Rose Domain

Introduction

The Rose domain lets you incorporate objects from Rational Rose models in your reporting.

Among the objects that you can extract are packages, classes, uses cases, and diagrams.

In addition, you can extract external documents that are associated with a specific object. That procedure is provided in the Help.

Aliases

The following names in the Rose domain are aliased in the reporting interface:

Name in Domain	Name in Interface	Item Type / Location
Packages RootUseCasePackage Subsystems AllModules RootLogicalPackage AllScenarios RootSubsystem	AllPackages UseCaseView AllSubsystems AllComponents LogicalView AllInteractionDiagrams ComponentView	Relationship / Model Class
HasStateDiagram	HasStateActivityDiagram	Property / Package Class
StateDiagram StateDiagrams StateMachine StateMachines UseCases Scenarios Associations	FirstStateActivityDiagram StateActivityDiagrams FirstStateActivityModel StateActivityModels MyUseCases InteractionDiagrams MyAssociations	Relationship / Package Class
Abstract HasStateDiagram	IsAbstract HasStateActivityDiagram	Property / Class Class
StateDiagram StateMachine StateDiagrams Relationships SubClasses SuperClasses Modules Operations Attributes	FirstStateActivityDiagram FirstStateActivityModel StateActivityDiagrams MyRelationships MySubClasses MySuperClasses Components MyOperations MyAttributes	Relationship / Class Class
ReturnType CppImage HasStateDiagram	ReturnClass C++Image HasStateActivityDiagram	Property / Operation Class
StateDiagram StateMachine StateDiagrams StateMachines Parameters	FirstStateActivityDiagram StateActivityModel StateActivityDiagrams StateActivityModels Arguments	Relationship / Operation Class
InitValue Derived Static	InitialValue IsDerived IsStatic	Property / Attribute Class

Name in Domain	Name in Interface	Item Type / Location
HasLinkClass Derived	HasLinkElement IsDerived	Property / Association Class
Role1 LinkClass Role2	RoleA LinkElement RoleB	Relationship / Association Class
StateMachine	StateActivityModel	Class
HasDiagram	HasStateActivityDiagram	Property / StateMachine Class
Diagram Diagrams AllDiagrams StateMachines	FirstStateActivityDiagram StateActivityDiagrams AllStateActivityDiagrams StateActivityModels	Relationship / StateMachine Class
Scenario	InteractionDiagram	Class
Navigable Static Aggregate	IsNavigable IsStatic IsAggregate	Property / Role Class
Module	Component	Class
ModuleDiagrams AllModules MainDiagram Modules	ComponentDiagrams AllComponents Diagram Components	Relationship / Subsystem Class
ObjectInstance	Object	Class
MyClassName	ClassName	Property / ObjectInstance Class
SupplierRole ClientRole	Supplier Client	Relationship / Link Class
Abstract HasStateDiagram	IsAbstract HasStateActivityDiagram	Property / UseCase Class
StateDiagram StateMachine StateDiagrams StateMachines Scenarios	FirstStateActivityDiagram FirstStateActivityModel StateActivityDiagrams StateActivityModels InteractionDiagrams	Relationship / UseCase Class
StateMachines StateDiagrams ParentStateMachine	StateActivityModels StateActivityDiagrams ParentStateActivityModel	Relationship / Activity Class
SyncItem	Synchronization	Class
StateDiagram	StateActivityDiagram	Class
HasStateMachine	HasStateActivityModel	Property / StateDiagram Class
StateMachine	StateActivityModel	Relationship / StateDiagram Class
ModuleDiagram	ComponentDiagram	Class

Name in Domain	**Name in Interface**	**Item Type / Location**
StateMachines StateDiagrams ParentStateMachine	StateActivityModels StateActivityDiagrams ParentStateActivityModel	Relationship / State Class
ExternalDocument	String	Class
Type	Kind	Property / Note Class
InheritRelationship	InheritsRelationship	Class
Virtual	IsVirtual	Property / InheritRelationship Class
MappedPoints	ItemsPositionInfo	Property / Diagram Class
MappedArtifacts	Items	Relationship / Diagram Class

Action
Rose Domain

An action is an operation that is associated with a state transition.

Class Hierarchy: Item>Action

SubClasses of Action
This class has no subclasses.

Properties Specific to Action

Properties	Inherited From	Description
Arguments		The arguments that accompany the trigger event.
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
OnEvent_Arguments		For a Rose action that is an event, the name of the event.
OnEvent_Condition		For a Rose action that is an event, the associated guard condition.
OnEvent_Event		For a Rose action that is an event, the arguments for that event.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
Target		The name of the event object.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Action
This class has no relationships.

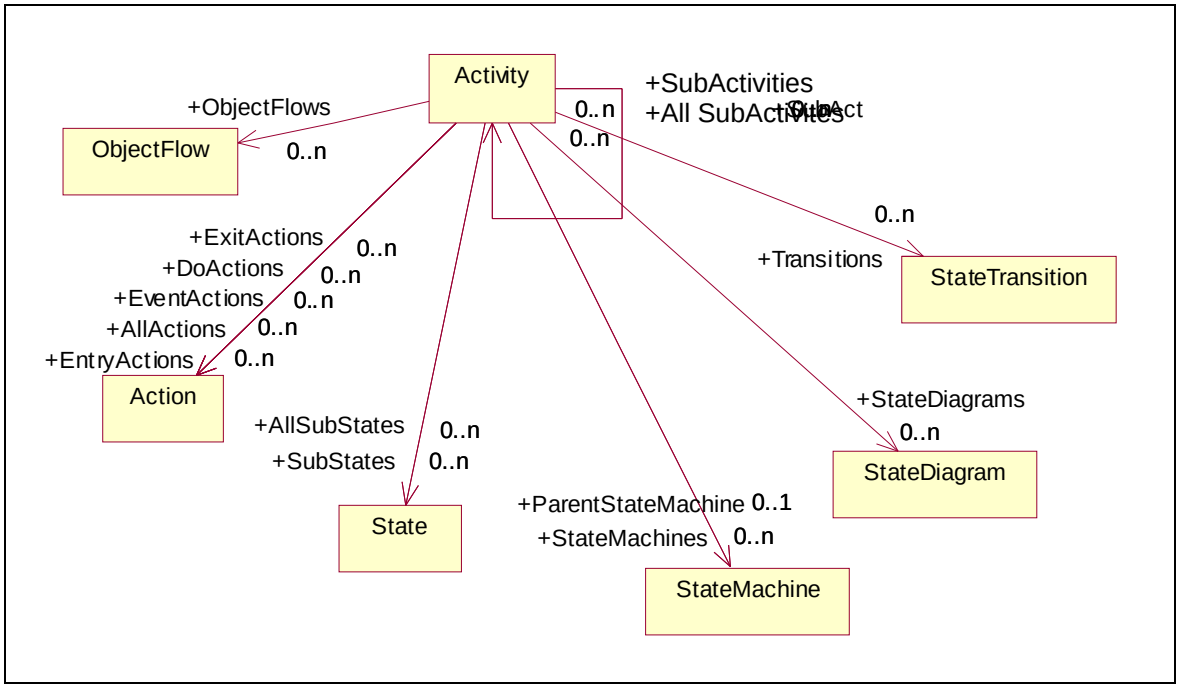
Activity
Rose Domain

The Activity class is an abstract class that exposes activity functionality in the Rose extensibility interface. With the Rose Activity class, you can:
Retrieve information about activities, such as name, documentation, stereotype.
Retrieve objects associated with activities such as parent activities, parent states, parent state machines, child activities, child decisions, child states, child synchronizations, outgoing transitions, and swimlanes.
Create and retrieve tool and property settings for activities.
Open specification sheets for activities.
Add, delete, and retrieve an activity's actions, state machines, and events.
Add and delete transitions.

Class Hierarchy: Item>Activity

SubClasses of Activity

This class has no subclasses.



Properties Specific to Activity

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Activity

Name	Kind	Class	Description
AllActions	0..n	Action	All actions associated with this Activity -- EntryActions, ExitActions, DoActions, and EventActions.
AllSubActivities		Activity	All activities associated with this Activity.
AllSubStates	0..n	State	States associated with this Activity.
DoActions	0..n	Action	The Do actions for this Activity.
EntryActions	0..n	Action	The Entry actions for this Activity.
EventActions	0..n	Action	The Event actions for this Activity.
ExitActions	0..n	Action	The Exit actions for this Activity.
ObjectFlows	0..n	ObjectFlow	The associated object flows for this Activity.
ParentStateMachine	0..1	StateMachine	Parent state machine associated with this Activity.
StateDiagrams	0..n	StateDiagram	State or activity diagrams internal to this Activity.
StateMachines	0..n	StateMachine	State machines internal to this Activity.
SubActivities		Activity	Activities that are part of this Activity.
SubStates	0..n	State	States that are part of this Activity.
Transitions	0..n	StateTransition	Transitions that exit from this Activity.

