



Enterprise Architect

User Guide Series

# SysML Models

Author: Sparx Systems

Date: 26/07/2018

Version: 1.0

# Table of Contents

Systems Engineering	3
Parametric Diagram Modeling Assistant	9
Create ConstraintBlock from Equations	10
Bind Parameters of a ConstraintProperty	15
Systems Modeling Language (SysML)	16
SysML Activity Toolbox	19
SysML Block Definition Toolbox	24
Generate Properties From Block Associations	28
Block Element Compartments	29
Show Direction on SysML Ports	31
Nested Ports in SysML	32
SysML Interaction Toolbox	33
SysML Internal Block Toolbox	35
SysML Model Elements Toolbox	38
SysML Parametrics Toolbox	42
SysML Requirements Toolbox	45
SysML StateMachine Toolbox	48
SysML Use Case Toolbox	51
A SysML Requirements Model	53
A SysML Operational Domain Model	54
Compose System Design	56
Create Reusable Subsystems	58
Migrate SysML Model to Later SysML Version	60
Simple Parametric Simulation (Legacy)	62

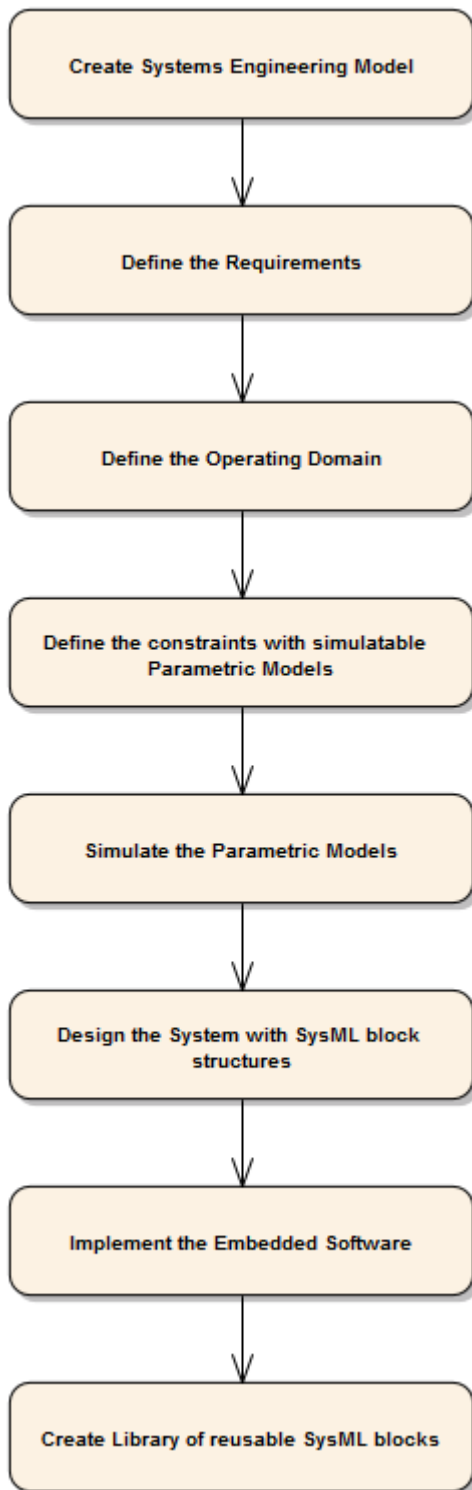
# Systems Engineering

Systems engineering is a field of engineering that deals with how to research, design and manage complex systems over their life time. It focuses on the whole system and typically involves a number of sub-disciplines including requirements, reliability, logistics, design, testing and maintenance; it requires sophisticated project management techniques.

Systems engineering not only deals with the problem of the system itself but also involves processes, optimization and risk management. Enterprise Architect provides a platform for the system engineer using the Systems Modeling Language (SysML) and model based development, and in addition to the engineering models the tool provides support for project and process management.

## Model Systems using SysML in Enterprise Architect

Work through the steps represented by this flowchart:

**Step**





Create a Systems Engineering model to develop your system

Create a Requirements model to define the systems requirements and expectations

Create an Operational Domain model, which describes the environment that the system operates within, and the entities it interacts with

Create Constraint models to describe the systems operating characteristics using parametric models
Simulate the parametric models to verify their correctness and obtain the desired characteristic
Design the system's composition using SysML Blocks and Parts
Implement the embedded software using UML Classes and behavioral models
Create a Library of reusable SysML blocks, representing subsystems that can be reused on other projects, and other common Type definitions

## Areas to consider

Subject	Details
<p>Systems Engineering Modeling</p> 	<p>A step by step guide to creating your first Systems Engineering Model. See the <i>Create a Systems Engineering model from a template</i> section.</p>
<p>SysML</p> 	<p>Get started with SysML with this introduction to the MDG Technology, available toolboxes and more.</p>
<p>SysML Requirements</p> 	<p>The SysML Requirements Model provides the system requirements, the expected abstract behavior, and the operating constraints that the designed system must conform to.</p>
<p>Parametrics</p> 	<p>SysML Parametric models support the engineering analysis of critical system parameters, including the evaluation of key metrics such as performance, reliability and other physical characteristics.</p>

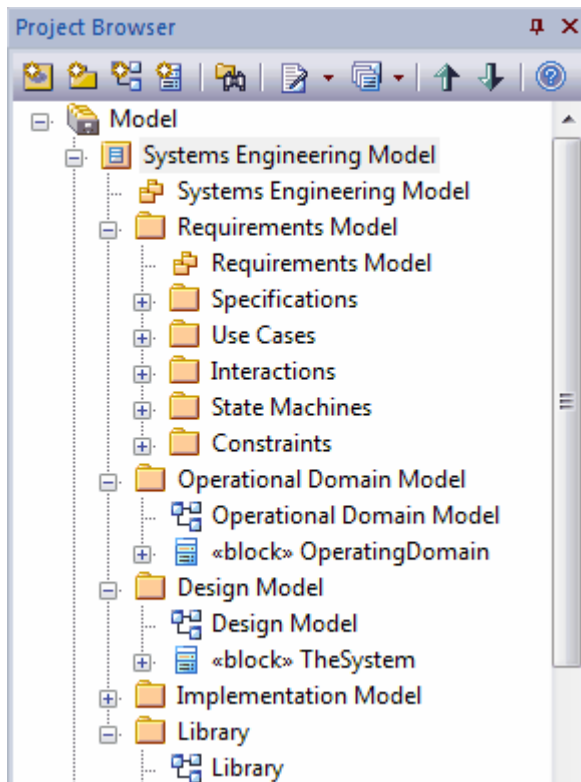
## Create a Systems Engineering Model from a template

Step	Action
1	<p>In the Project Browser, either:</p> <ul style="list-style-type: none"> <li>Click on the 'New Model From Pattern' icon in the Project Browser toolbar, or</li> <li>Right-click on a model root node and select the 'Add a Model using Wizard' option, or</li> <li>Right-click on a Package and select the 'Add a Model using Wizard' option</li> </ul> <p>The Model Wizard view displays.</p>

2	In the 'Select From' field, click on the drop-down arrow and select 'Systems Engineering Model'. Alternatively, if it is listed in the 'Technology' panel, select the 'Systems Engineering Model' item.
3	In the 'Name' panel, select the checkbox next to the 'Systems Engineering Model' icon.
4	Click on the OK button.

## Example

This model structure is created in the Project Browser:



This Systems Engineering Model diagram encapsulates the key components of the Systems Engineering model.

Systems Engineering Model captures the requirements, operating conditions and performance constraints of a complex system. It captures the design of the system in terms of distinct, configurable, reusable subsystem components.

#### Requirements Model

- + Navigation Links
- + Specifications
- + Use Cases
- + Interactions
- + State Machines
- + Constraint Blocks

Requirements Model contains the defined system requirements, its desired behavior, and operating constraints the system must conform to.

#### Domain Model

- + ListeningDomain

Domain Model contains the SysML blocks that describe the operating conditions under which the design system will operate.

#### Design

- + Portable Audio Player

Design contains the SysML blocks that describe the system's structure. It describes how reusable subsystems fit together to fulfil the design needs.

#### Implementation Model

- + Hardware
- + Software

Implementation Model describes the computing hardware and software artifacts necessary to implement the actual embedded, software-intensive system.

#### Module Library

- + Audio Player Value Types
- + Constraint Blocks
- + Specifications and Types
- + External
- + Components

Module Library contains a collection of reusable SysML blocks to develop the design and constraints of this system and others in this project.

## Notes

- Systems Modeling Language (SysML) is provided with the Unified edition and Ultimate edition of Enterprise Architect





## Parametric Diagram Modeling Assistant

This feature is available from Enterprise Architect release 14.1.

From release 14.1, Enterprise Architect provides a set of convenient tools to help you create ConstraintBlocks and parameters, by parsing mathematical equations to create parameter binding connectors, and using a hierarchical element picker to create embedded elements.

## Create ConstraintBlock from Equations

This feature is available from Enterprise Architect release 14.1.

You can create a SysML Constraint Block with parameters from one or more constraints (mathematical equations) by using the new 'Edit Constraint Block' dialog.

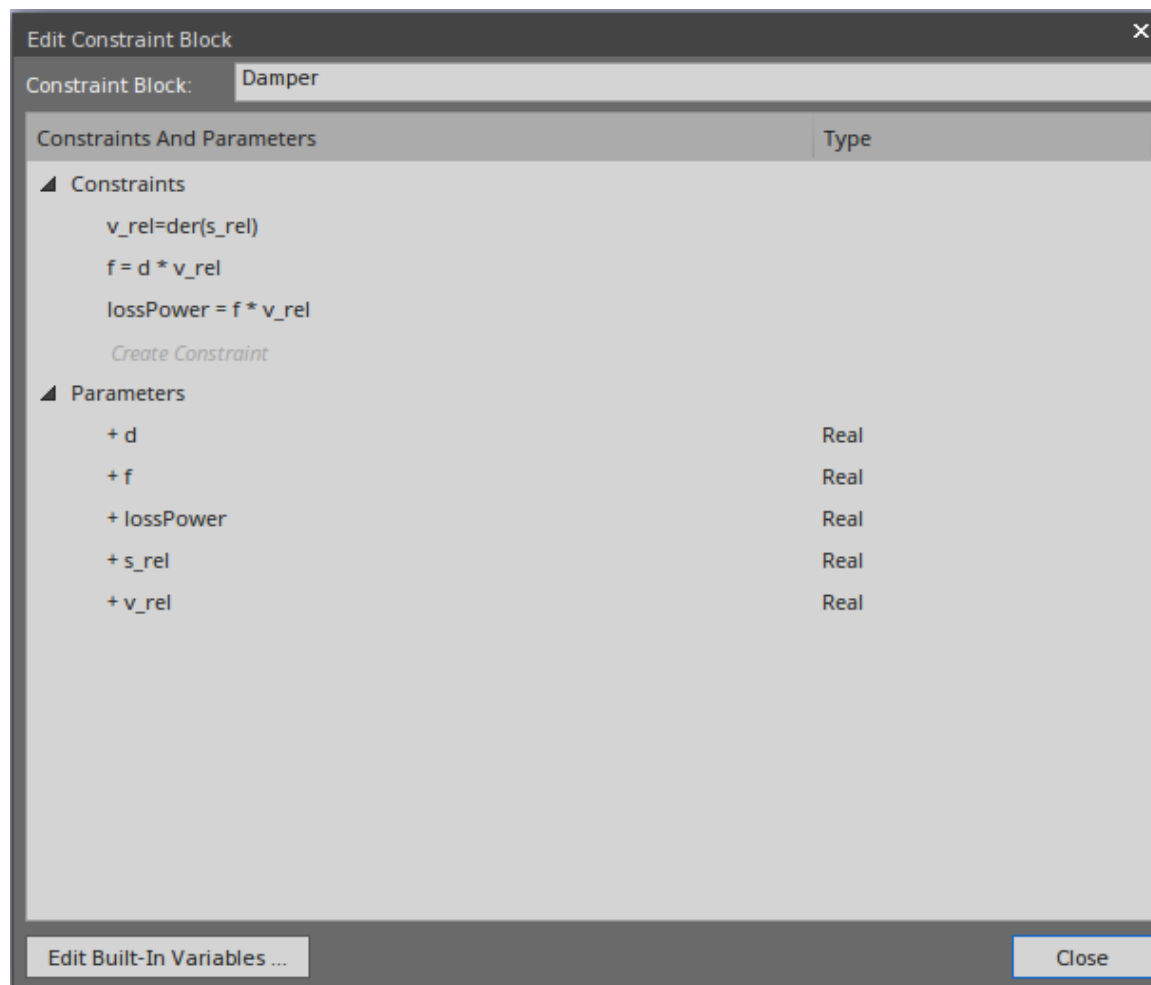
The constraints will be parsed and parameters will be extracted.

