Appendix D

Using SysML concepts in UML

I did never know so full a voice issue from so empty a heart: but the saying is true ‘The empty vessel makes the greatest sound’.

William Shakespeare (1564–1616), Henry V – ACT IV, SCENE IV

D.1 Introduction

SysML includes some useful notation and concepts not found in UML, in particular required and provided features, flow properties and parametric constraint blocks. The diagrams omitted from SysML (timing and deployment) may prove problematic for systems engineers wishing to fully model a System, particularly those working with software engineers who may be modelling with the UML and therefore using the omitted diagrams.

One solution for systems engineers is to use UML rather than SysML, but to add SysML constructs to UML. This appendix illustrates how this may be done.

D.2 Features

SysML allows for properties and operations to be marked as required, provided or required and provided. Adding this to UML is achieved by defining three stereotypes (Figure D.1).

An example of these in use is shown in Figure D.2.

D.3 Full and proxy ports

SysML allows for ports to be marked as full or proxy. Adding this to UML is achieved by defining two stereotypes (Figure D.3).

An example of these in use is shown in Figure D.4.

D.4 Flow properties and ports that use them

In order to model the SysML concept of flow properties a new stereotype is needed, along with an associated tag that uses an enumerated type. This is seen in Figure D.5.
Appendix omitted from Third Edition of “SysML for Systems Engineering” by Jon Holt and Simon Perry

720  SysML for systems engineering

![Diagram](image1)

**Figure D.1** Stereotypes to add required, provided, etc., features to UML

![Diagram](image2)

**Figure D.2** UML class showing use of provided and required features

![Diagram](image3)

**Figure D.3** Stereotypes to add full and proxy ports to UML

![Diagram](image4)

**Figure D.4** UML class showing use of full and proxy ports
A comparison of a class using this stereotype and the corresponding SysML block is shown in Figure D.6.

Although SysML no longer explicitly includes the idea of a flow port (this concept was dropped in SysML version 1.3, the version that this book is based on) it does allow for ports that are typed by blocks containing flow properties to be represented differently (see, for example, Figure B.5). In UML such ports could be indicated using a «flow» stereotype along with a tag to indicate the direction of the flow. These are defined as shown in Figure D.7.
Note that the diagram in Figure D.7 makes use of the ‘DirectionType’ defined in Figure D.5. Also, the notion of a port being conjugated is found in UML as well as SysML, so it does not have to be redefined.

A comparison of UML diagram using this stereotype and the corresponding SysML diagram is shown in Figure D.8.

![Diagram](image)

**Figure D.8** A comparison of ports with flow properties in SysML (a) and UML (b)

UML does allow alternative images to be associated with stereotypes, with stereotyped elements displayed using the alternative image rather than the textual stereotype notation. The various flow port stereotype definitions could be defined with alternative images in order to make the UML diagram much more similar to the SysML, but care is always needed when defining new graphical symbols as their use may make reading a diagram difficult for those not familiar with the new symbols. A standard UML element with a stereotype makes it clear to the reader that the basic UML notation has been extended.

Where SysML has item flows, these are based directly on UML’s information flows, which can, therefore, be used in UML to represent item flows. This has been done in Figure D.8.

### D.5 Parametric constraints

SysML has the concepts of parametric constraint blocks. These are defined on a SysML block definition diagram and used on a parametric diagram. Examples are shown in Figures D.9 and D.10.

Again, UML can be extended through stereotypes to include these concepts. Figure D.11 shows an example of the definition of the needed stereotypes. Again, this diagram needs to be accompanied by text and constraints describing the use of the stereotypes.

Figure D.12 shows a UML class diagram defining two parametric constraints using the stereotypes defined in Figure D.11. Note the use of an additional ‘constraints’ compartment on these classes. UML allows for any number of additional named compartments to be added to a class.
Appendix omitted from Third Edition of “SysML for Systems Engineering” by Jon Holt and Simon Perry

Using SysML concepts in UML    723

Figure D.9  SysML parametric constraint block definition

Figure D.10  SysML parametric constraint property usage (partial diagram)

Figure D.11  UML stereotype definitions for parametrics
D.6 Activity diagrams

SysML introduced some small notational additions to activity diagrams that allow rates and probabilities to be added to activity edges, and the concepts of buffering or no overwrite to be added to object nodes.

All of these concepts can be added to UML using either the existing constraints notation (for probabilities and defined rates) or stereotypes (for discrete and continuous rates and for the object node notations).

The stereotypes needed are shown in Figure D.14. UML diagrams produced using these stereotypes will look very similar to their SysML counterparts.

D.7 Requirement diagrams

SysML requirement diagrams make use of a number of stereotyped dependencies, together with stereotyped blocks and the nesting relationship. The required stereotypes are shown in Figure D.15.
By stereotyping dependencies and classes, and using composition to replace nesting, requirement diagrams can easily be drawn in UML. Note that the nesting relationship is part of the UML, but is usually used to show structures of nested packages. The meaning of the nesting relationship in its use on SysML requirement diagrams is one of composition to show that a requirement is composed of a number of sub-requirements. For this reason the composition relationship, rather than the nesting relationship, is recommended. In fact, earlier versions of the SysML specification used the composition relationship rather than the nesting relationship.

An example of a UML class diagram used as a requirements diagram is shown in Figure D.16.
The definition of the «testCase» stereotype is left as an exercise for the reader.

**Figure D.16 UML class diagram used to model requirements**

The System shall enable the Escapologist to perform the 'concrete coffin' Coffin Escape stunt.

The System shall ensure that the excitement of the Audience is maximised.

The System shall ensure that an Audience satisfaction survey is carried out after every performance.

The System shall allow the Coffin Escape stunt to be performed using different Fluid, not just Concrete. Examples include Custard, Water, etc.