perspective and View. The two diagrams in Figure 14 and Figure 15 illustrate this. These examples are taken from the Martian invasion case study model, discussed in Chapter 13 of [Holt & Perry 2013].

The model structure shown in Figure 14 is structured in a similar way to the model shown in Figure 13, showing a structure based on a structural and behavioural split influenced by engineering activity.
Diagramming Guidelines
For SysML and UML

The model has been constructed using an Architectural Framework that defines a number of Perspectives and Viewpoints (See Chapters 11 & 13 of [Holt & Perry 2013]). The model consists of a number of Views that conform to the Viewpoints (which are simply the definition of Views). Irrespective of the model structure defined by the user, the tool allows the model to be navigated by the Perspectives and Views defined by the Framework. An example is shown in Figure 15, which shows the packages containing the Views that make up the System Perspective. The root package shows the Perspective and the sub-Packages correspond to each View in that Perspective (not all Views are shown). The View packages show all the model diagrams that conform to that View, no matter where in the package structure shown in Figure 14 they reside. The Perspective and View structure is defined by and enforced by the tool; the user is not allowed to change this structure in any way.

Figure 15 - Model structure - viewed by Perspective and View

Note that the package names for the View packages are lower case, which contradicts the naming convention described in Section 2.1. This is because the tool used requires the use of stereotypes to name Views in a way that makes them browsable as shown in Figure 15; so the package labelled system structure view is displaying the all diagrams stereotyped «system structure view» and similarly for the others. The naming convention can thus be seen to be consistent with the guidelines for naming stereotypes given in Section 2.3.

Finally, many SysML tools will suggest a predefined model structure when a new model is created in the tool. While such structures may be of use in suggesting a starting point for the way the model is organised, they are rarely of much use beyond that. The model will be much easier to navigate if time is taken to define the structure that makes sense to the users of the model. The structure is up to you, but should be covered in your engineering Processes or modelling style guides.

6 Summary
This document has presented a number of guidelines and conventions that should be followed to ensure that the diagrams in a SysML model have a common look and feel. These guidelines are a starting point only; feel free to use them, but remember that any such guidelines should be tailored to the needs of your Organisation. Producing a SysML modelling Standard that is incorporated into the reader’s systems engineering Processes is to be encouraged as a way of enforcing the guidelines.

Wherever possible any SysML tool used should allow the modeller to organise a model as desired, perhaps providing multiple ways of viewing the model’s contents. Avoid using the tool’s default model structure, if it provides one; such a structure will rarely be suitable.

In addition, issues regarding the use of colour, shading, curved lines, default relationship settings etc. should be considered and guidelines produced. If possible the tool should be configured so that these defined settings are set as defaults.
7 References