### Access

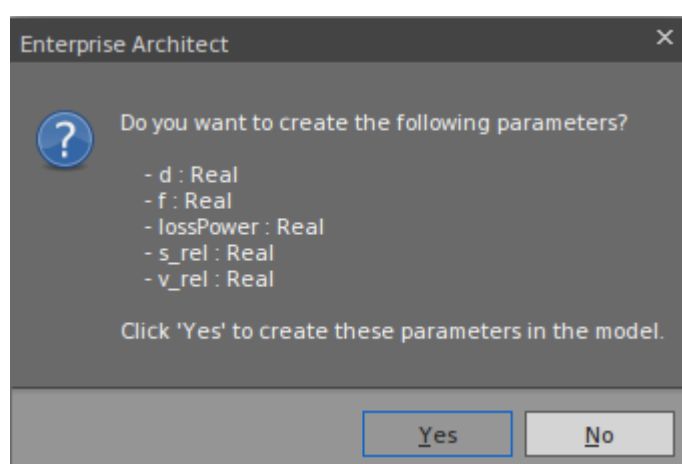
Other	Diagram Toolbox, SysML Block Definition page   Drag a Constraint Block icon and drop on a Block Definition diagram   'Edit Constraint Block' dialog displays
Context menu	Right-click on a ConstraintBlock   Edit ConstraintBlock

### Parsing Equations and create Parameters

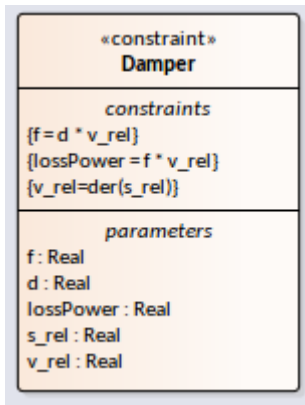
In this example, we type in three equations, and there are five parameters automatically extracted from the equations. The '+' sign at the start of the parameter indicates that these parameters are newly added (they currently do not exist in the model).



Click on the Close button; a message box will prompt you to select whether or not to create these parameters. Click on the Yes button to create the parameters in the model. Alternatively, you can right-click on the parameter and choose the 'Create Parameter' context menu option. In this way, we can create a single parameter.



This image shows the created ConstraintBlock *Damper*.

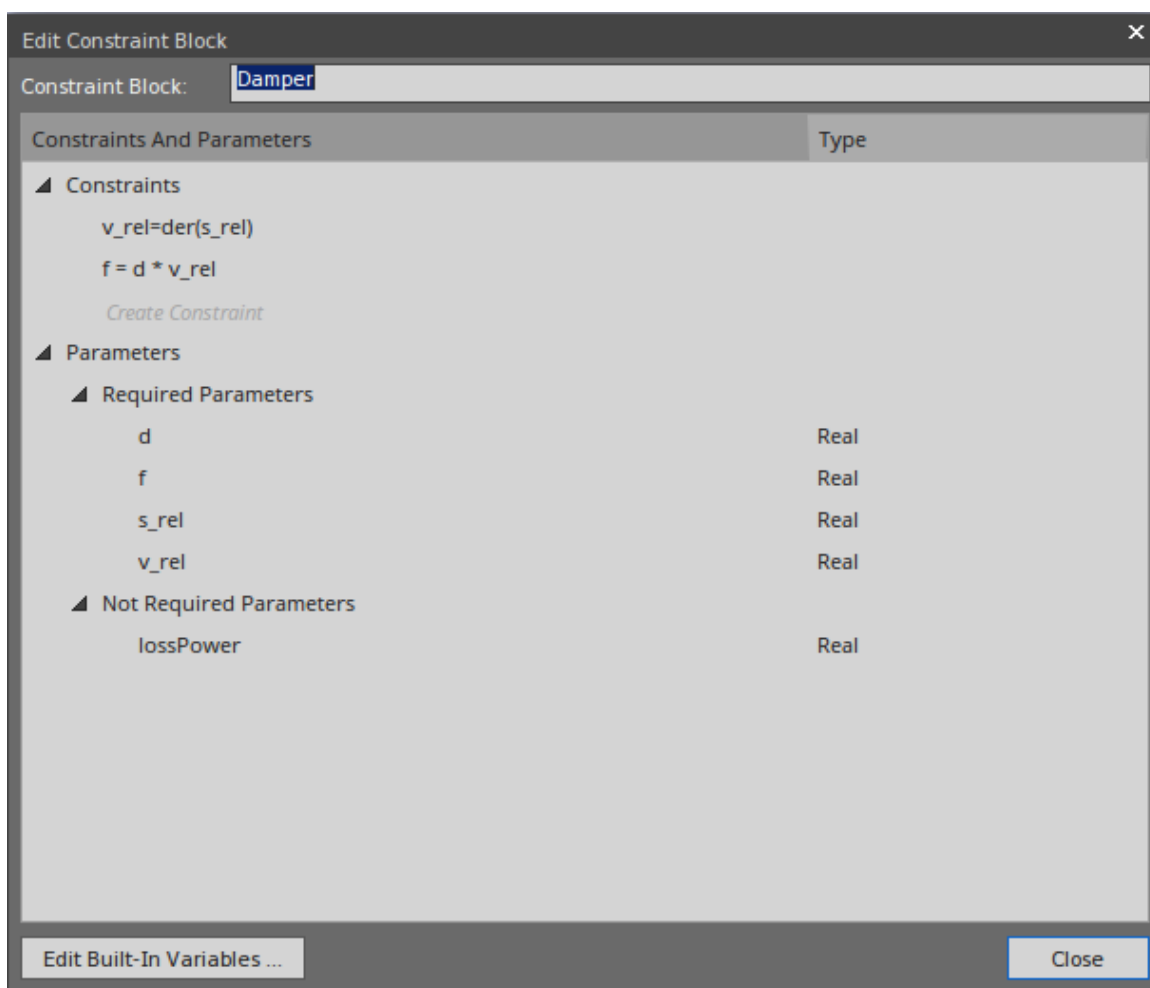


## Delete not required parameters

In our example, if we open the dialog and delete the constraint

$lossPower = f * v_{rel}$

'lossPower' will be reported as Not Required.



- You can right click on the parameter and choose the 'Delete Parameter' context menu option
- You can also right click on the 'Not Required Parameters' group and choose the 'Delete All Not-Required

Parameters' option

## Map Existing parameters to equation parsed parameters

In our example, suppose we open the dialog and change the constraint:

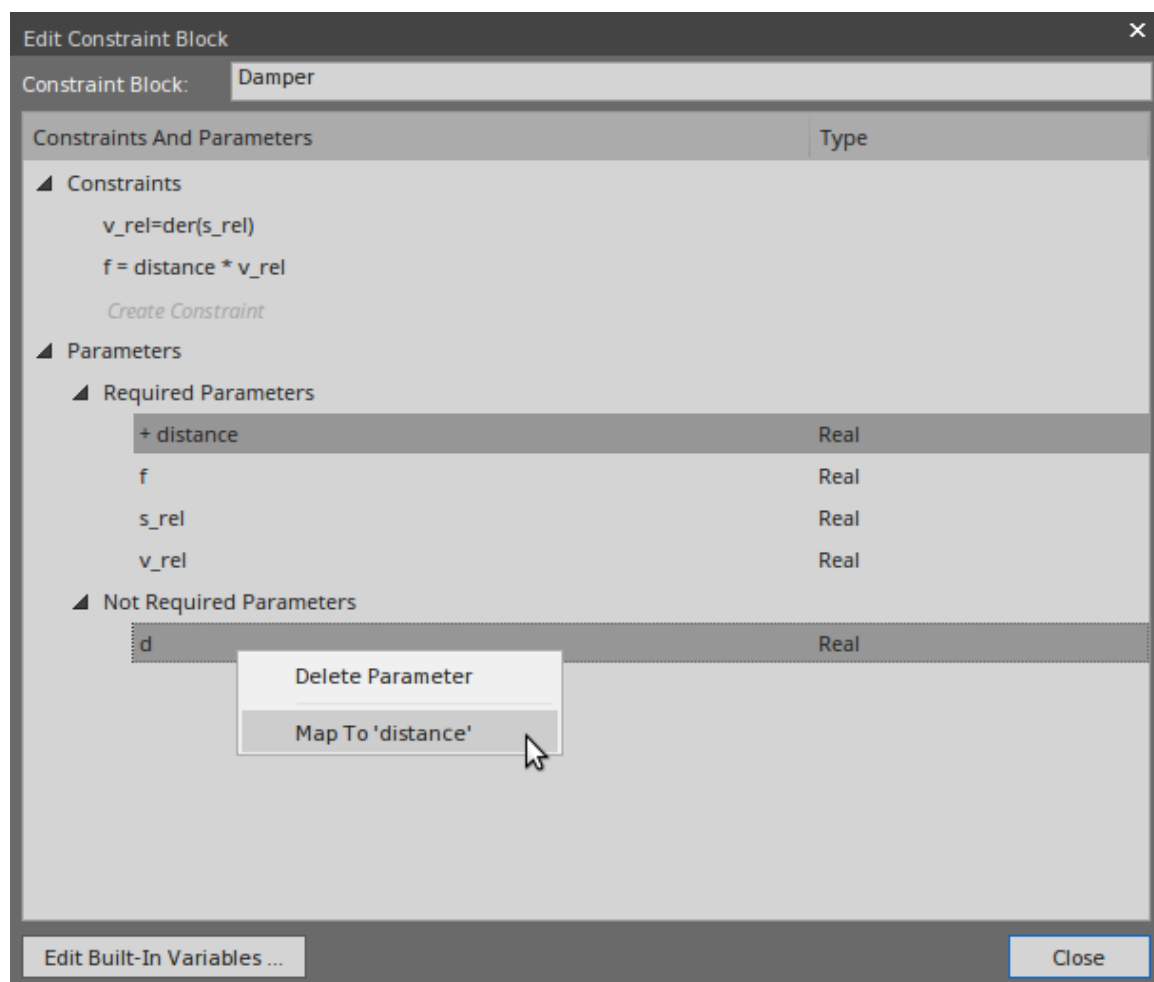
$$f = d * v\_rel$$

to

$$f = \text{distance} * v\_rel$$

These changes will occur:

- Parameter 'distance' is extracted and reported as a new parameter (it does not exist in the current model)
- Parameter 'd' is reported as 'Not Required'



You can delete 'd' and create 'distance'; however, there might be binding connectors connecting to parameter 'd' and all we want is to rename parameter 'd' to 'distance'. So, a better solution is to right-click on the 'Not Required' parameter 'd' and choose 'Map to 'distance''.

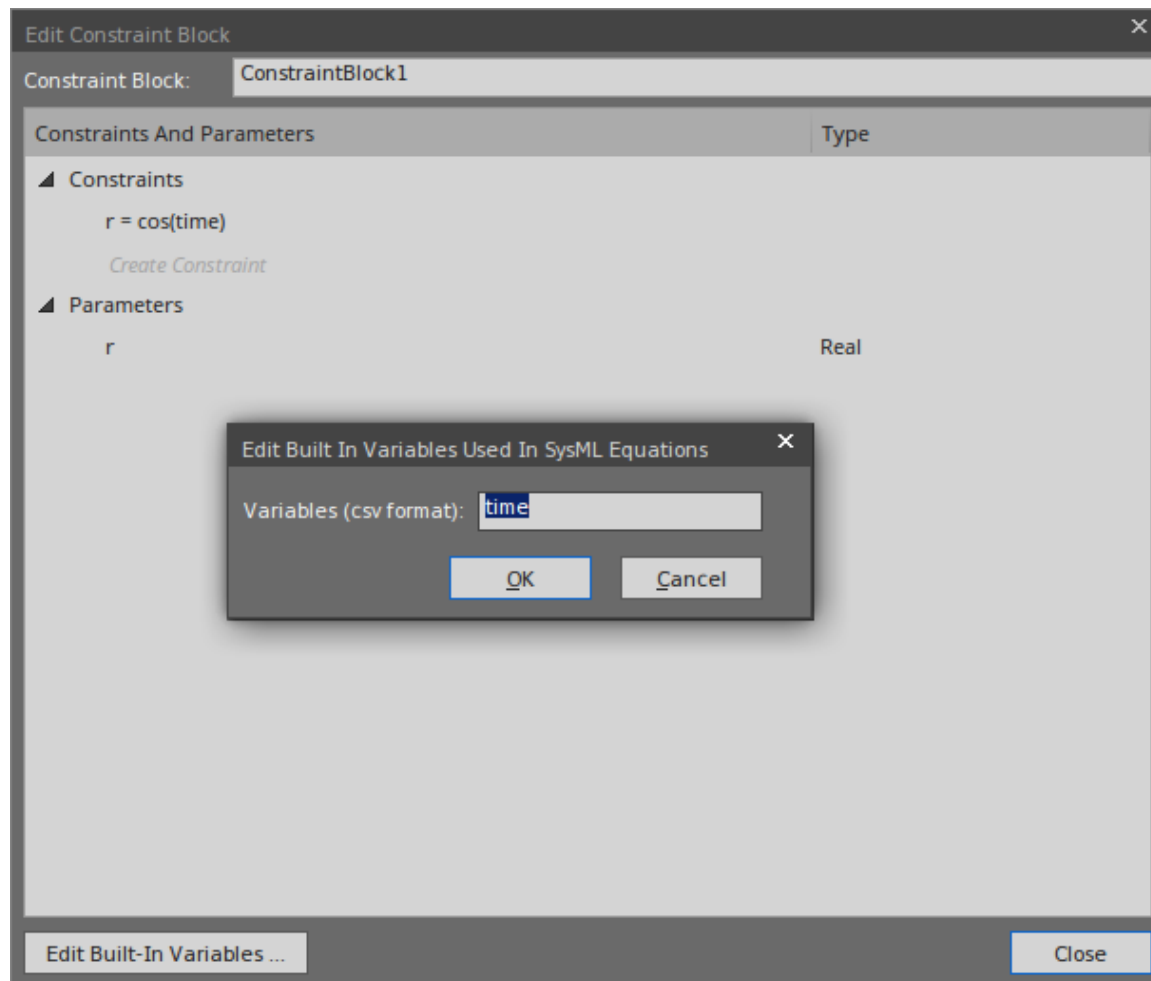
## Functions and Built-In Variables

When an equation contains functions, you must follow the format for functions: that is, function name followed immediately by a parenthesis.

When an equation contains variables that you don't want be extracted, you can configure them to be built-in variables. Click on the Edit Built-In Variables... button to enter comma separated variables.