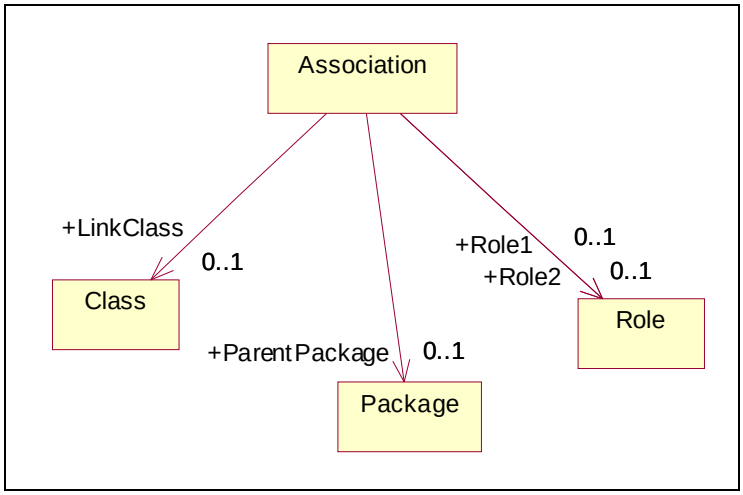
Association
Rose Domain

An association provides a pathway for communication. The communication can be between use cases, actors, classes, or interfaces. Associations are the most general of all relationships and consequentially the most semantically weak. If two objects are usually considered independently, the relationship is an association.

Class Hierarchy: Item>Association

SubClasses of Association

This class has no subclasses.



Properties Specific to Association

Properties	Inherited From	Description
Constraints		The text from the Constraints field in the association specification.
Derived		True if the association is derived; otherwise False.
Documentation	Item	Documentation for the item.
HasLinkClass		True if the association has an attached Association class, otherwise False.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Association

Name	Kind	Class	Description
LinkClass	0..1	Class	The linked class attached to the Association.
ParentPackage	0..1	Package	Parent package attached to the Association.
Role1	0..1	Role	The first role defined in the Association.
Role2	0..1	Role	The second role defined in the Association.

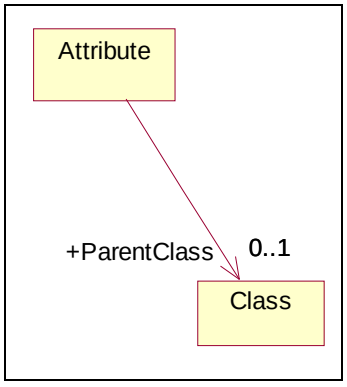
Attribute
Rose Domain

Attributes are data members of a class whose type is not another class. Attributes define the characteristics of a class. Each object in a class has the same attributes, but the values of the attributes may be different.

Class Hierarchy: Item>Attribute

SubClasses of Attribute

This class has no subclasses.



Properties Specific to Attribute

Properties	Inherited From	Description
Containment		Specifies the physical containment of the attribute. Returns Value, Reference, or Unspecified, depending on the state of the Containment radio control on the attribute specification.
Derived		True if the Derived check box is selected in the attribute specification, otherwise False.
Documentation	Item	Documentation for the item.
ExportControl		The export control of the attribute. Returns Public, Protected, Private, or implementation.
InitValue		The initial value of the Attribute.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Static		True if the Static check box is selected in the attribute specification, otherwise False.
Stereotype	Item	Stereotype of the item.

Properties	Inherited From	Description
Type		The type of the Attribute.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Attribute

Name	Kind	Class	Description
ParentClass	0..1	Class	The class in which this attribute is defined.

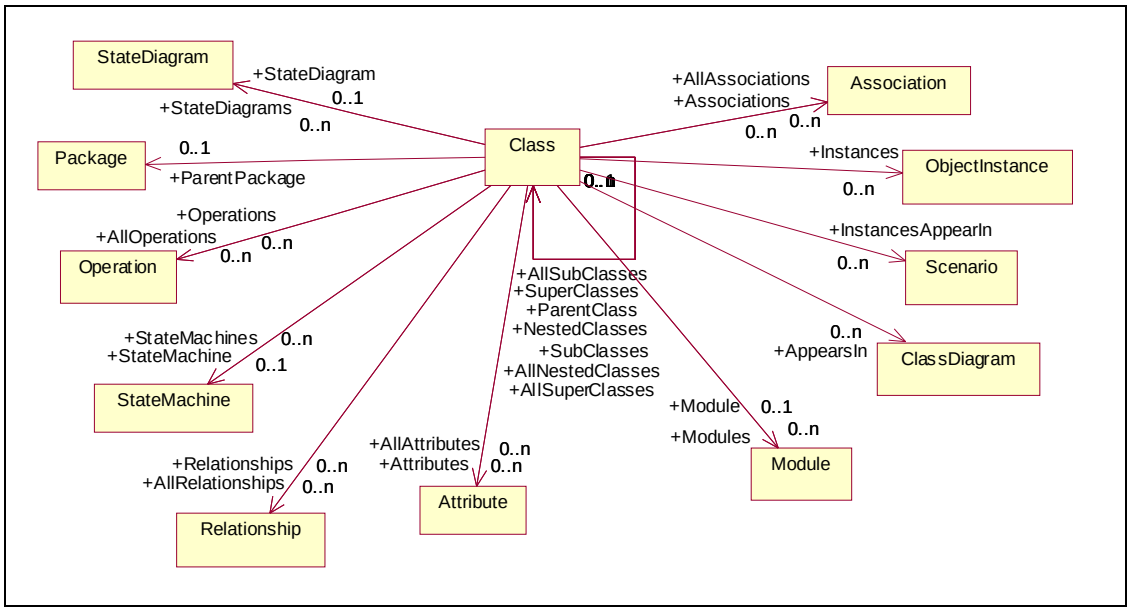
Class
Rose Domain

A class captures the common structure and common behavior of a set of objects. A class is an abstraction of real-world items. When these items exist in the real world, they are instances of the class, and referred to as objects. Rational Rose stores class information in a class specification.

Class Hierarchy: Item>Class

SubClasses of Class

ClassUtility
InstantiatedClass
InstantiatedClassUtility
MetaClass
ParameterizedClass
ParameterizedClassUtility



Properties Specific to Class

Properties	Inherited From	Description
Abstract		True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality		The string in the Cardinality field of the class specification.
Concurrency		Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
ExportControl		Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType		Returns TRUE if this class is a fundamental type.
HasStateDiagram		Returns TRUE if the class has an associated state diagram, otherwise FALSE.
IsNested		Returns TRUE if the Class is nested, otherwise FALSE.
Kind		The kind of Class.
Name	Item	Name of the item.
Persistence		This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space		The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Class

Name	Kind	Class	Description
AllAssociations	0..n	Association	All associations where this Class plays a role, including those inherited from other classes.
AllAttributes	0..n	Attribute	All attributes of this Class, including those inherited from other classes.
AllNestedClasses		Class	All nested classes of this Class.
AllOperations	0..n	Operation	All operations of this Class, including those inherited from other classes.
AllRelationships	0..n	Relationship	All relationships of this Class, including those inherited from other classes.

Name	Kind	Class	Description
AllSubClasses		Class	All classes in the lineage of this Class. For example, if A inherits from B and B inherits from C, then AllSubClasses of C would include B and A.
AllSuperClasses		Class	All classes in the ancestry of this Class. For example, if A inherits from B and B inherits from C, then AllSuperClasses of A would include B and C.
AppearsIn	0..n	ClassDiagram	The class diagrams where this Class appears.
Associations	0..n	Association	The associations where this Class plays a role.
Attributes	0..n	Attribute	Attributes that are defined by this Class. Does not include inherited attributes.
Instances	0..n	ObjectInstance	Object instances associated with this Class.
InstancesAppearIn	0..n	Scenario	Interaction diagrams that include instances of this Class.
Module	0..1	Module	The first module associated with this Class.
Modules	0..n	Module	All modules associated with this Class.
NestedClasses		Class	Classes that are nested within this Class.
Operations	0..n	Operation	Operations that are defined by this Class. Does not include inherited operations.
ParentClass		Class	Parent class of this Class, if it is nested.
ParentPackage	0..1	Package	The enclosing package.
Relationships	0..n	Relationship	Relationships that are defined by this Class. Does not include inherited relationships.
StateDiagram	0..1	StateDiagram	The (first) state/activity diagram associated with this Class.
StateDiagrams	0..n	StateDiagram	All state/activity diagrams associated with this Class.
StateMachine	0..1	StateMachine	The (first) state machine associated with this Class.
StateMachines	0..n	StateMachine	All state machine associated with this Class.

Name	Kind	Class	Description
SubClasses		Class	The classes that directly inherit from this Class. Only includes immediate subclasses. For example, if A inherits from B and B inherits from C, then MySubClasses of C would include B but not A.
SuperClasses		Class	The classes that this Class directly inherits from. Only includes immediate superclasses. For example, if A inherits from B and B inherits from C, then MySuperClasses of A would include B but not C.