For example:

If we input the equation ' $r = \cos(\text{time})$ ' and configure 'time' as a built-in variable, only one parameter (r) will be extracted.



# Bind Parameters of a ConstraintProperty


## Access

Context menu	In a diagram, click on an existing ConstraintProperty   Edit ConstraintProperty...
Other	In Project Browser   Drag a ConstraintBlock and drop it on a Parametric diagram A ConstraintProperty with all the parameters will be created

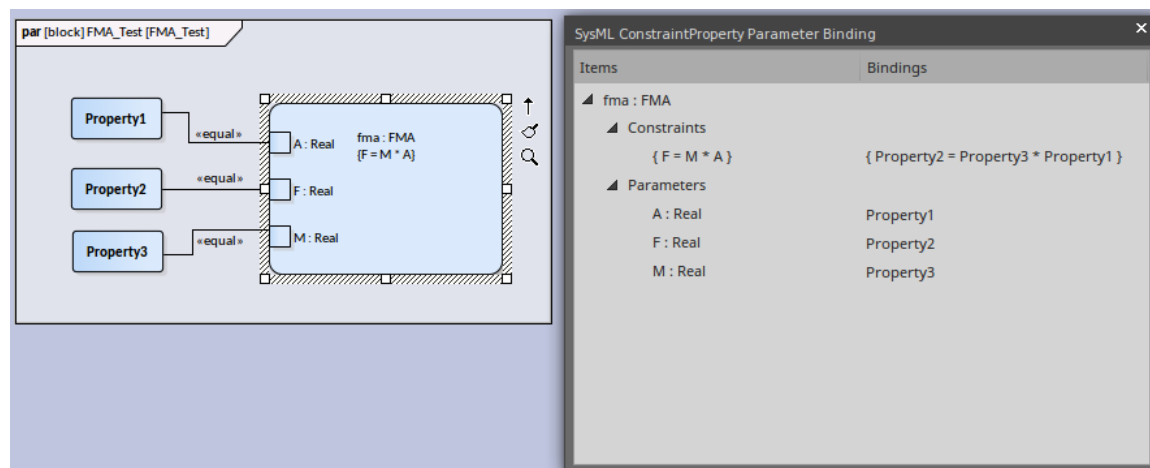
## Bind parameter to properties in context

In this example, we create the ConstraintBlock 'FMA' and use it as a ConstraintProperty 'fma' in the context of a block 'FMA\_Test', which contains three properties: 'Property1', 'Property2' and 'Property3'.

Right click on ConstraintProperty 'fma' and select the 'Edit ConstraintProperty...' context menu option to open the 'SysML ConstraintProperty Binding' dialog.

Click on the  button in the same row as a parameter to open the 'Hierarchy Properties Picker' dialog; choose a property to bind to the parameter.

After binding, the property will be shown on the diagram and a connector will connect it to the parameter of the constraint property.



# Systems Modeling Language (SysML)

To develop SysML models quickly and simply, you can use one of four versions of the MDG Technology for SysML, integrated with Enterprise Architect. The available versions correspond to SysML 1.1, SysML 1.2, SysML 1.3 and SysML 1.4/1.5.

This text is derived from the official OMG SysML site of the Object Management Group.

'The OMG Systems Modeling Language (OMG SysML™) is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. The language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models. SysML was developed in response to requirements developed jointly by the OMG and the International Council on Systems Engineering (INCOSE) by the diverse group of tool vendors, end users, academia, and government representatives.'

## Access

On the Diagram Toolbox, click on  to display the 'Find Toolbox Item' dialog and specify:

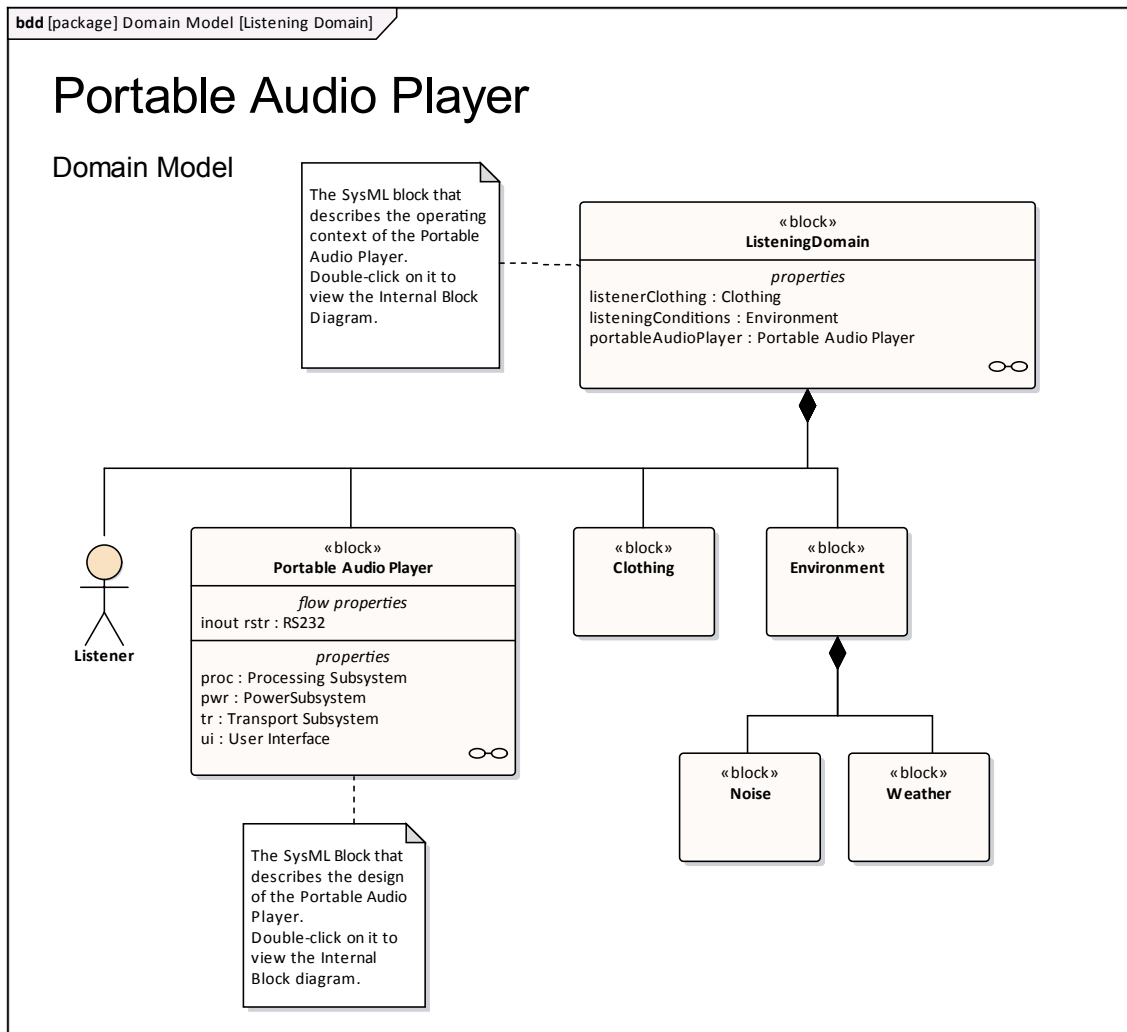
- SysML 1.1
- SysML 1.2
- SysML 1.3 or
- SysML 1.5

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5

## Example SysML Diagram

This is an example of a SysML Block diagram, which can also be found in the EAExample model under Systems Engineering > SysML Example: Portable Audio Player > Domain Model > Listening Domain.





## SysML Integration

Facilities	Detail
SysML In Enterprise Architect	<p>The MDG Technology for SysML provides:</p> <ul style="list-style-type: none"> <li>Each of the nine SysML diagram types, accessed through the 'New Diagram' dialog</li> <li>A collection of SysML pages in the Diagram Toolbox that contain the SysML elements and relationships for each of the supported diagram types</li> <li>SysML element and relationship entries in the 'Toolbox Shortcut Menu' and Quick Linker</li> <li>Simulation of SysML Parametric diagrams, which supports engineering analysis of critical system parameters including the evaluation of key metrics such as performance, reliability and other physical characteristics</li> </ul>
SysML Toolboxes	<p>The MDG Technology for SysML provides Diagram Toolbox pages that you can access through the 'Find Toolbox Item' dialog. If you enable SysML as the active technology, you can also open the SysML Toolbox pages by default.</p> <p>These sets of Toolbox pages are available:</p>

	<ul style="list-style-type: none"> <li>• 'Activity' contains the constructs required to construct SysML Activity models</li> <li>• 'Block Definition' contains the constructs required to design SysML blocks, constraint blocks, data and value types</li> <li>• 'Interaction' contains the constructs required to construct SysML interactions and Sequence diagrams</li> <li>• 'Internal Block' contains the constructs required to design SysML block compositions within Internal Block diagrams</li> <li>• 'Model Elements' contains the constructs required to build SysML models, Package structures and views</li> <li>• 'Parametrics' contains the constructs required to construct SysML Parametric diagrams using constraint blocks</li> <li>• 'Requirements' contains the constructs required to build SysML Requirements models</li> <li>• 'StateMachine' contains the constructs required to build SysML StateMachines</li> <li>• 'Use Case' contains the constructs required to build SysML Use Case models</li> </ul> <p>With the 'Model Elements' pages there is a set of SysML Common elements and relationships; these are also provided with the other 'SysML' Toolbox pages if the active technology is set on the Default Tools toolbar to 'SysML 1.1', 'SysML 1.2', 'SysML 1.3', 'SysML 1.4' or 'SysML 1.5'.</p>
Upgrade SysML Models	You can migrate a SysML model (or part of a model) to a later SysML version, using the Automation Interface.
Disable SysML	If you prefer not to use SysML in Enterprise Architect, you can disable it (and subsequently re-enable it) using the 'MDG Technologies' dialog (using the 'Specialize > Technologies > Manage' ribbon option).

## Notes

- Systems Modeling Language (SysML) is provided with the Unified and Ultimate editions of Enterprise Architect
- SysML 1.5 is virtually identical to SysML 1.4 so the versions are supported and processed as the same thing and you do not need to upgrade your SysML 1.4 models; references to the latest version of SysML have, however, been updated to '1.5'

## SysML Activity Toolbox

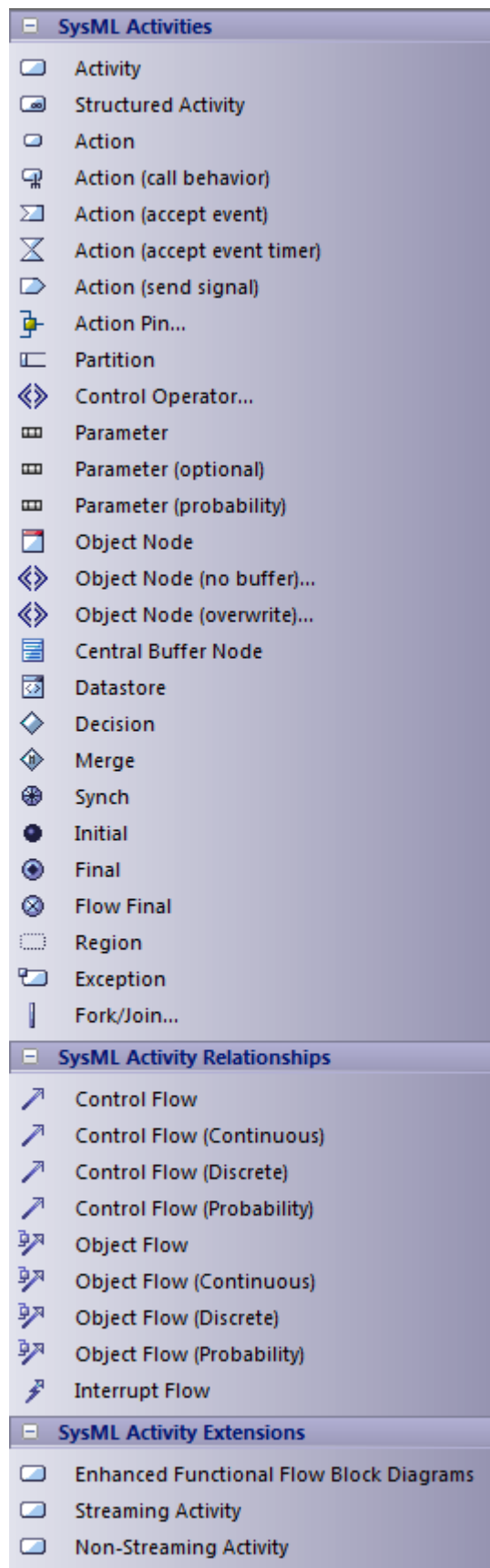
When you are constructing SysML models, you can populate the Activity diagrams using the icons on the 'SysML Activity' pages of the Diagram Toolbox.

The element that owns the Activity diagram is automatically represented by a diagram frame enclosing the Activity diagram elements. You can:


- Hide the frame by right-clicking on the diagram and selecting the 'Hide Diagram Frame' option (and show it again by selecting the 'Show Diagram Frame' option)
- Make the frame selectable to move or resize it, by right-clicking on it and selecting the 'Selectable' option
- Create structural elements (e.g. ActivityParameters if the owner is an Activity) on the frame and create connectors between them and other elements on the diagram

If set to non-selectable, the frame will auto-resize to fit the bounds of the diagram, expanding from its default size but not shrinking smaller.

Note that diagrams showing Diagram Frames applied under release 14.0 or higher of Enterprise Architect will draw the parent object on the diagram when opened in a release prior to release 14.0.



## Access

On the Diagram Toolbox, click on  to display the 'Find Toolbox Item' dialog and specify 'SysML n.n Activity' (whichever version you are using).

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5

## SysML Activities

Item	Action
Activity	Defines a SysML Block of executable behavior as a UML Activity.
Structured Activity	Defines a SysML Block of executable behavior as a UML Structured Activity.
Action	Declares a unit of execution in an Activity as a UML Action.
Action (call behavior)	Declares a unit of execution that calls another behavior.
Action (accept event)	Declares a unit of execution that accepts an event raised by the system.
Action (accept event timer)	Declares a unit of execution that accepts an event raised by a time epoch.
Action (send signal)	Declares a unit of execution that sends a signal as an event.
Action Pin	Defines the data values passed out of and into an Action. See also: <a href="#">Action Pin</a>
Partition	Creates an Activity Partition to group execution elements according to the node responsible for their execution.
Control Operator	Controls the execution of an Activity.
Parameter	Provides access to input and output objects within the Activity.
Parameter (discrete)	Defines a parameter with a rate of flow where the increment of time between items is non-zero.
Parameter (continuous)	Defines a parameter with a rate of flow where the increment of time between items approaches zero.
Parameter (optional)	Defines a parameter whose contents are optional in the Activity's execution.
Parameter (probability)	Tags a parameter with the probability of the parameter being in use in the Activity.
Object Node	Declares a variable in the Activity, typed by a ValueType, DataType or Block.
Object Node (no buffer)	Declares an ObjectNode in an Activity that discards unconsumed tokens.

Object Node (overwrite)	Declares an ObjectNode in an Activity that overwrites tokens.
Central Buffer Node	Declares an ObjectNode that stores tokens for consumption throughout the Activity.
Datastore	Defines permanently stored data. See also: <a href="#">Datastore</a>
Decision	Creates a branch of control in an Activity, based on a decision.
Merge	Merges two or more Activity control branches.
Synch	Establishes a rendezvous point for two or more Activity flows, in order to synchronize their execution in the Activity.
Initial	Declares the start of an Activity's execution.
Final	Declares the end of an Activity's execution, and the termination of the Activity.
Flow Final	Declares the end of an Activity's execution path without terminating the Activity.
Region	Groups a subset of an Activity into a common execution context.
Exception	Declares a node of execution that happens outside the normal flow of execution of an Activity.
Fork/Join	Simultaneously branches / joins a set of Control or Object Flows.

## SysML Activity Relationships

Item	Action
Control Flow	Establishes a flow of logic between two Activity nodes.
Control Flow (Continuous)	Declares a continuous control flow.
Control Flow (Discrete)	Declares a discrete control flow.
Control Flow (Probability)	Tags a control flow with a probability of the likelihood of the flow's traversal.
Object Flow	Establishes a flow of objects (data) between two Activity nodes.
Object Flow (Continuous)	Declares a continuous object flow.
Object Flow (Discrete)	Declares a discrete object flow.
Object Flow (Probability)	Tags an object flow with the probability of the flow's traversal.
Interrupt Flow	Declares a control flow that interrupts flow within a Region.

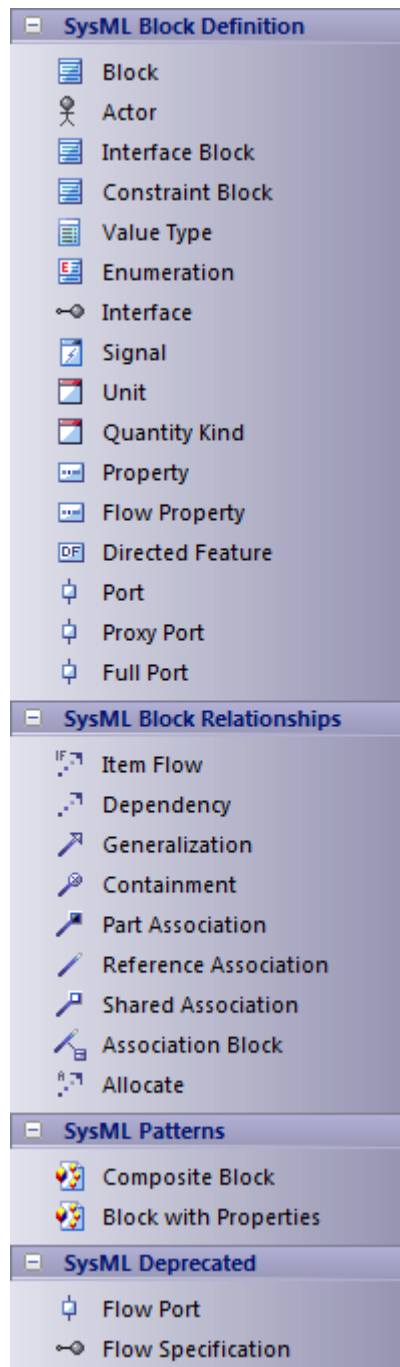
## SysML Activity Extensions

Item	Action
Enhanced Functional Flow Block Diagrams	Action: Declares an Activity used to contain an Enhanced Functional Flow Block diagram (EFFBD).
Streaming Activity	Declares an Activity where the flow of tokens passes through its parameters continuously throughout the Activity's execution.
Non-Streaming Activity	Declares an Activity where the flow of tokens passes through its parameters at the start of the Activity's execution.

# SysML Block Definition Toolbox


When you are constructing SysML models, you can populate the Block Definition diagrams using the icons on the SysML Block Definition pages of the Diagram Toolbox.

You can also generate Property elements on the Block, based on the Association relationships created for the Block element. These Properties (or Parts) are initially created in the Project Browser, but you can quickly render them on the Internal Block diagram for the Block.



## Access



On the Diagram Toolbox, click on  to display the 'Find Toolbox Item' dialog and specify 'SysML n.n Block Definition' (whichever version you are using).

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5

## SysML Block Definition Objects

Item	Action
Block	Defines a composite system entity in SysML.
Actor	Represents a user that interacts with one or more SysML systems.
Interface Block	A specialized kind of block that has no behaviors or internal parts, which is used to type Proxy Ports.
Constraint Block	Defines a composite constraint as a system of parametric equations.
ValueType	Defines a SysML quantity, expressed as a measurable dimension with specific units.
Enumeration	Defines a data type as a set of symbols or values.
Interface	Defines an element that describes a specification of an interaction point with properties and methods.
Signal	Defines a SysML message, containing attributes, exchanged between system blocks in an interaction.
Instance Specification	TODO
Unit	Represents a standard unit of measure in SysML.
QuantityKind	Identifies a measurable quantity in SysML.
Property	Describes the decomposition of a SysML Block in the context of its whole, using instances of reusable SysML Blocks.
Flow Property	Creates a single kind of Flow element that can flow to or from a block.
Directed Feature	<p>Generates an operation if dropped on a Block, or a DirectedFeature (Property) element if dropped on the diagram, both of which indicate that the associated Block either owns (provided) the feature, uses (required) the feature owned by another Block, or both (providedrequired).</p> <p>The direction value is defined in the 'SysML1.n' page of the Element Properties window, or the Tagged Values window for the element or the operation.</p>

Port	Describes a structural interaction point of a SysML Block, which in turn connects interacting parts of a block.
Proxy Port	Exposes features of the owning Block or its internal parts.
Full Port	Specifies an element of the system separate from the owning Block or its internal parts.

## SysML Block Definition Relationships

Item	Action
Item Flow	Specifies the items that flow across a connector in an interaction point. Used in the same way as UML Information Flows. See <a href="#">Using Information Flows</a>
Dependency	Establishes a traceable relationship describing how one element is dependant upon another.
Generalization	Describes an element as a specialized descendant of another element, containing additional properties and behavior.
Containment	Graphically displays ownership of one element within a parent element.
Part Association	Describes the characteristics of a connection between a SysML Block and its internal parts, such as the multiplicity and type.
Reference Association	Describes the characteristics of a connection between separate SysML Blocks, such as the multiplicity and type.
Shared Association	Describes the characteristics of a common connection between SysML Blocks, such as the multiplicity and type.
Association Block	Describes an Association connector that is defined by a SysML Block.
Connector	Structural connection between Properties and Ports.
Allocate	Connects model elements to formalize a refinement of behavior, structure, constraints or design expectations.

## SysML Patterns

Item	Action
Composite Block	A Pattern that creates a Composite Block, made up of Blocks related by Aggregation relationships.

Block with Parts	A Pattern that creates a Composite Block made up of nested parts.
------------------	---

## SysML Deprecated

Item	Action
Flow Port	Describes what flows in and out of interacting SysML Blocks. This element type is deprecated. Instead create a Port that is typed by an Interface Block that owns Flow Properties.
Flow Specification	Defines a set of flow properties that correspond to individual pieces of a common interaction point. The element type is deprecated. Instead create an Interface Block that owns Flow Properties.

# Generate Properties From Block Associations

On a SysML Block diagram, an Association between two Block elements has ends that are represented by properties. If an Association end is navigable, the property that represents it is owned by the Block element at the other end of the Association.

In Enterprise Architect you can automatically create Part elements to represent these owned properties, in one of two ways:

- Create a Part Association between two Blocks, which will generate an anonymous Part property on the target Block
- Create a Reference Association between two Blocks, which will generate an anonymous Reference property on the target Block
- Create a directed Association ('Start > View > Preferences > Links > select 'Association default = source --> target' checkbox'), which generates an anonymous Reference property on the source Block element
- Create an Association between two Blocks, give one or both of the Association End Roles a name, then go to the 'Internal Block Diagram' of the other end and select 'Synchronize Structural Elements' from the diagram's context menu (that is, after naming the target end role, go to the 'Internal Block Diagram' of the source element and Synchronize to create the property)

In both cases, the Part property is considered to be bound to the Association End property – they represent the same property, so changing one updates the other, either automatically or at the next synchronization; that is, if you change the Association Source Role name, multiplicity or Aggregation setting, the Part name, multiplicity and isReference setting are updated; if you change the Part details, the Association properties are updated.

If you want to change the binding of a Property to an Association, or bind existing Associations and Properties that are not yet bound to each other, right-click on the Property in:

- The Project Browser, and select 'Add | Bind to Connector Role', or
- The Internal Block diagram and select 'Advanced | Bind to Connector Role'

In each case the 'Choose Connector Role to Bind' dialog displays, listing the Associations issuing from the parent Block element. Select the Association to bind the Property to, and click on the OK button.

If you subsequently delete an Association that is bound to a property, when you save the diagram you are prompted to confirm whether to also delete the property or keep it, unbound to a connector. If the property element is locked, it cannot be deleted.

## Notes

- You can create an Association connector on a SysML Block diagram by dragging the Reference Association icon from the Toolbox or selecting the Association option from the Quick Linker arrow drop-down list
- Associations have direction when it is explicitly selected on the connector Properties, when the connector has assigned Source and Target roles, and/or when the Association default = source --> target checkbox is selected on the 'Preferences' dialog

## Block Element Compartments

SysML elements such as Blocks and Constraint Blocks can use compartments to display child or related elements. These compartments can assist in easily identifying different types of properties owned by a Block, to see what Requirements a Block satisfies, or to see how different types are related to each other.

You can show or hide these compartments using the

- 'Compartment Visibility' dialog (press Ctrl+Shift+Y on the parent element on a diagram) or
- 'Compartments' tab of the docked Properties window for diagrams (Alt+1) or
- 'Diagram Properties' dialog, 'Element' page ('Design > Diagram > Edit > Properties > Elements' ribbon option)

Compartments that display child elements will be visible by default, whereas compartments that display related elements (elements linked by connectors) are hidden by default.

Elements will only appear in compartments if they do not also appear as elements on the diagram. Also, a compartment will only be displayed if at least one matching element exists for it. So, for example, the 'flowport' compartment will not be displayed unless the parent owns at least one Port that has a «flowPort» stereotype and is not on the diagram. If a compartment is not listed, it may be necessary to locate and remove from the diagram any related or child elements, save the diagram, and reload the diagram to refresh the compartment list.

The table *SysML Block Child Element Compartments* provides a list of compartments, including which element/connector types are required for them to be displayed.

### SysML Block child element compartments

These compartments are displayed according to the child elements of the Block.

Compartment Name	Child Element Types Listed
adjunct	Lists Parts with the «AdjunctProperty» stereotype.
bound reference	Lists Parts with the «BoundReference» stereotype.
classifier behavior	Shows the behavioral classifier (Interaction, StateMachine or Activity) if set.
constraints	Lists Parts with the «constraintProperty» stereotype.
flow ports	Lists Ports with the «flowPort» stereotype.
flow properties	Lists Parts with the «flowProperty» stereotype.
full ports	Lists Ports with the «fullPort» stereotype.
owned behaviors	Lists Behavioral elements (Interactions, StateMachines and Activities) owned by this block.
participants	Lists Parts with the «participantProperty» stereotype.
parts	Lists Properties created by adding a Part Association connector between blocks.
ports	Lists any other Ports not listed in the previous compartments.
properties	Lists Parts without a stereotype.

proxy ports	Lists Ports with the «proxyPort» stereotype.
references	Lists Parts for which the isReference Tagged Value is set to true.
values	Lists Parts that are typed by a «valueType» element.
«stereotype»	Lists Parts with stereotypes other than those identified above.
directed features	Lists Parts with the «DirectedFeature» stereotype.

## SysML Block related element compartments

These compartments are displayed based on the relationships between a Block and other elements.

Compartment Name	Object Displayed
allocatedFrom	Shows the Source of a connector with the «allocate» stereotype.
allocatedTo	Shows the target of a connector with the «allocate» stereotype.
refines	Shows the target of a connector with the «refines» stereotype.
satisfies	Shows the target of a connector with the «satisfy» stereotype.
tracedFrom	Shows the target of a connector with the «trace» stereotype.
verifies	Shows the target of a connector with the «verify» stereotype.