ClassDiagram
Rose Domain

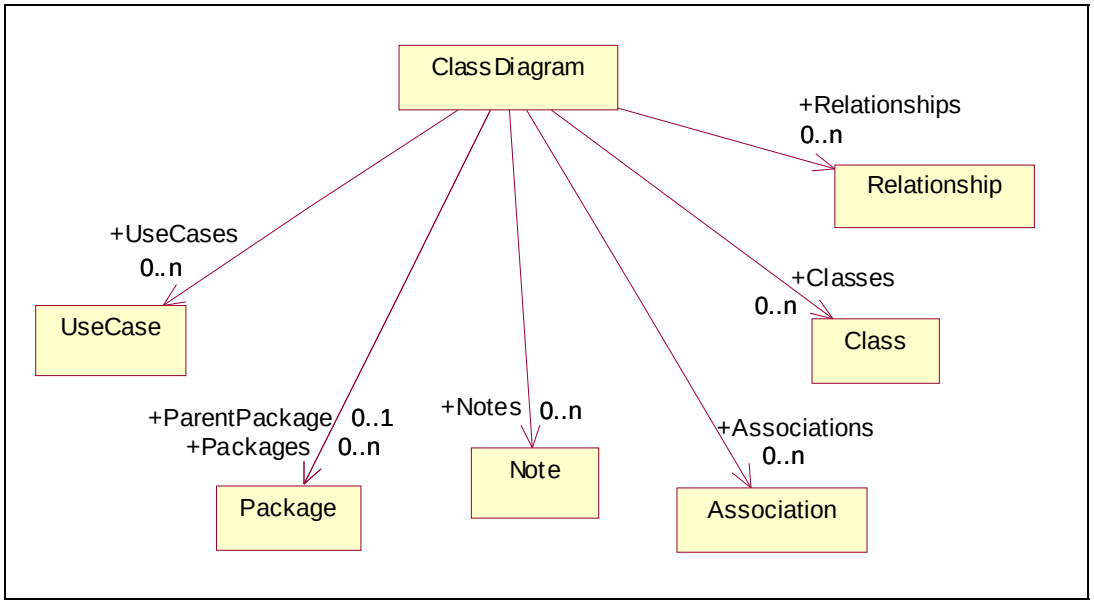
A class diagram shows the relationships between packages and classes; the essential relationships include association, inherits, has, and uses. Each class diagram provides a logical view of the current model.

Class diagrams contain icons representing packages and classes. Class diagrams can be considered as filtered views into the model. They do not necessarily depict all the classes or relationships in the model. For example, iterating over all the classes in the main diagram of a package will not necessarily return all the classes defined in that category.

Class Hierarchy: Diagram>ClassDiagram

SubClasses of ClassDiagram

This class has no subclasses.



Properties Specific to ClassDiagram

Properties	Inherited From	Description
Documentation	Diagram	The documentation text associated with the Diagram.
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.
Name	Diagram	Name of the Diagram.

Properties	Inherited From	Description
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to ClassDiagram

Name	Kind	Class	Description
Associations	0..n	Association	The associations where this class diagram plays a role.
Classes	0..n	Class	All of the classes that appear on the diagram.
Notes	0..n	Note	Notes that appear in the diagram.
Packages	0..n	Package	All packages associated with this class diagram.
ParentPackage	0..1	Package	The package that this diagram is contained in, if applicable.
Relationships	0..n	Relationship	All of the relationships that appear on the diagram.
UseCases	0..n	UseCase	All of the use cases that appear on the diagram.

ClassUtility
Rose Domain

A class utility is a set of operations that provide additional functions for classes. Class utilities are used to:
· Denote one or more free subprograms.
· Name a class that only provides static members and/or static member functions.

Class Hierarchy: Item>Class>ClassUtility

SubClasses of ClassUtility
This class has no subclasses.

Properties Specific to ClassUtility

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.
Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns TRUE if this class is a fundamental type.
HasStateDiagram	Class	Returns TRUE if the class has an associated state diagram, otherwise FALSE.
IsNested	Class	Returns TRUE if the Class is nested, otherwise FALSE.
Kind	Class	The kind of Class.
Name	Item	Name of the item.
Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.

Properties	Inherited From	Description
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ClassUtility

This class has no relationships.

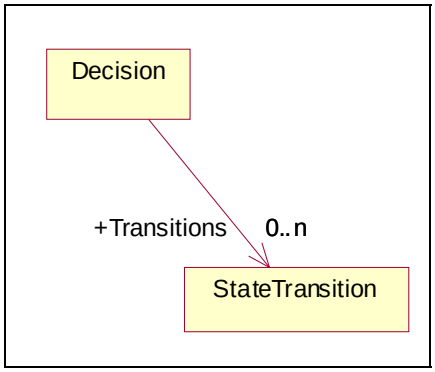
Decision
Rose Domain

The Decision class is an abstract class that exposes decision functionality in the Rose extensibility interface.

Class Hierarchy: Item>Decision

SubClasses of Decision

This class has no subclasses.



Properties Specific to Decision

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Decision

Name	Kind	Class	Description
Transitions	0..n	StateTransition	State transition for this Decision.

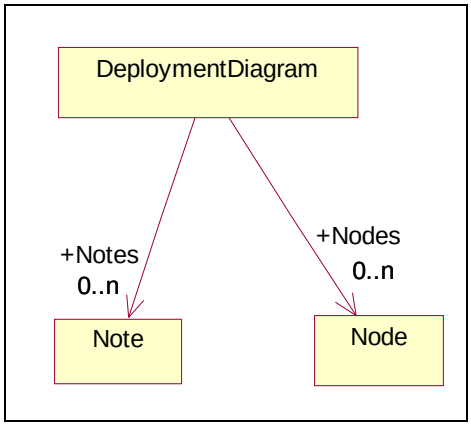
DeploymentDiagram
Rose Domain

A deployment diagram shows the allocation of processes to processors in the physical design of a system. A deployment diagram may represent all or part of the process architecture of a system.

Class Hierarchy: Diagram>DeploymentDiagram

SubClasses of DeploymentDiagram

This class has no subclasses.



Properties Specific to DeploymentDiagram

Properties	Inherited From	Description
Documentation	Diagram	The documentation text associated with the Diagram.
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.
Name	Diagram	Name of the Diagram.
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to DeploymentDiagram

Name	Kind	Class	Description
Nodes	0..n	Node	Processors and devices contained in the diagram.
Notes	0..n	Note	Notes that appear in the diagram.

Device
Rose Domain

A device is a hardware component with no computing power. The Rose device class exposes properties and methods that allow you to define and manipulate the characteristics of devices.

Class Hierarchy: Item>Node>Device

SubClasses of Device
This class has no subclasses.

Properties Specific to Device

Properties	Inherited From	Description
Characteristics	Node	Characteristics of the processor or device.
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Device
This class has no relationships.

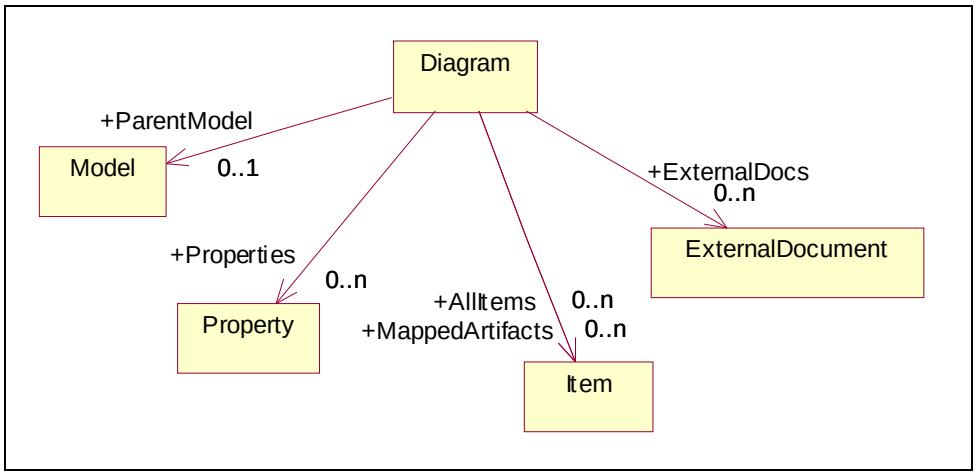
Diagram
Rose Domain

Exposes a set of properties and methods, which all other diagram classes (for example, class diagrams, and scenario diagrams) inherit. These properties and methods determine the size and placement of a diagram on the Rose user's computer screen.

Class Hierarchy: Diagram

SubClasses of Diagram

ClassDiagram
DeploymentDiagram
ModuleDiagram
Scenario
StateDiagram
UseCaseDiagram



Properties Specific to Diagram

Properties	Inherited From	Description
Documentation		The documentation text associated with the Diagram.
MappedPoints		A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.
Name		Name of the Diagram.
QualifiedName		Qualified name of the Diagram.
UniqueID		The unique ID for the Diagram.

Relationships Specific to Diagram

Name	Kind	Class	Description
AllItems	0..n	Item	All model objects placed on this Diagram.
ExternalDocs	0..n	ExternalDocument	External documents attached to this Diagram.
MappedArtifacts	0..n	Item	Items that are associated with this Diagram.
ParentModel	0..1	Model	Model that this Diagram is contained in.
Properties	0..n	Property	Property artifact types associated with this Diagram.