## SysML Constraint Block element compartments

In addition to the compartments that a Block element can display, a Constraint Block can also display:

Compartment Name	Child Element Types Listed
parameters	Shows any Part that does not have the «constraintProperty» stereotype.

## Show Direction on SysML Ports

In SysML you can show Ports, Full Ports and Proxy Ports with arrows indicating the flow direction.

### Create a new Port with direction arrows

Step	Action
1	Create a Port, Full Port or Proxy Port from the 'SysML Block Definition' pages of the Diagram toolbox.
2	Right-click on the Port and select 'Advanced   Set Property Type'. The 'Select <Item>' dialog opens.
3	Select a Block or Interface Block that owns one or more Flow Property elements. Click on the OK button. The Port will automatically display arrows based upon the aggregated 'direction' Tagged Values of all the Flow Property elements owned by the Block or Interface Block that is set as the Property Type. For example, if the Block contains an 'in' Flow Property and an 'out' Flow Property, the Port typed with this Block will display an 'inout' arrow.

## Nested Ports in SysML

This statement is derived from the SysML 1.5 Specification:

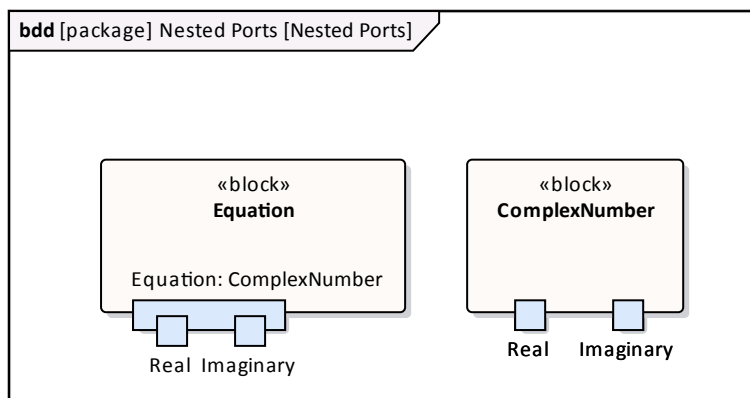
'Ports nest other ports in the same way that blocks nest other blocks. The type of the port is a block (or one of its specializations) that also has ports.'

For example, a complex number is made up of two members (of type Real) - the real value and the imaginary value. This is represented by the Block `ComplexNumber` with two Ports, `Real` and `Imaginary`. An `Equation` Block has a Port that takes a `ComplexNumber`, but we might want to connect the real and imaginary portions of that variable to different sources. Therefore we have to show them, on the `Equation` Port, in order to connect them up.

This facility is available in Enterprise Architect for all versions of SysML and for UML.

### To create a representation of this example

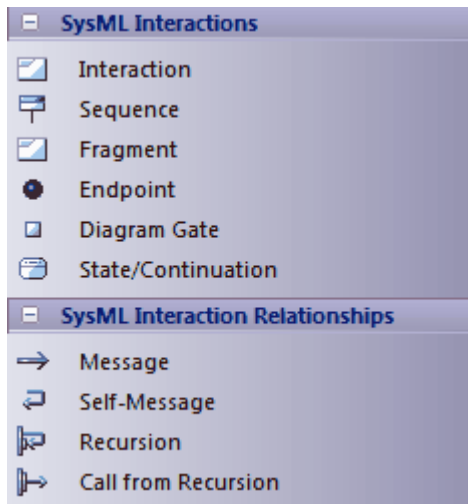
1. Create a SysML Block named 'Equation' with a Port named 'Equation'.
2. Create a SysML Block named 'ComplexNumber' with Ports named 'Real' and 'Imaginary'.
3. Display the Properties window (Alt+1) for the Port 'Equation' and select the 'Property' tab.
4. In the 'Type' field, click on the drop-down arrow and select the 'Select Type' option, then locate and select the 'ComplexNumber' Block element; this sets the 'Type' field to 'ComplexNumber'.
5. Right-click on Port 'Equation' and select the 'Features | Interaction Points...' option. The Features window displays at the 'Interaction Points' tab.
6. Select the 'Show Owned/Inherited' checkbox. Ports 'Real' and 'Imaginary' are shown in the list panel.
7. Select the checkboxes against 'Real' and 'Imaginary'. The two Ports are now nested in Port 'Equation'.





## SysML Interaction Toolbox

When you are constructing SysML models, you can populate the Interaction and Sequence diagrams using the icons on the SysML Interaction pages of the Diagram Toolbox.



### Access

On the Diagram Toolbox, click on  to display the 'Find Toolbox Item' dialog and specify 'SysML n.n Interaction' (whichever version you are using).

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5

### SysML Interaction Objects

Item	Action
Interaction	Defines a SysML Block of executable behavior as a UML Interaction.
Sequence	References an instance of a SysML Block as a Lifeline in the Interaction.
Fragment	Declares a portion of an interaction as a group with specific behavior semantics.
Endpoint	Creates an entry or exit point for the Interaction.
Diagram Gate	Creates an endpoint for the interaction, which bridges between nested interactions.
State/Continuation	Constrains the Interaction with assertions of the state that the lifeline is expected to be in.

## SysML Interaction Relationships

Item	Action
Message	Describes a message exchange between two lifelines in an Interaction.
Self-Message	Describes a message exchange between a lifeline and itself in an Interaction.
Recursion	Describes a recursive message exchange between a lifeline and itself in an Interaction.
Call from Recursion	Describes a message exchange between two lifelines within a recursive exchange.

## SysML Internal Block Toolbox

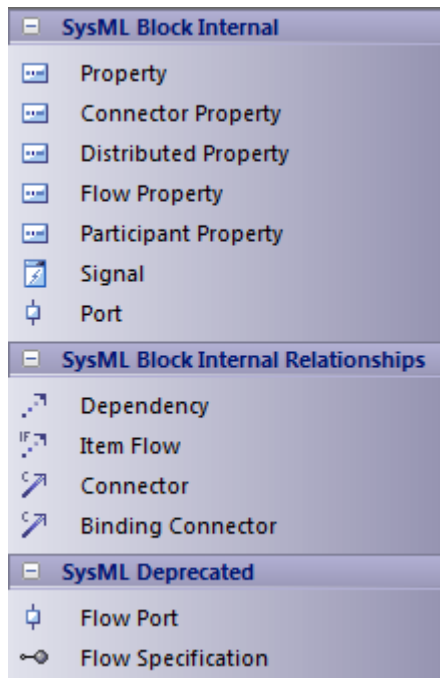
When you are constructing SysML models, you can populate the Internal Block diagrams using the icons on the 'SysML Block Internal' pages of the Diagram Toolbox.

The Block that owns the Internal Block diagram is automatically represented by a diagram frame enclosing the Internal Block diagram elements. You can:

- Hide the frame by right-clicking on the diagram and selecting the 'Hide Diagram Frame' option (and show it again by selecting the 'Show Diagram Frame' option); if the diagram frame is shown, the 'Diagram Frame' settings on the 'Start > View > Preferences > Diagram' page will be ignored
- Make the frame selectable to move or resize it, by right-clicking on it and selecting the 'Selectable' option
- Create Ports on the frame and create connectors between them and any internal structural elements
- Generate Property elements on the diagram inside the Block frame, based on the Associations that the Block element has on the Block diagram

If set to non-selectable, the frame will auto-resize to fit the bounds of the diagram, expanding from its default size but not shrinking smaller.

Note that diagrams showing Diagram Frames applied under release 14.0 or higher of Enterprise Architect will draw the parent object on the diagram when opened in a release prior to release 14.0.



### Access

On the Diagram Toolbox, click on  to display the 'Find Toolbox Item' dialog and specify 'SysML n.n Internal Block' (whichever version you are using).

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5

## SysML Block Internal

Item	Description
Property	Describes the decomposition of a SysML Block in the context of its whole using instances of reusable SysML Blocks.
Connector Property	Creates an instance of an Association Block.
Distributed Property	Creates a stereotype of Property, to apply a probability distribution to the values of the property.
Flow Property	Creates a single kind of Flow element that can flow to or from a block.
Participant Property	Creates the end of a connector owned by an Association Block.
Directed Feature	Creates a Feature that might be required, provided or both.
Adjunct Property	Creates a Property for which the value is constrained to the value of a connector typed by an Association Block, Call Action, Object Node, variable, parameter, Interaction Use or SubMachine State.
Bound Reference	Creates a Property with the <<BoundReference>> stereotype. Such properties will have binding connectors to highlight their use as constraining other properties.
End Path Multiplicity	Creates a Property with the EndPathMultiplicity stereotype. Such properties will be related by redefinition to properties that have BoundReference applied.
Classifier Behavior Property	Creates a Property with the ClassifierBehaviorProperty stereotype. Such properties will constrain their values to be the executions of classifier behaviors.
Signal	Defines a SysML message, containing attributes, exchanged between system Blocks in an interaction.
Port	Describes a structural interaction point of a SysML Block which, in turn, connects between interacting parts of a block.

## SysML Block Internal Relationships

Item	Description
Dependency	Establishes a traceable relationship describing how one element is dependant upon another.
Item Flow	Specifies the items that flow across a connector in an interaction point. Used in the same way as UML Information Flows.
Connector	Establishes Communication links between parts.

Binding Connector	Establishes a connection between two parts in a system decomposition.
-------------------	---

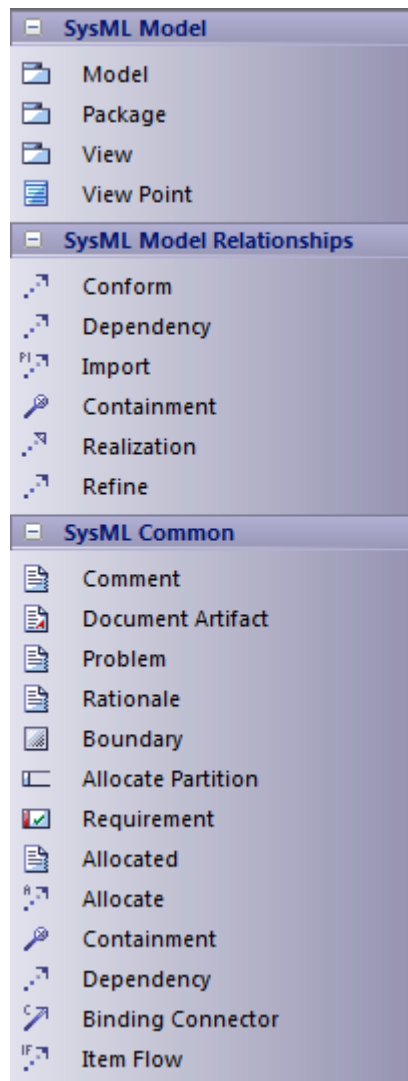
## SysML Deprecated

Item	Description
Flow Port	Describes what flows in and out of interacting SysML Blocks. This element type is deprecated. Instead create a Port that is typed by an Interface Block that owns Flow Properties.
Flow Specification	Defines a set of flow properties that correspond to individual pieces of a common interaction point. The element type is deprecated. Instead create an Interface Block that owns Flow Properties.


# SysML Model Elements Toolbox

When you are constructing SysML models, you can populate the diagrams with Model, Package and View elements using the icons on the 'SysML Model' pages of the Diagram Toolbox.

The SysML Model toolbox includes a page of SysML Common element and relationship icons. You can add this page to all Diagram Toolboxes so that it is always available regardless of what type of diagram you are using; to do this, set the SysML 1.5 Technology to 'Active' ('Specialize > Technologies > Manage : SysML 1.5 : Set Active' ribbon path).



## Access

On the Diagram Toolbox, click on  to display the 'Find Toolbox Item' dialog and specify 'SysML n.n Model Elements' (whichever version you are using).

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5

## SysML Model Element Objects

Item	Description
Model	Creates a Package containing a SysML Model.
Model Library	Creates a Package containing a SysML Model Library.
View	Creates a stereotyped Class that defines a SysML View of a system, from the perspective of a SysML View Point.
View Point	Creates a stereotyped Class that defines a SysML View Point, which specifies the rules and conventions for the construction and use of Views.
Stakeholder	Creates a stereotyped Class that defines a SysML Stakeholder.
Package	Groups model constructs in a single unit of containment.

## SysML Model Relationships

Item	Description
Conform	Establishes a conformance dependency of a View to the defining View Point.
Dependency	Establishes a traceable relationship describing how one element is dependant upon another.
Import	Represents a reuse of elements from one model Package in another.
Containment	Graphically displays ownership of one element within a parent one.
Realization	Identifies a design fulfillment of a specification between elements.
Refine	Represents a refinement of one element by another.
Expose	Relates a View to another model element.

## SysML Common

Item	Description
Comment	Creates a textual annotation that can be attached to a set of elements of any other type.