ExternalDocument
Rose Domain

Exposes properties and methods that allow you to create external documents (reports) from within the Rose environment. For example, you can start Word for Windows and output information from a Rose model into a Word document.

Class Hierarchy: ExternalDocument

SubClasses of ExternalDocument
This class has no subclasses.

Properties Specific to ExternalDocument

Properties	Inherited From	Description
CollIndex		The index of the ExternalDocument within the collection of documents that contains it. It is used internally to identify the document.
ParentUID		The unique id of the external document's parent class
Value		The actual path of the document.

Relationships Specific to ExternalDocument
This class has no relationships.

HasRelationship
Rose Domain

The Has Relationship indicates a containment or aggregation relationship between classes. The has relationship, available only with the Booch notation, denotes a whole and part relationship between two classes. This relationship is used to show how instances of the supplier, or aggregate, class are physically constructed from instances of the client class. The FromClass relationship returns the aggregate class. The ToClass relationship returns the client class, whose instances are part of aggregate class instances.

Class Hierarchy: Item>Relationship>HasRelationship

SubClasses of HasRelationship

This class has no subclasses.

Properties Specific to HasRelationship

Properties	Inherited From	Description
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Containment		Specifies the physical containment of the relationship. Returns Value, Reference, or Unspecified, depending on the state of the Containment radio control on the relationship specification. Containment is also shown by adornments on relationships in diagrams.
Documentation	Item	Documentation for the item.
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.

Properties	Inherited From	Description
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Static		Specifies whether the instance of the part class is owned by the class itself and not by its individual instances. Returns True, if the Static check box is checked on the relationship specification. Otherwise, returns False. Static relationships are also designated by special adornments on relationships in diagrams.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to HasRelationship

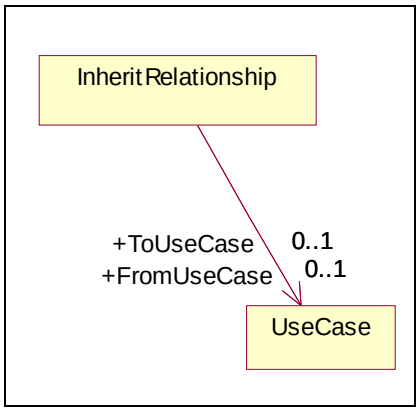
This class has no relationships.

InheritRelationship
Rose Domain

Indicates an inheritance relationship between classes.

Class Hierarchy: Item>Relationship>InheritRelationship

SubClasses of InheritRelationship
This class has no subclasses.



Properties Specific to InheritRelationship

Properties	Inherited From	Description
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Documentation	Item	Documentation for the item.
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
FriendshipRequired		Indicates whether the supplier class grants rights to the client class to access its nonpublic parts. Returns TRUE if the Friendship required check box is checked on the relationship specification. Otherwise, returns FALSE.

Properties	Inherited From	Description
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.
Virtual		Boolean value indicating whether the relation is virtual.

Relationships Specific to InheritRelationship

Name	Kind	Class	Description
FromUseCase	0..1	UseCase	The supplier use case of the inherits relationship, if it is a use case.
ToUseCase	0..1	UseCase	The client use case of the inherits relationship, if it is a use case.

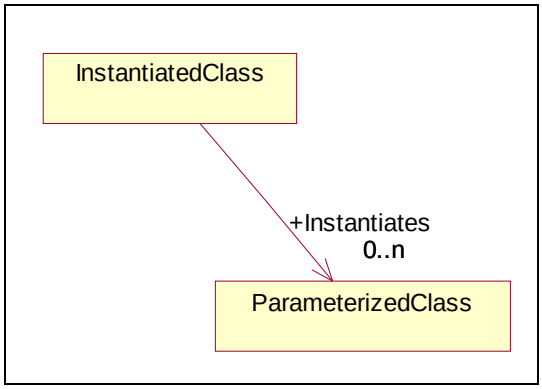
InstantiatedClass
Rose Domain

A class which instantiates a parameterized class. Instantiated classes are created by supplying the actual values for the formal parameters of the parameterized class. An instantiated class is concrete, meaning that its implementation is complete, and it may have object instances.

Class Hierarchy: Item>Class>InstantiatedClass

SubClasses of InstantiatedClass

This class has no subclasses.



Properties Specific to InstantiatedClass

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.
Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns TRUE if this class is a fundamental type.
HasStateDiagram	Class	Returns TRUE if the class has an associated state diagram, otherwise FALSE.

Properties	Inherited From	Description
IsNested	Class	Returns TRUE if the Class is nested, otherwise FALSE.
Kind	Class	The kind of Class.
Name	Item	Name of the item.
Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to InstantiatedClass

Name	Kind	Class	Description
Instantiates	0..n	ParameterizedClass	The parameterized class that this instantiated class instantiates.

InstantiatedClassUtility
Rose Domain

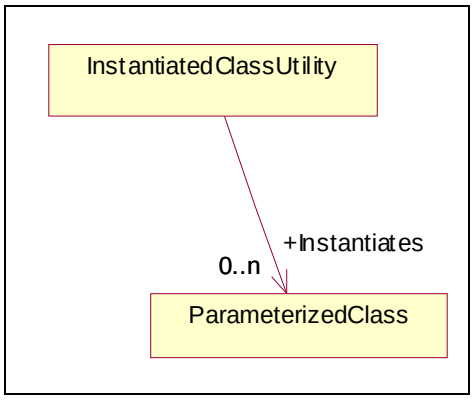
A class utility which instantiates a parameterized class utility. Instantiated class utilities are created by supplying the actual values for the formal parameters of the parameterized class utility.

An instantiated class utility is displayed as a 3-part box, with the class name in the top part, a list of attributes (with optional types and values) in the middle part, and a list of operations (with optional argument lists and return types) in the bottom part.

Class Hierarchy: Item>Class>InstantiatedClassUtility

SubClasses of InstantiatedClassUtility

This class has no subclasses.



Properties Specific to InstantiatedClassUtility

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.
Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns TRUE if this class is a fundamental type.

Properties	Inherited From	Description
HasStateDiagram	Class	Returns TRUE if the class has an associated state diagram, otherwise FALSE.
IsNested	Class	Returns TRUE if the Class is nested, otherwise FALSE.
Kind	Class	The kind of Class.
Name	Item	Name of the item.
Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to InstantiatedClassUtility

Name	Kind	Class	Description
Instantiates	0..n	ParameterizedClass	The parameterized class utility that this instantiated class utility instantiates.

Item
Rose Domain

Item maps to RoseItem objects. Every RoseItem is a model element and therefore inherits all Element properties and methods. Item specifies the type of model element that the stereotype settings apply to. Valid items include:

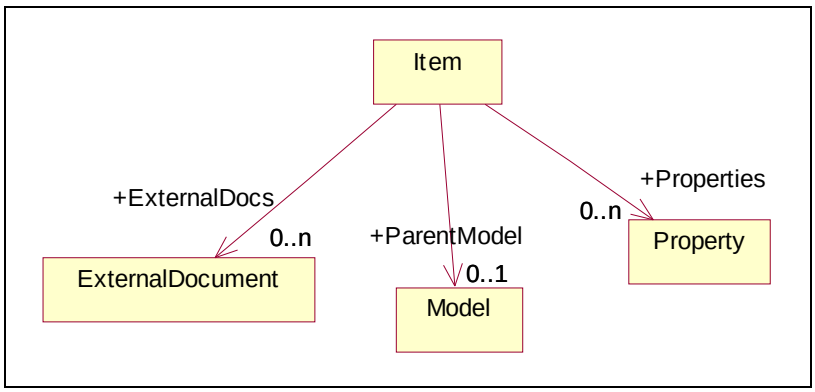
Class
Component
Package (includes logical package, use case package, and component package)
Logical Package
Component Package
Use Case Package
Processor
Device
Use Case
Association
Generalization
Dependency
Connection
Class Attribute
Operation

The default setting is Class.

Class Hierarchy: Item

SubClasses of Item

Action
Activity
Association
Attribute
Class
Decision
Link
Message
Model
Module
ModuleVisibilityRelationship
Node
ObjectInstance
Operation
Package
Parameter
Process
Relationship
State
StateTransition
Subsystem
SyncItem
UseCase



Properties Specific to Item

Properties	Inherited From	Description
Documentation		Documentation for the item.
Name		Name of the item.
QualifiedName		Qualified name of the item.
Stereotype		Stereotype of the item.
UniqueID		The unique ID of the item.

Relationships Specific to Item

Name	Kind	Class	Description
ExternalDocs	0..n	ExternalDocument	ExternalDocuments associated with this Item.
ParentModel	0..1	Model	Parent Model associated with this Item.
Properties	0..n	Property	The Property artifact types associated with this Item.

Link
Rose Domain

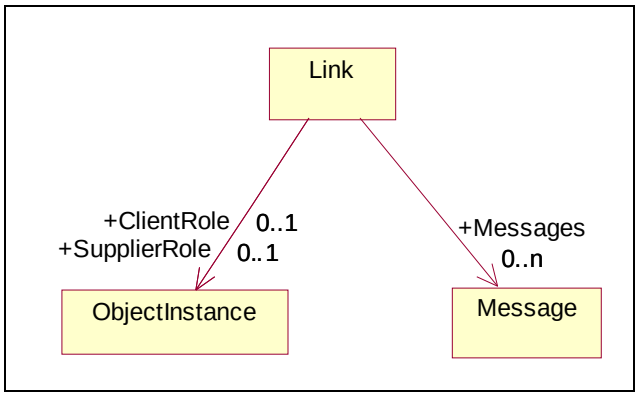
Objects interact through their links to other objects. A Link is an instance of an association, in the same way that an object is an instance of a class.

Rose Link properties and methods allow you to define links between objects and determine the nature of the objects' associations.

Class Hierarchy: Item>Link

SubClasses of Link

This class has no subclasses.



Properties Specific to Link

Properties	Inherited From	Description
ClientIsShared		True if the Shared box is checked on the client side; otherwise False.
ClientVisibility		One of Unspecified, Field, Parameters, Local, or Global.
Documentation	Item	Documentation for the item.
IsLinkToSelf		True if the link goes from an object to itself.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierIsShared		True if the Shared box is checked on the supplier side; otherwise False.
SupplierVisibility		One of Unspecified, Field, Parameters, Local, or Global.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Link

Name	Kind	Class	Description
ClientRole	0..1	ObjectInstance	The client object instance (role) of the Link.
Messages	0..n	Message	Messages associated with the Link.
SupplierRole	0..1	ObjectInstance	The supplier object instance (role) of the Link.

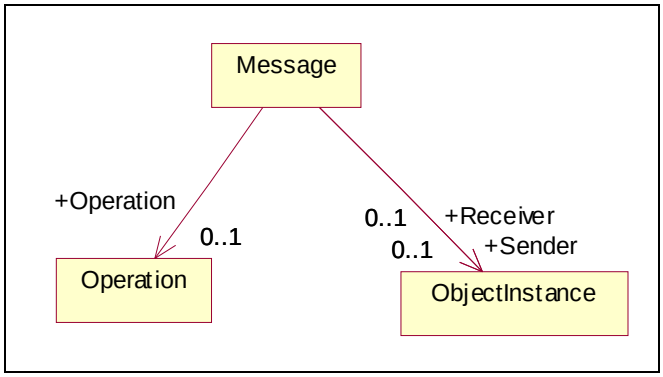
Message
Rose Domain

Any message associated with an object. Messages define the interaction between objects. The Rose message class inherits all of the Item (RoseItem) properties and methods. In addition message class methods allow you to retrieve message sender and receiver, along with other message-specific information.

Class Hierarchy: Item>Message

SubClasses of Message

This class has no subclasses.



Properties Specific to Message

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Frequency		Frequency of the message.
HierarchicalSeqNumber		Hierarchical sequence number of the message.
IsOperation		Boolean value indicating whether the message is associated with an operation.
Name	Item	Name of the item.
NameWithoutParentheses		Name of a message without the parameters enclosed in parentheses from Rose that are added when a message is associated with a class operation.
QualifiedName	Item	Qualified name of the item.
SeqNumber		Sequence number of the message.
Stereotype	Item	Stereotype of the item.

Properties	Inherited From	Description
Synchronization		Concurrency semantics for the operation named in the Operations Field; one of Simple, Synchronous, Balking, Timeout or Asynchronous.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Message

Name	Kind	Class	Description
Operation	0..1	Operation	The associated Operation with this Message.
Receiver	0..1	ObjectInstance	Object that receives the Message.
Sender	0..1	ObjectInstance	Object that sends the Message.

MetaClass
Rose Domain

A metaclass is a class whose instances are classes rather than objects. Metaclasses provide operations for initializing class variables and serve as repositories to hold class variables where a single value is required by all objects of a class. Smalltalk and CLOS support the use of metaclasses. C++ does not directly support metaclasses.

A metaclass is displayed as a 3-part box, with the class name in the top part, a list of attributes (with optional types and values) in the middle part, and a list of operations (with optional argument lists and return types) in the bottom part.

Not all languages directly support metaclasses.

Class Hierarchy: Item>Class>MetaClass

SubClasses of MetaClass

This class has no subclasses.

Properties Specific to MetaClass

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.
Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns TRUE if this class is a fundamental type.
HasStateDiagram	Class	Returns TRUE if the class has an associated state diagram, otherwise FALSE.
IsNested	Class	Returns TRUE if the Class is nested, otherwise FALSE.
Kind	Class	The kind of Class.
Name	Item	Name of the item.

Properties	Inherited From	Description
Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to MetaClass

This class has no relationships.

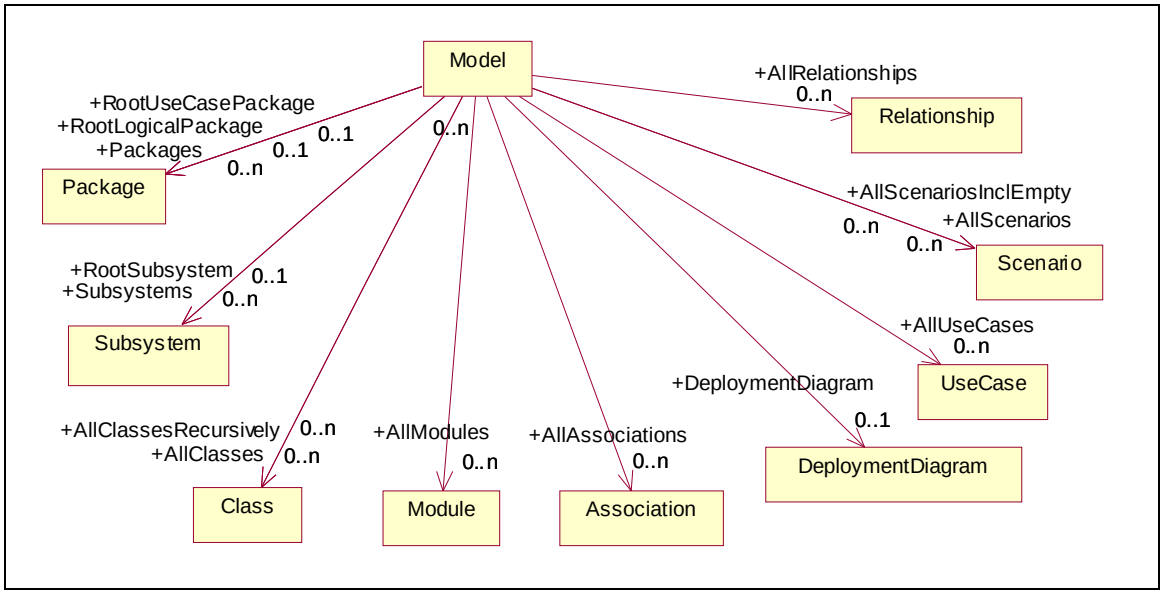
Model
Rose Domain

A Rose model file. A model file contains a Rose model, which describes your problem domain and system software. Model files use the default extension .mdl. Models are the highest hierarchical elements of the Rose source domain. Most templates start with connections to a Model.

Class Hierarchy: Item>Model

SubClasses of Model

This class has no subclasses.



Properties Specific to Model

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
DriveLetter		Drive letter in the path of the Model.
Extension		The segment of a SimpleName following the last period. For example, the Extension of c:\bill\file.txt is txt. If the SimpleName contains no period, then Extension returns a null string.
FullName		The full name, including the path, of the Model.
Name	Item	Name of the item.

Properties	Inherited From	Description
NameMinusExtension		Segment of a SimpleName preceding the last period. For example, the NameMinusExtension of c:\bill\file.txt is file. If the SimpleName contains no period, then NameMinusExtension returns the SimpleName.
NamePrefix		The simple name of the model file with the .<extension> removed (such as rose for rose.mdl).
ParentDirectoryPath		Directory containing the object.
Path		The complete pathname of an object. For example, c:\bill\file.txt
QualifiedName	Item	Qualified name of the item.
SimpleName		The simple name of the Model. The context-independent portion of an object's name. For example, the SimpleName of c:\bill\file.txt is file.txt.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Model

Name	Kind	Class	Description
AllAssociations	0..n	Association	All associations in the Model.
AllClasses	0..n	Class	All classes in the Model, including actors.
AllClassesRecursively	0..n	Class	All classes in the model, including their nested classes recursively. Note that AllClasses does not return nested classes.
AllModules	0..n	Module	All modules in the Model (including subsystems).
AllRelationships	0..n	Relationship	All relationships in the Model.
AllScenarios	0..n	Scenario	All scenario diagrams in the Model that are not empty.
AllScenariosInclEmpty	0..n	Scenario	All scenario diagrams in the Model, with and without content.
AllUseCases	0..n	UseCase	All use cases in the Model.
DeploymentDiagram	0..1	DeploymentDiagram	The deployment diagram (process diagram) for the Model.