	The attachment is created separately, using a Notelink connector.
Document Artifact	Attaches a linked document to the diagram by associating this element with the document.
Element Group	Creates a textual annotation that indicates how many model elements it is attached to.
Problem	A stereotyped Comment that documents the failure of model elements to satisfy a requirement.
Rationale	A stereotyped Comment that documents the justification for decisions.
Boundary	Defines a conceptual boundary, to visually group logically related elements.
Allocate Partition	A stereotyped Activity Partition that contains elements deemed to be allocated to the classifier of the partition.
Requirement	Specifies the capabilities of the system, or the conditions that it should satisfy.
Allocated	A stereotyped Comment that defines the source element being allocated to the target element in an Allocate relationship.
Allocate	<p>A stereotyped Abstraction that relates model elements to formalize a refinement of behavior, structure, constraints or design expectations.</p> <p>The Allocate relationship points from the element being allocated to the element that is the target of the allocation.</p> <p>The system provides an 'Allocations' search that lists all Allocate abstractions in tabular format, showing the 'To' and 'From' elements. Select the 'Start &gt; Explore &gt; Search &gt; Model' ribbon option, then select 'SysML 1.4' as the Search Category; 'Allocations' defaults as the Search Type.</p> <p>If a SysML Block element has any Allocate relationships to or from elements that are not visible on the same diagram, those elements can be listed in 'AllocatedTo' and 'AllocatedFrom' compartments of the Block element on the diagram. Press Ctrl+Shift+Y to display the 'Compartment Visibility' dialog and select the 'Allocatedto' and/or 'Allocatedfrom' checkboxes.</p> <p>Alternatively, you can list the hidden elements on a linked Note. Create a Note element on the diagram and link it to the visible Block element with a Notelink connector. Right-click on the connector, select the 'Link this Note to an Element feature' option and, on the 'Link note to element feature' dialog, click on the drop-down arrow on the 'Feature Type' field and select 'AllocatedTo' or 'AllocatedFrom'. Click on the OK button; the names of the linked elements are now displayed in the Notes element. (To show both 'To' and 'From' elements, create a separate Note for each type.)</p> <p>For either compartments or Notes, you should save the diagram just before setting up the facility, and possibly reload the diagram to activate the facility. If there are no Allocate relationships, or the related elements are on the diagram, the options are not available.</p>
Containment	Graphically displays ownership of one element within a parent one.
Dependency	Establishes a traceable relationship describing how one element is dependant upon another.



Binding Connector	A stereotyped Connector that establishes a connection between two parts in a system decomposition.
Item Flow	A stereotyped Information Flow that specifies the items that flow across a connector in an interaction point.

## SysML Parametrics Toolbox

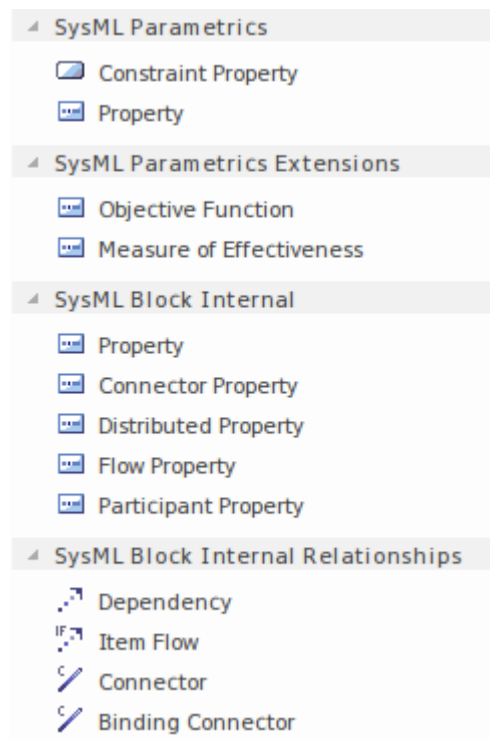
When you are constructing SysML models, you can populate the SysML Parametric diagrams with constraint blocks, using the icons on the 'SysML Parametrics' pages of the Diagram Toolbox.

The Block that owns the Parametric diagram is automatically represented by a diagram frame enclosing the Parametric diagram elements. You can:


- Hide the frame by right-clicking on the diagram and selecting the 'Hide Diagram Frame' option (and show it again by selecting the 'Show Diagram Frame' option)
- Make the frame selectable to move or resize it, by right-clicking on it and selecting the 'Selectable' option
- Create Ports and Parts on the frame and create connectors between them

If set to non-selectable, the frame will auto-resize to fit the bounds of the diagram, expanding from its default size but not shrinking smaller.

Note that diagrams showing Diagram Frames applied under release 14.0 or higher of Enterprise Architect will draw the parent object on the diagram when opened in a release prior to release 14.0.



### Access

On the Diagram Toolbox, click on  to display the 'Find Toolbox Item' dialog and specify 'SysML n.n Parametrics' (whichever version you are using).

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5

### SysML Parametrics Objects

Item	Description
Constraint Property	Instantiates a Constraint Block for use in a Parametric diagram.
Property	Defines a SysML property typed by a DataType, ValueType or Block.

## SysML Parametrics Extensions

Item	Description
Objective Function	Defines a SysML Constraint Block for use as an objective function to evaluate Measures of Effectiveness (MOEs).
Measure of Effectiveness	Defines a SysML property for use as a Measure of Effectiveness (MOE).

## SysML Block Internal

Item	Description
Property	Describes the decomposition of a SysML Block in the context of its whole using instances of reusable SysML Blocks.
Connector Property	Creates an instance of an Association Block.
Distributed Property	Creates a stereotype of Property, to apply a probability distribution to the values of the property.
Flow Property	Creates a single kind of Flow element that can flow to or from a block.
Participant Property	Creates the end of a connector owned by an Association Block.

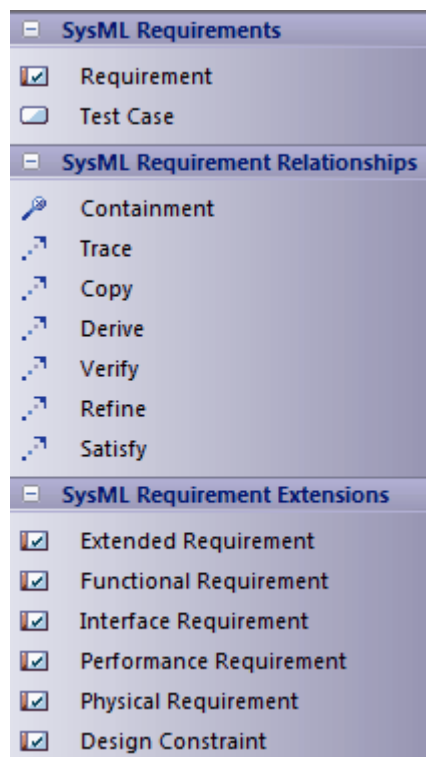
## SysML Block Internal Relationships

Relationship	Description
Dependency	Establishes a traceable relationship describing how one element is dependant upon another.
Item Flow	Specifies the items that flow across a connector in an interaction point. Used in the same way as UML Information Flows.

Connector	Establishes Communication links between parts.
Binding Connector	Establishes a connection between two Parts in a system decomposition.

## SysML Requirements Toolbox

When you are constructing SysML models, you can populate the Requirements diagrams using the icons on the SysML Requirements pages of the Diagram Toolbox.



### Access

On the Diagram Toolbox, click on  to display the 'Find Toolbox Item' dialog and specify 'SysML n.n Requirements' (whichever version you are using).

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5

### SysML Requirement Objects

Page	Item
Requirement	Specifies the capabilities of the system, or the conditions that it should satisfy.
Test Case	Describes the verification of a Requirement through methods of inspection, analysis, demonstration or testing.

## SysML Requirement Relationships

Item	Description
Containment	Graphically displays ownership of one element within a parent element.
Trace	Declares a trace relationship between a SysML Requirement and another SysML element.
Copy	Declares a copy of one SysML Requirement by another.
Derive	Derives a SysML Requirement from another.
Verify	Declares a verification of a SysML Requirement by another SysML element.
Refine	Declares a refinement of a SysML Requirement by another SysML element.
Satisfy	Declares that the SysML Requirement is satisfied by another SysML element.

## SysML Requirement Extensions

Item	Description
Extended Requirement	Extends a SysML Requirement with additional Tag properties.
Functional Requirement	Declares a SysML Requirement that describes the operation, or behavior, that the system must perform.
Interface Requirement	Declares a SysML Requirement that describes how the system connects, or interfaces with, other systems.
Performance Requirement	Declares a SysML Requirement that describes how the system performs against defined capabilities or conditions.
Physical Requirement	Declares a SysML Requirement that describes the physical characteristics, or physical constraints, of the system.
Design Requirement	Declares a SysML Requirement that specifies a constraint on the implementation of the system.

## Notes

- SysML Requirements contain the Tagged Values 'Text' and 'ID', the values of which are not immediately visible in the Tagged Values window; you can see the values more easily if you have the Summary View window open (click

on the 'Start > Explore > Properties > Summary' ribbon option) when you click on these elements

## SysML StateMachine Toolbox

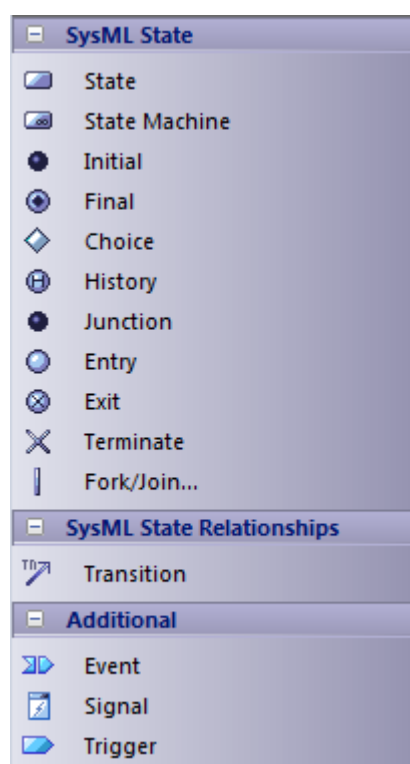
When you are constructing SysML models, you can populate the StateMachine diagrams using the icons on the 'SysML StateMachine' pages of the Diagram Toolbox.

The Block that owns the StateMachine diagram is automatically represented by a diagram frame enclosing the StateMachine diagram elements. You can:

- Hide the frame by right-clicking on the diagram and selecting the 'Hide Diagram Frame' option (and show it again by selecting the 'Show Diagram Frame' option)
- Make the frame selectable to move or resize it, by right-clicking on it and selecting the 'Selectable' option
- Create Ports and Parts on the frame and create connectors between them

If set to non-selectable, the frame will auto-resize to fit the bounds of the diagram, expanding from its default size but not shrinking smaller.

Note that diagrams showing Diagram Frames applied under release 14.0 or higher of Enterprise Architect will draw the parent object on the diagram when opened in a release prior to release 14.0.



### Access

On the Diagram Toolbox, click on  to display the 'Find Toolbox Item' dialog and specify 'SysML n.n StateMachine' (whichever version you are using).

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5



## SysML StateMachine Objects

Item	Description
State	Declares a significant condition in the life of a SysML Block within its StateMachine.
StateMachine	Describes the life-cycle behavior of a SysML Block in terms of its states and transitions.
Initial	Declares the starting state of the StateMachine.
Final	Declares the ending state of the StateMachine, and its completion.
Choice	Declares a Junction with a mandatory 'else' transition.
History	Represents the last active State of the StateMachine prior to its interruption.
Junction	Declares a decision point at which a Transition branches out into multiple guarded, alternative paths.
Entry	Declares an Entry point between StateMachines, SubstateMachines and Regions.
Exit	Declares an Exit point between StateMachines, SubstateMachines and Regions.
Terminate	Declares a termination State in which the StateMachine no longer operates.
Fork/Join	Simultaneously branches and joins a set of Transitions.

## SysML State Relationships

Item	Description
Transition	Establishes a life-cycle path between one State and another, based on its operational conditions.

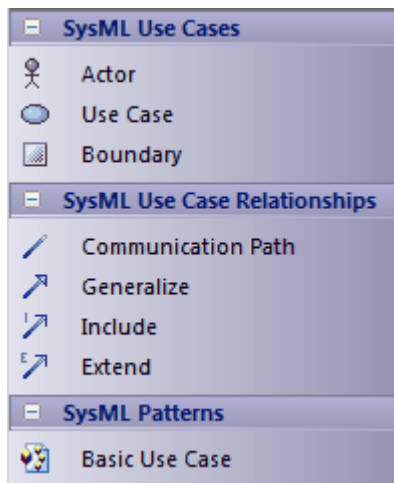
## Additional

Item	Description
Event	Depicts the action of sending a signal.
Signal	A specification of Send request instances communicated between objects.

Trigger	Indicates an event that initiates an action (and might arise from completion of a previous action).
---------	---

# SysML Use Case Toolbox

When you are constructing SysML models, you can populate the Use Case diagrams using the icons on the 'SysML Use Cases' pages of the Diagram Toolbox.



## Access

On the Diagram Toolbox, click on  to display the 'Find Toolbox Item' dialog and specify 'SysML n.n Use Cases' (whichever version you are using).

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5

## SysML Use Case Objects

Item	Description
Actor	Represents a user that interacts with one or more SysML systems.
Use Case	Describes the expected functionality of a system as a UML Use Case.
Boundary	Graphically binds elements in a diagram within a border.

## SysML Use Case Relationships

Item	Description

Communication Path	Declares which Actors perform in the Use Case.
Generalize	Describes an element as a specialized descendant of another element, containing additional properties and behavior.
Include	Describes one Use Case as a subset of another.
Extend	Describes one Use Case as an extension of another.