Name	Kind	Class	Description
Packages	0..n	Package	All packages in the Model, including use case packages (but not including subsystems in the Component View).
RootLogicalPackage	0..1	Package	The highest-level package in the Model; its name is Logical View. All other packages are nested beneath it.
RootSubsystem	0..1	Subsystem	The highest-level subsystem in the Model; its name is Component View. All other subsystems are nested beneath it.
RootUseCasePackage	0..1	Package	The root use case package in the Model; its name is UseCase View. All other use-case packages are nested beneath it.
Subsystems	0..n	Subsystem	All subsystem components in the Model.

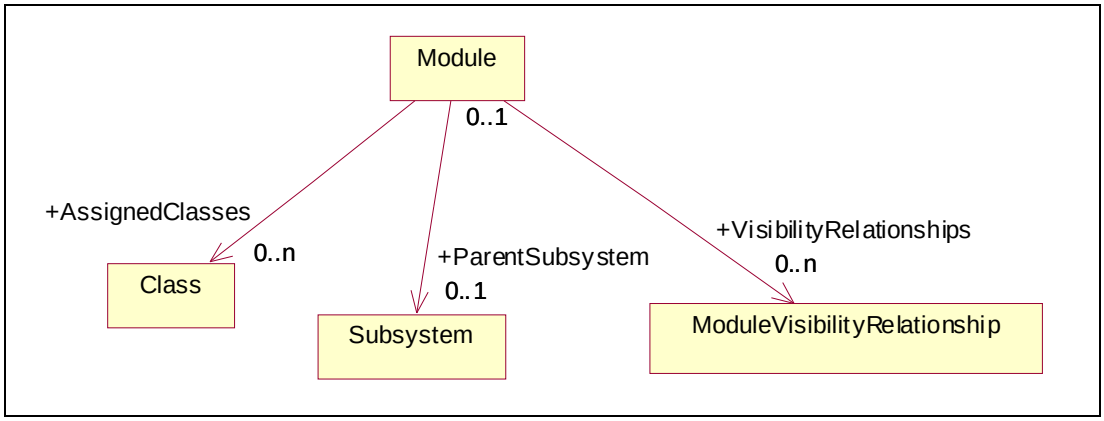
Module
Rose Domain

A module is a unit of code that serves as a building block for the physical structure of a system. The module class exposes properties and methods that allow you to define and manipulate the characteristics of modules.

Class Hierarchy: Item>Module

SubClasses of Module

This class has no subclasses.



Properties Specific to Module

Properties	Inherited From	Description
AssignedLanguage		Specifies the programming language assigned to the Module.
Declarations		Text of the declarations belonging to the Module.
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
Part		Defines the Module as a part of a subsystem: a Specification, Body, Generic, or Main.
Path		The path of the Module.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
Type		The type of Module.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Module

Name	Kind	Class	Description
AssignedClasses	0..n	Class	Associated classes to this Module.
ParentSubsystem	0..1	Subsystem	Parent subsystem of this Module.
VisibilityRelationships	0..n	ModuleVisibilityRelationship	The module visibility relationships for this Module.

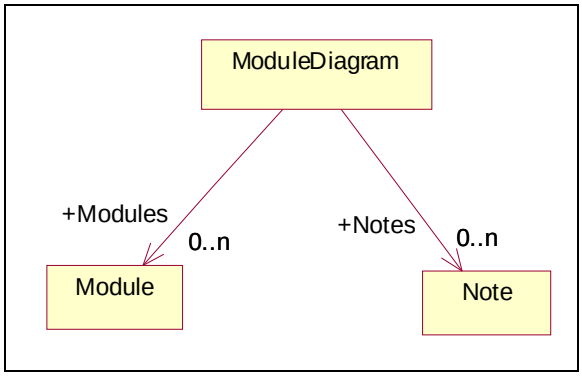
ModuleDiagram
Rose Domain

A module diagram maps the allocation classes and objects to modules. The module diagram class exposes properties and methods that allow you to add, retrieve, and delete classes and objects in a module diagram.

Class Hierarchy: Diagram>ModuleDiagram

SubClasses of ModuleDiagram

This class has no subclasses.



Properties Specific to ModuleDiagram

Properties	Inherited From	Description
Documentation	Diagram	The documentation text associated with the Diagram.
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.
Name	Diagram	Name of the Diagram.
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to ModuleDiagram

Name	Kind	Class	Description
Modules	0..n	Module	Modules in the module diagram.
Notes	0..n	Note	Notes that appear in the module diagram.

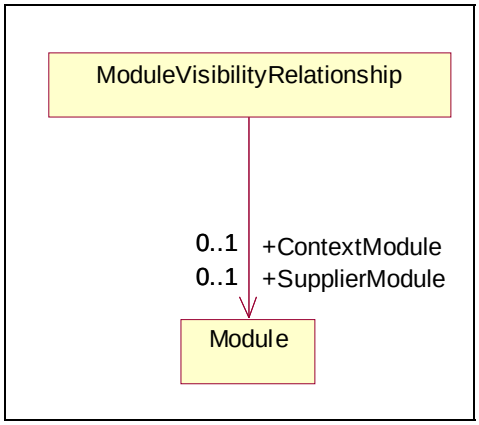
ModuleVisibilityRelationship
Rose Domain

The ModuleVisibilityRelationship class describes the context and supplier relationship between modules.

Class Hierarchy: Item>ModuleVisibilityRelationship

SubClasses of ModuleVisibilityRelationship

This class has no subclasses.



Properties Specific to ModuleVisibilityRelationship

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ModuleVisibilityRelationship

Name	Kind	Class	Description
ContextModule	0..1	Module	The consumer module.
SupplierModule	0..1	Module	The supplier module.

Node
Rose Domain

Node is an abstract class for processors and devices.

Class Hierarchy: Item>Node

SubClasses of Node

Device
Processor

Properties Specific to Node

Properties	Inherited From	Description
Characteristics		Characteristics of the processor or device.
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Node

This class has no relationships.

Note
Rose Domain

A note captures the assumptions and decisions applied during analysis and design. Notes may contain any information, including plain text, fragments of code, or references to other documents. Notes are also used as a means of linking diagrams. A note holds an unlimited amount of text and can be sized accordingly.

Class Hierarchy: Note

SubClasses of Note

This class has no subclasses.

Properties Specific to Note

Properties	Inherited From	Description
CollIndex		The ordinal position of the note in the collection from which it comes.
Text		Text of the Note.
Type		The type of Note.

Relationships Specific to Note

This class has no relationships.

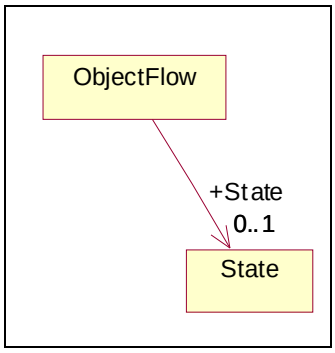
ObjectFlow
Rose Domain

The ObjectFlow class is an abstract class that exposes Rose's object flow functionality in the extensibility interface. An object flow on an activity diagram represents the relationship between an activity and the object that creates it (as an output) or uses it (as an input).

Class Hierarchy: Item>Relationship>ObjectFlow

SubClasses of ObjectFlow

This class has no subclasses.



Properties Specific to ObjectFlow

Properties	Inherited From	Description
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Documentation	Item	Documentation for the item.
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.

Properties	Inherited From	Description
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ObjectFlow

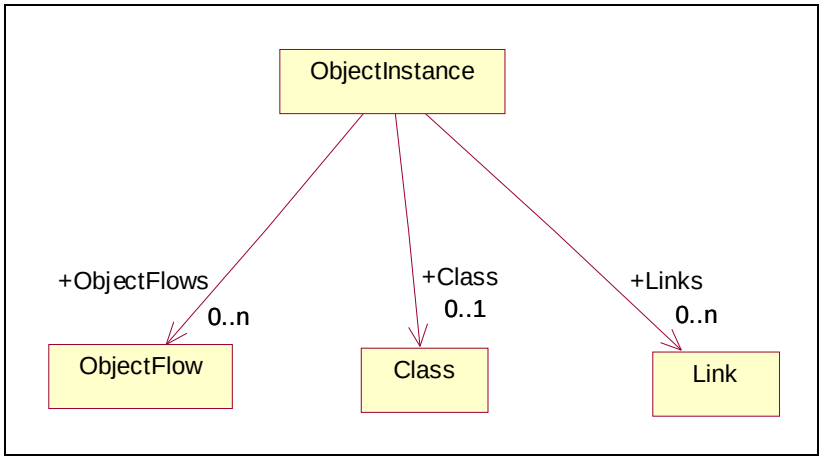
Name	Kind	Class	Description
State	0..1	State	Associated State of the ObjectFlow.

ObjectInstance
Rose Domain

The ObjectInstance class exposes a set of properties and methods that:
Determine the characteristics of objects in a model (for example, the class associated with the object and whether multiple instances of the object exist)
Allow you to retrieve objects from a model

Class Hierarchy: Item>ObjectInstance

SubClasses of ObjectInstance
This class has no subclasses.



Properties Specific to ObjectInstance

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
IsClass		True if the ObjectInstance is a class.
MultipleInstances		True if the Multiple Instances box is checked; otherwise False.
MyClassName		The ObjectInstance class name.
Name	Item	Name of the item.
Persistence		Persistent, Static, or Transient depending on the value of the Persistence radio control in the object specification.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ObjectInstance

Name	Kind	Class	Description
Class	0..1	Class	The class of the object.
Links	0..n	Link	The links associated with the object.
ObjectFlows	0..n	ObjectFlow	The associated ObjectFlows with this ObjectInstance.

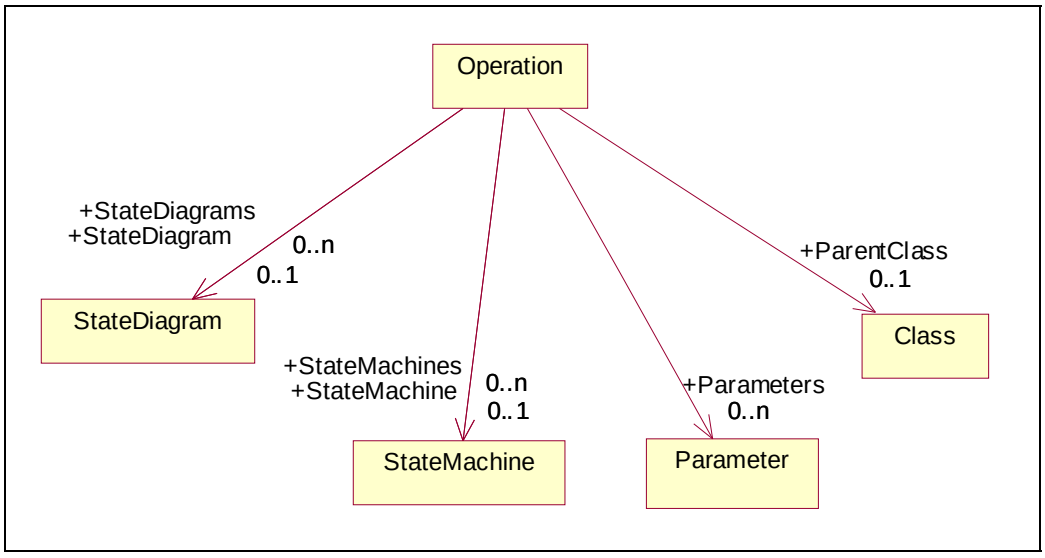
Operation
Rose Domain

Operations denote services provided by the class. Operations can be methods for accessing and modifying class fields or methods that implement characteristic behaviors of a class.
The operations of a class are listed in the Operations list box in the class specification. Rational Rose stores operation information in an operation specification. You can access operation specifications only through the class specification.

Class Hierarchy: Item>Operation

SubClasses of Operation

This class has no subclasses.



Properties Specific to Operation

Properties	Inherited From	Description
AdaImage		An Ada code segment that represents the declaration of the operation. This image is derived from the operation name and the operation parameters. Although the AdaImage is semantically consistent with your actual code, it may differ in terms of format, depending on the rules and styles you use for code generation and/or reverse engineering.
COMImage		A COM code segment that represents the declaration of the operation.

Properties	Inherited From	Description
Concurrency		Denotes the semantics of the operation in the presence of multiple threads of control. Returns Sequential, Guarded, or Synchronous, depending on the state of the Concurrency radio control in the More dialog of the operation specification.
CppImage		A C++ code segment that represents the prototype of the operation. This image is derived from the operation name and the operation parameters. Although the C++Image is semantically consistent with your actual code, it may differ in terms of format, depending on the rules and styles you use for code generation and/or reverse engineering.
Documentation	Item	Documentation for the item.
Exceptions		Textual list of the exceptions that can be raised by the operation. The Exceptions text field appears in the More dialog of the operation specification.
ExportControl		Specifies the type of access allowed by the class for this operation. Will return Public, Protected, Private, or Implementation, depending on the state of the Export Control radio control in the operation specification.
HasStateDiagram		True if the operation has an associated StateDiagram
JavaImage		A Java code segment that represents the declaration of the operation.
Name	Item	Name of the item.
Postconditions		Text describing the post-conditions of the operation. The PostText is that text which appears in the Dynamic Semantics field of the operation specification when the Post radio button is selected.

Properties	Inherited From	Description
Preconditions		Text describing the preconditions of the operation. The PreText is that text which appears in the Dynamic Semantics field of the operation specification when the Pre radio button is selected.
Protocol		The Protocol field lists a set of operations that a client may perform on an object and the legal orderings in which they may be invoked. The protocol of an operation has no semantic impact. The Protocol text field appears in the More dialog of the operation specification.
Qualification		Identifies language-specific features that allow you to qualify the method. The Qualification text field appears in the More dialog of the operation specification.
QualifiedName	Item	Qualified name of the item.
ReturnType		For operations that are functions, refers to the class that is returned by the function. The ReturnClass text field appears in the Return Class field on the operation specification.
Semantics		Text describing the action of the main operation. The SemanticsText is that text which appears in the Dynamic Semantics field of the operation specification when the Semantics radio button is selected.
Size		Text describing the size of the class.
Stereotype	Item	Stereotype of the item.
Time		A statement about the relative or absolute time required to complete an operation. The Time text field appears in the More dialog of the operation specification.
UMLImage		The image of the operation and parameters using UML standard notation.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Operation

Name	Kind	Class	Description
Parameters	0..n	Parameter	The formal parameters of the Operation. These appear in the Arguments list box in the operation specification.
ParentClass	0..1	Class	Class to which this Operation belongs.
StateDiagram	0..1	StateDiagram	The top-level state diagram associated with this Operation.
StateDiagrams	0..n	StateDiagram	All state diagrams associated with this Operation.
StateMachine	0..1	StateMachine	The top-level state machine associated with this Operation.
StateMachines	0..n	StateMachine	All state machines associated with this Operation.

Package
Rose Domain

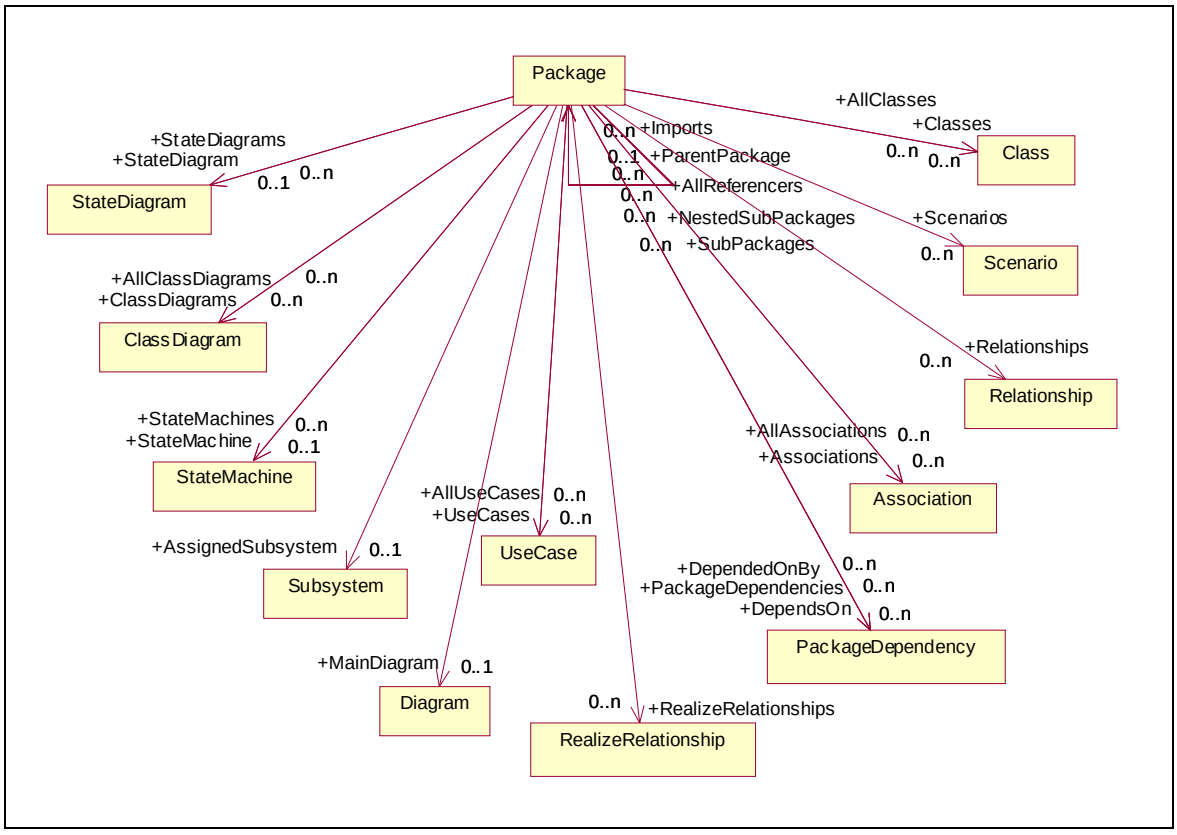
Packages serve to partition the logical model of a system. They are clusters of highly related classes that are themselves cohesive, but are loosely coupled relative to other such clusters. You can use packages to group classes and other packages. Rational Rose stores data describing the package in a package specification.