## SysML Patterns

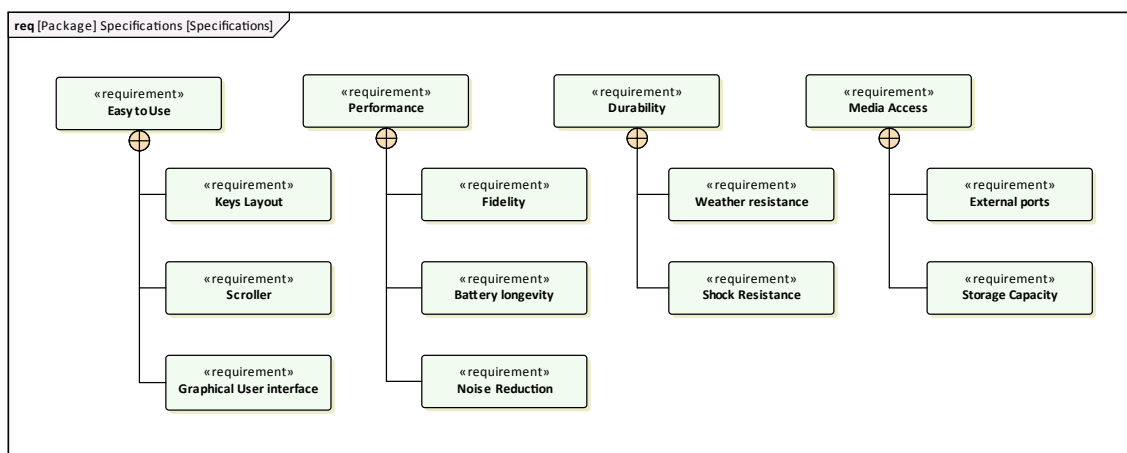
Item	Description
Basic Use Case	A Pattern that creates a typical simple Use Case diagram of Actor, Use Case and System Boundary elements.

## A SysML Requirements Model

Requirements engineering is a fundamental aspect of a systems engineering model. The discipline focuses on eliciting, analyzing and managing customer requirements early in the process. Once the requirements are understood, trade studies can be conducted to formally assess design options, typically using weighted choices. The requirements are managed as first class citizens and are formally allocated to development items and verification methods.

Enterprise Architect has extensive functionality to assist the modeler with every aspect of the requirements engineering discipline, including elicitation, modeling, management and testing. High quality engineering documentation can be generated out-of-the-box using a wide range of built-in templates, carefully crafted to extract the information in the models and present it in visually compelling and high quality documentation in a wide range of formats, including docx, pdf and html. The documentation engine is highly configurable, and documentation to match any engineering or organizational standard can be generated by creating templates and setting generation options.

The SysML Requirements Model provides the system requirements, the expected abstract behavior and the operating constraints that the designed system must conform to. This diagram shows an example requirements model for a Portable Audio Player.



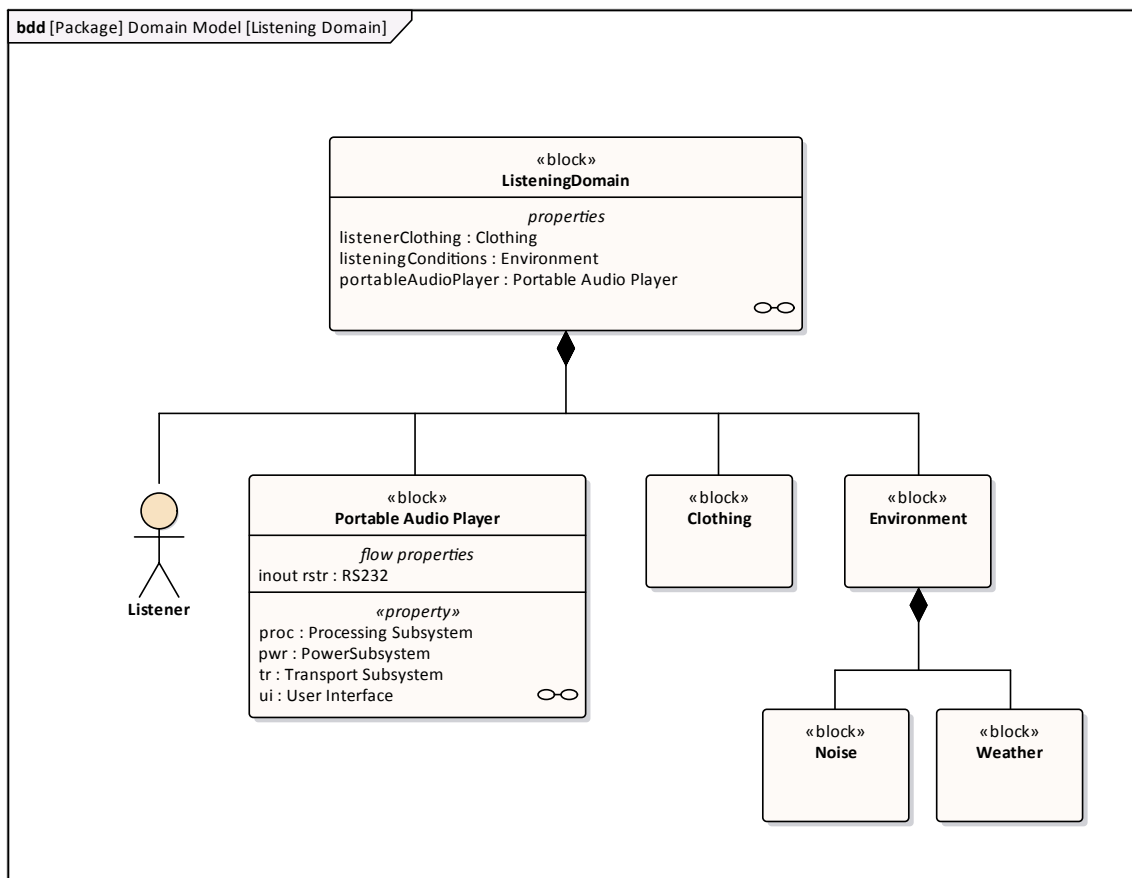
This example displays several top level requirements such as 'Ease of Use' and then breaks those requirements down into more refined requirements such as the 'Graphical User Interface'.

## A SysML Operational Domain Model

Systems engineering is an interdisciplinary field of engineering that takes a whole-of-system view of a problem and its solution. The operational domain model is a central part of any model-based approach and describes the system in the context of its environment. This includes the humans that are intended to operate and interact with the system, external objects that might influence the system, and environmental elements that could impact the system. The operational domain model is a useful starting point to get an overview of a system and how it will operate.

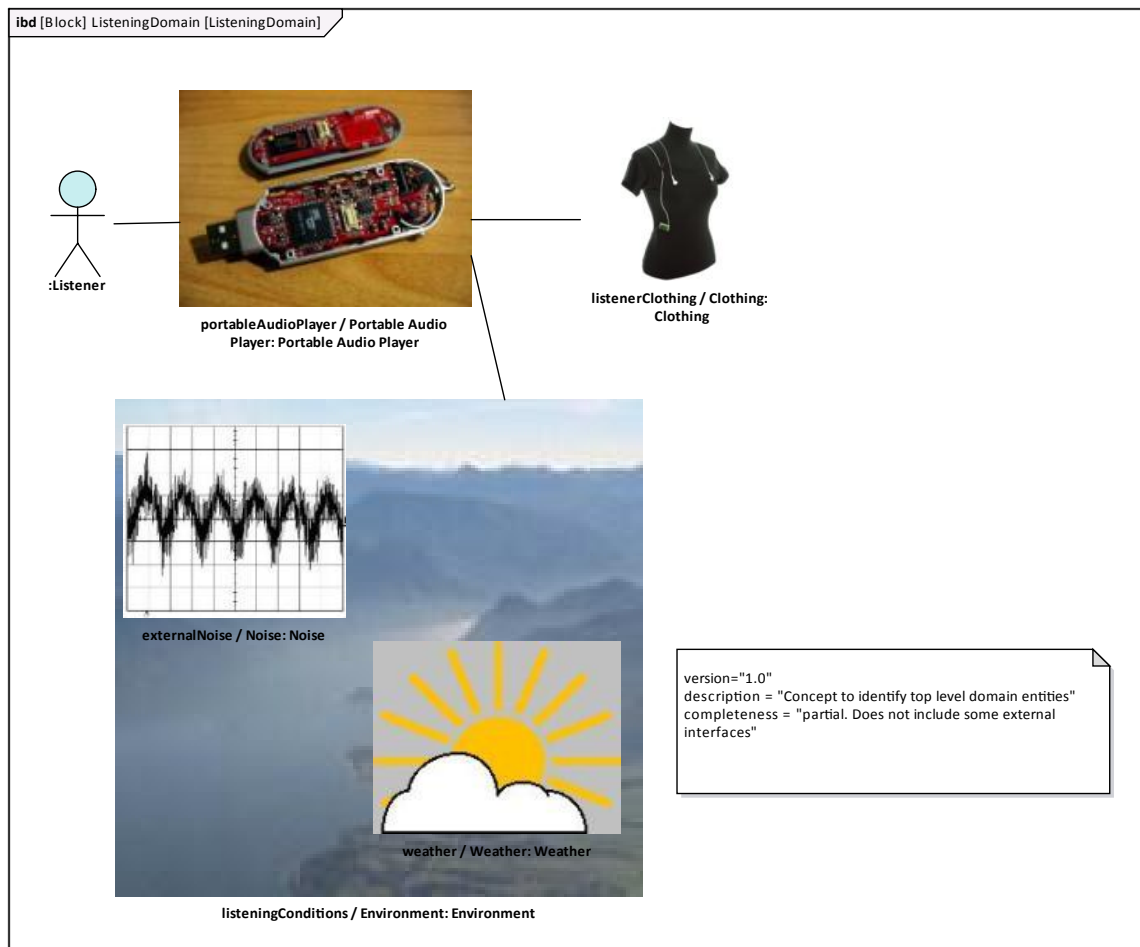
Enterprise Architect provides a range of features that help the engineer to construct an operational domain model, including standard SysML Block definition and Internal Block diagrams, and the ability to include pictorial representations of elements that make the diagrams more compelling. The elements can also be hyper-linked, enabling the viewer to use a diagram as a launching pad to more detailed models and diagrams.

The SysML Operational Domain Model defines the system's operating environment, which describes the operating conditions that the system is intended to operate within. This diagram shows an example Operational Domain model for a Portable Audio Player; the SysML Block Definition diagram describes the Operational Domain (in this example - the ListeningDomain) as a system composition.



In the example, the ListeningDomain is defined as a system containing other subsystems; the domain contains subsystems that define the Listener (that is, User), the Portable Audio Player, Clothing (which the user wears), and the External Environment.

Details of the ListeningDomain system are further detailed in the ListeningDomain's Internal Block diagram:



In the example, the ListeningDomain system's detailed composition shows how the Portable Audio Player and other sub-systems fit together to form the Listening Domain; it also describes the binding relationships between the parts, which define how the parts are functionally bound to one another.

# Compose System Design

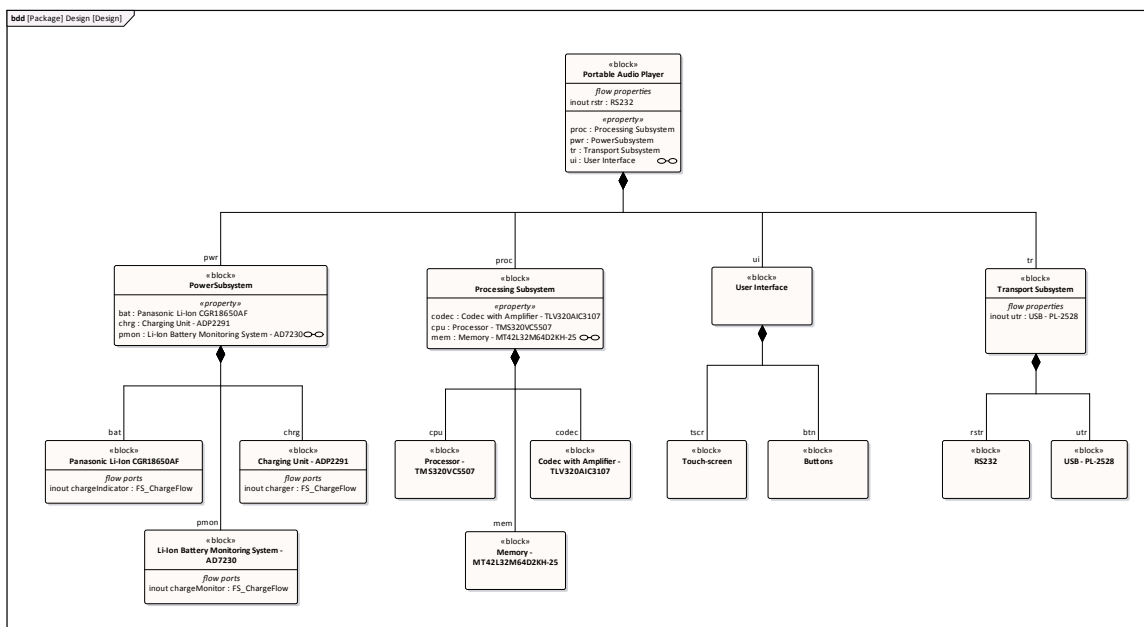
The systems engineering language has a strong focus on design. Once the requirements have been elicited, modeled and analyzed, the attention of the engineer turns to design. The systems are typically complex and must be broken down into a number of Subsystems that will interact with each other through known and published interfaces. The decomposition of the system is modeled using a Block Definition diagram displaying the subsystems, which in turn can be further decomposed. These subsystems can be traced back to the requirements and then to the stakeholders who own the requirements.

A Block Definition diagram can be created in Enterprise Architect to model the decomposition of a system into a hierarchy of subsystems using the composition association. Subsystems can be hyper-linked to more detailed diagrams, which allow the viewer to click through from the system level through to all its constituent parts. The subsystems can also be linked back to the requirements they are implementing.

## The SysML Design Model

The SysML Design Model contains the blocks that define the system's composition; it describes the manner in which reusable subsystems fit together to fulfill the design requirements.

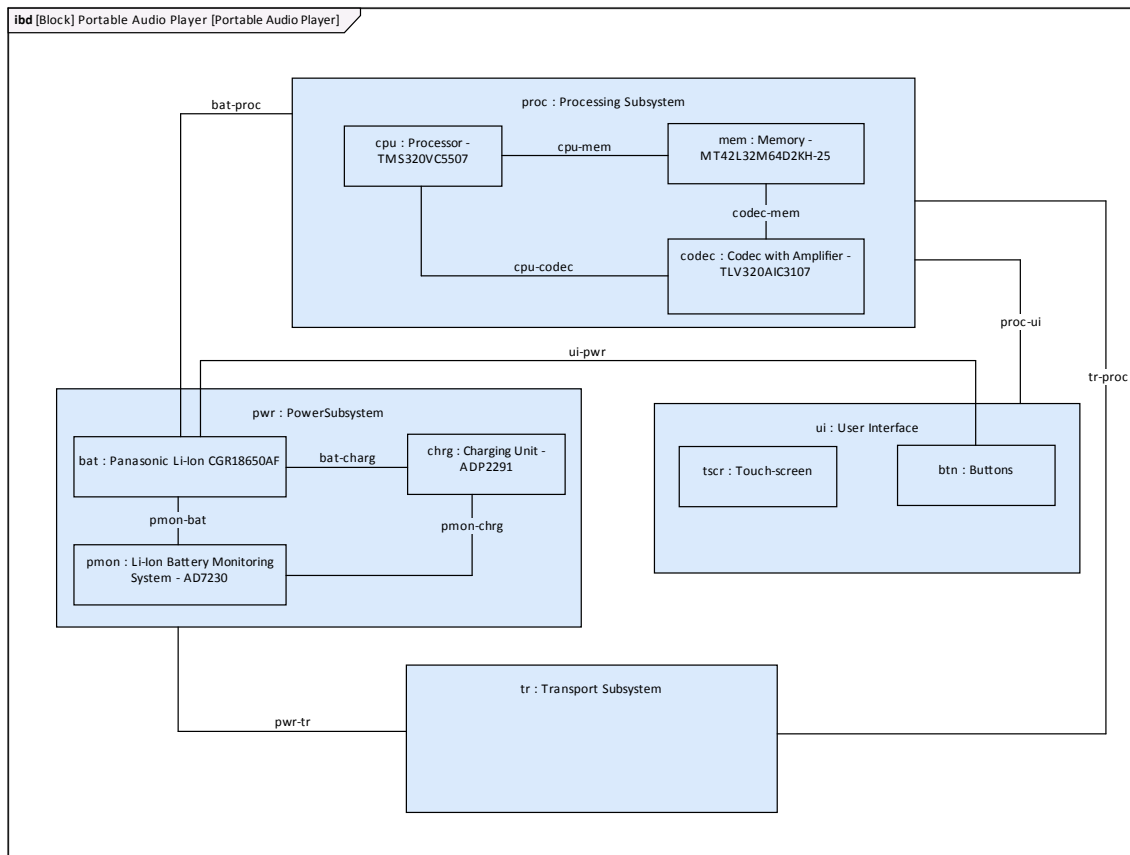
This diagram shows an example Design Model for a Portable Audio Player; the SysML Block Definition diagram describes the Portable Audio Player as a composition of various reusable off-the-shelf subsystems and in-house designed ones.



In this example, the Portable Audio Player is defined as a SysML system containing subsystems that perform specific tasks; the design contains subsystems for supplying power, performing playback and audio processing, and interfacing with other devices and the user interface.

Details of the Portable Audio Player's composition are further described in detail within the Portable Audio Player's Internal Block diagram:





The example describes the Portable Audio Player's composition, detailing how each of the sub-systems is structured. The example also describes the binding relationships between the parts, which define how they are functionally bound to one another; for example, the CPU, Memory and Codec are interfaced together in the Processing Subsystem.

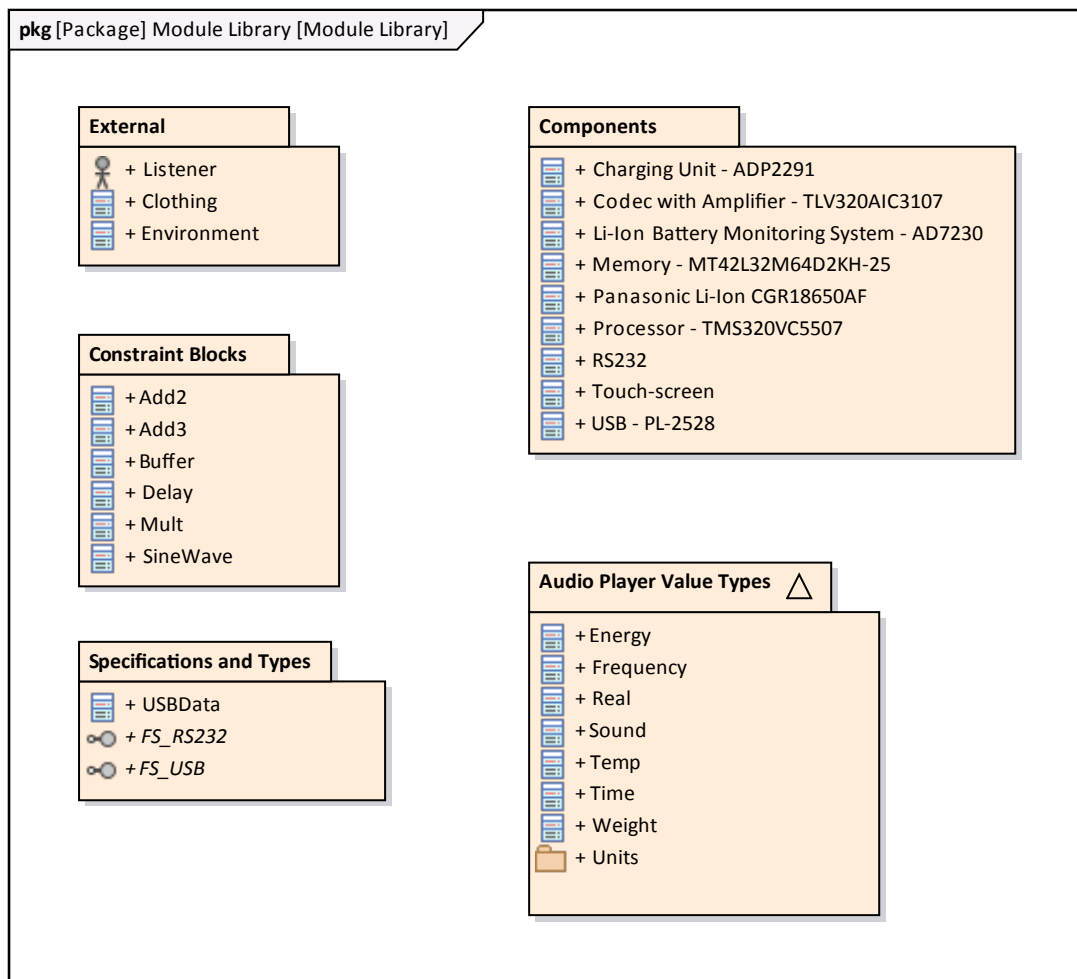
# Create Reusable Subsystems

Model-based Systems Engineering provides the flexibility and expressiveness to define complex systems quickly and effectively, by reusing common entities across design projects. Before the model based approach became prevalent, systems were defined using document based methods with little opportunity for re-use. The SysML contains a series of reusable libraries such as the SI Definitions and the SI Value Types, but also supports a modeler in creating additional domain or technology-specific libraries that could be reused within an organization, or published for wider use by a community of users or an entire industry.

Enterprise Architect provides a range of functionality to assist in creating, discovering, visualizing and reusing libraries of elements such as subsystems, parametric constraints, common data types, common value types, dimensions and units. The Reusable Asset Service could be used to store these assets in order to provide a canonical set of libraries governed by the respective standards agencies.

## SysML Design Model

A Library is a Package containing many reusable subsystems, parametric constraints, common data types and common value types, dimensions and units. This diagram shows an example library model:



In the example Library, each of the child Packages contains child models that capture these reusable entities:

- Blocks defining systems such as those listed in the Components Package, or those defined in the External Package
- ConstraintBlocks defining parametric constraints for use in parametric models
- Value Types describing quantities, expressed as measurable dimensions in specific units

- Data Types and Flow Specifications describing data structures and Flows

# Migrate SysML Model to Later SysML Version

Enterprise Architect provides a useful feature to migrate a model from one version of SysML to the next. A model (or part of a model) created in an older version of the SysML MDG Technology can be migrated to the next version using the automation interface. This function updates the Tagged Values and, if required, stereotypes to the later version for all elements, attributes, connectors and diagrams under the selected Package or element.

There is no facility to migrate a model from SysML 1.4 to SysML 1.5, because the two releases are functionally the same.

## Example Script for Migrating SysML 1.1 to SysML 1.2

This VB script calls the Migrate() function to migrate the SysML 1.1 Package or element to SysML 1.2:

```
Sub MigrateElement (sGUID, lngPackageID)
    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.Migrate sGUID, "SysML1.1", "SysML1.2"

    'refresh the model
    If lngPackageID <> 0 Then
        Repository.RefreshModelView (lngPackageID)
    End If
End Sub

Sub MigrateSelectedItem
    Dim selType
    Dim selElement as EA.Element
    Dim selPackage as EA.Package

    selType = GetTreeSelectedItemType
    If selType = 4 Then 'means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox "Element Migration Completed",0,"SysML Migration"
    ElseIf selType = 5 Then 'means Package
        set selPackage = GetTreeSelectedObject
        MigrateElement selPackage.PackageGUID, selPackage.PackageID
        MsgBox "Package Migration Completed",0,"SysML Migration"
    Else
        MsgBox "Select a Package or Element in the Project Browser to initiate migration",0,"SysML Migration"
    End If
End Sub
```

Sub Main

    MigrateSelectedItem

End Sub

Main

## Migrate SysML 1.1 to SysML 1.3

Migrate from SysML 1.1 to SysML 1.2, as explained earlier, then migrate from SysML 1.2 to SysML 1.3, as described in the next section.

## Migrate SysML 1.2 to SysML 1.3

The MDG Technology for SysML 1.3 has a built-in script for migrating 1.2 models to 1.3. You must enable both the MDG Technology for SysML 1.2 and the MDG Technology for SysML 1.3. Firstly, you select the SysML 1.2 Package in the Project Browser, then you open the Scripting window and open the SysML 1.3 script group, and finally you execute the Migrate script.

## Migrate SysML 1.3 to SysML 1.4

The MDG Technology for SysML 1.4 has a built-in script for migrating 1.3 models to 1.4. You must enable both the MDG Technology for SysML 1.3 and the MDG Technology for SysML 1.4. Firstly, you select the SysML 1.3 Package in the Project Browser, then you open the Scripting window and open the SysML 1.4 script group, and finally you execute the Migrate script.

## Notes

- When migrating from SysML1.1 to SysML1.2, the:
  - Stereotype dimension is changed to quantitykind
  - Stereotype dataType is removed from SysML1.2
  - Tagged Value dimension is migrated to quantitykind; this applies to stereotypes <<unit>> and <<valueType>>
  - Tagged Value isConjugated of stereotype <<flowport>> is migrated to custom properties

## Simple Parametric Simulation (Legacy)

This functionality has been superseded by the [Parametric Simulation using OpenModelica](#) from Enterprise Architect Release 13.0 onwards.


This topic is retained for the use of those who are maintaining existing models using the pre-13.0 functionality.

You use the scripting function of Enterprise Architect to simulate a SysML model from a Parametric diagram, using the 'Simulation Configuration' dialog. This is limited when compared to the OpenModelica-based simulation, because the behavior of each Constraint Block is represented by a script that calculates outputs from a known set of inputs instead of solving for the unknowns dynamically.

### Access

Context Menu	On Parametric diagram   Right-click   SysML   Simulate Diagram...
--------------	---

### Simulate a SysML model

Step	Action
1	<p>The 'Parameters' panel lists all of the parameters that can be assigned input.</p> <p>Select each of the required parameters and click on the right Arrow button to assign them as input.</p> <p>Parameters designated as input parameters are listed in the 'Inputs' panel on the right.</p> <p>There must be at least one input parameter assigned for the simulation to execute.</p>
2	<p>Assign a set of values for each of the designated input parameters.</p> <p>For each input parameter, in the 'Input Values' panel select one of the two possible value kinds:</p> <ul style="list-style-type: none"> <li>Discrete - To enter a constant or a comma-separated range of discrete values</li> <li>Range - To enter a range of values beginning at the 'From' value and ending at the 'To' value; the input values are incremented by the 'Step' value</li> </ul>
3	<p>Specify the classes of output value:</p> <ul style="list-style-type: none"> <li>'Parameters' - To output the parameters' data, select the checkbox</li> <li>'Variables' - To output the data generated within each internal variable, select the checkbox; internal variables are automatically generated by the simulator</li> </ul>
4	<p>Specify how the simulation results are to be reported.</p> <p>The 'Output Format' panel enables you to choose how the simulation outputs the simulation data:</p> <ul style="list-style-type: none"> <li>Plot To Graph: To plot the results on a 2-dimensional graph, select the checkbox; if you select this option, you must specify an input parameter for the plot's X Axis</li> <li>Title - To enter a title for the graph, type in the title text</li> <li>Output to File - To output the results to a CSV text file, select the checkbox and type or browse (click on ) for the file name</li> </ul>
5	Click on the OK button to execute the simulation.

## Notes

- Systems Modeling Language (SysML) Parametric Model Simulation is available in the Unified and Ultimate editions of Enterprise Architect