Note: When you create an OPEN command directly to a package, be sure to specify the name of the .mdl file and the name of the package, even if the package is contained in a separate .cat file.

Class Hierarchy: Item>Package

SubClasses of Package

This class has no subclasses.



Properties Specific to Package

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Global		True if the Package is global, otherwise False.

Properties	Inherited From	Description
HasAssignedSubsystem		True if the Package has a subsystem associated with it, otherwise False.
HasStateDiagram		True if the Package has a state/activity diagram.
IsUseCasePackage		True if the Package is a descendent of the UseCase View package, otherwise False.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Package

Name	Kind	Class	Description
AllAssociations	0..n	Association	All associations that are defined in this Package, or in any nested packages.
AllClassDiagrams	0..n	ClassDiagram	All class diagrams that are defined in this Package, or in any nested packages.
AllClasses	0..n	Class	All classes that are defined in this Package, or in any nested packages.
AllReferencers		Package	All packages that import this Package. Does not include indirect referencers.
AllUseCases	0..n	UseCase	All use cases that are defined in this Package, or in any nested packages.
AssignedSubsystem	0..1	Subsystem	The subsystem associated with this Package, as specified in the package specification.
Associations	0..n	Association	All associations that are immediate members of this Package.
ClassDiagrams	0..n	ClassDiagram	All class diagrams that are immediate members of this Package.
Classes	0..n	Class	All classes that are immediate members of this Package. All member classes are returned, regardless of whether they appear on any diagrams.
DependedOnBy	0..n	PackageDependency	Associates this Package as the supplier package in a package dependency

Name	Kind	Class	Description
Dependencies	0..n	Relationship	The relationships of type DependencyRelation.
DependsOn	0..n	PackageDependency	Associates this Package as the receiver package in a package dependency
Imports		Package	All packages that are imported by this Package. Does not include indirect dependencies. For example if A imports B and B imports C, A does not directly import C.
MainDiagram	0..1	Diagram	The first class or use case diagram called "Main," or the first class or use case diagram, in that order. An empty diagram will ot be included.
NestedSubPackages		Package	All packages that are descendents of this Package.
PackageDependencies	0..n	PackageDependency	The relationships of type PackageDependency.
PackageRelationships	0..n	Relationship	The relationships of types Relationships, CategoryDependency, RealizeRelation, and DependencyRelation.
ParentPackage		Package	The enclosing Package. This relationship will result in an error if applied to the TopLevelCategory.
RealizeRelationships	0..n	RealizeRelationship	The relationships of type RealizeRelationship.
Relationships	0..n	Relationship	All the relationships defined within this Package.
Scenarios	0..n	Scenario	The scenario diagrams in the package.
StateDiagram	0..1	StateDiagram	The top-level state/activity diagram associated with this Package.
StateDiagrams	0..n	StateDiagram	All state diagrams associated with this Package.
StateMachine	0..1	StateMachine	The top-level state machine associated with this Package.
StateMachines	0..n	StateMachine	All state machines associated with this Package.
SubPackages		Package	All packages that are immediate children of this Package.
UseCases	0..n	UseCase	All use cases that are immediate members of this Package.

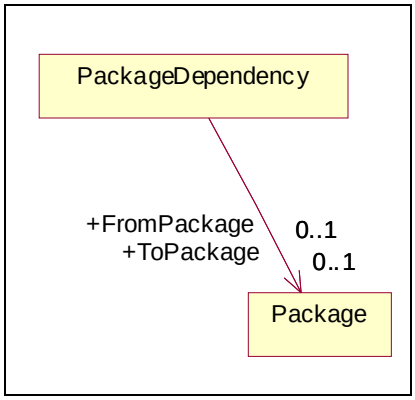
PackageDependency
Rose Domain

The package dependency indicates that one package in a model uses the services or facilities of another.

Class Hierarchy: Item>PackageDependency

SubClasses of PackageDependency

This class has no subclasses.



Properties Specific to PackageDependency

Properties	Inherited From	Description
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Documentation	Item	Documentation for the item.
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.

Properties	Inherited From	Description
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName		Name of the package that is the supplier in the package dependency.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to PackageDependency

Name	Kind	Class	Description
FromPackage	0..1	Package	The supplier package.
ToPackage	0..1	Package	The receiver package.

Parameter
Rose Domain

Formal parameter of an operation, instantiated class, or instantiated class utility.

Class Hierarchy: Item>Parameter

SubClasses of Parameter

This class has no subclasses.

Properties Specific to Parameter

Properties	Inherited From	Description
Const		Returns TRUE if the parameter is constant; otherwise False.
Documentation	Item	Documentation for the item.
InitValue		Initial value of the parameter.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
Type		The type of the parameter.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Parameter

This class has no relationships.

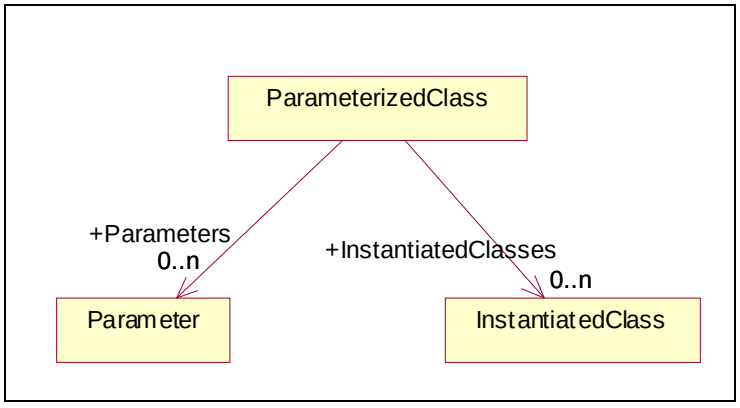
ParameterizedClass
Rose Domain

A parameterized class is a template for creating any number of instantiated classes that follow its format. A parameterized class declares formal parameters, which can be classes, objects, or operations.

Class Hierarchy: Item>Class>ParameterizedClass

SubClasses of ParameterizedClass

This class has no subclasses.



Properties Specific to ParameterizedClass

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.
Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns TRUE if this class is a fundamental type.
HasStateDiagram	Class	Returns TRUE if the class has an associated state diagram, otherwise FALSE.

Properties	Inherited From	Description
IsNested	Class	Returns TRUE if the Class is nested, otherwise FALSE.
Kind	Class	The kind of Class.
Name	Item	Name of the item.
Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ParameterizedClass

Name	Kind	Class	Description
InstantiatedClasses	0..n	InstantiatedClass	All instantiated classes of this parameterized class.
Parameters	0..n	Parameter	Formal, generic parameters declared by the parameterized class. The parameters appear in the Parameters list box in the More dialog of the class specification.

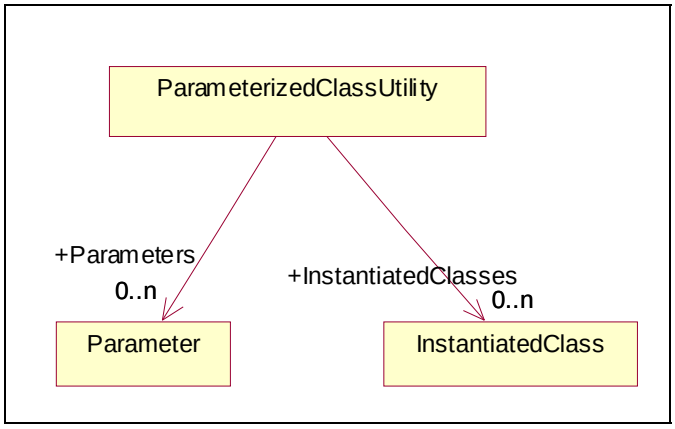
ParameterizedClassUtility
Rose Domain

A parameterized class utility is a set of operations or functions that are not associated with a higher level class (free subprograms) and are defined in terms of formal parameters. Parameterized class utilities are used as templates for creating instantiated class utilities.

Class Hierarchy: Item>Class>ParameterizedClassUtility

SubClasses of ParameterizedClassUtility

This class has no subclasses.



Properties Specific to ParameterizedClassUtility

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.
Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns TRUE if this class is a fundamental type.

Properties	Inherited From	Description
HasStateDiagram	Class	Returns TRUE if the class has an associated state diagram, otherwise FALSE.
IsNested	Class	Returns TRUE if the Class is nested, otherwise FALSE.
Kind	Class	The kind of Class.
Name	Item	Name of the item.
Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ParameterizedClassUtility

Name	Kind	Class	Description
InstantiatedClasses	0..n	InstantiatedClass	All instantiated class utilities of this parameterized class utility.
Parameters	0..n	Parameter	Formal, generic parameters declared by the parameterized class utility. The parameters appear in the Parameters list box in the More dialog of the class specification.

Process
Rose Domain

A process transforms data values. Lowest-level processes are pure functions without side effects.

Class Hierarchy: Item>Process

SubClasses of Process
This class has no subclasses.

Properties Specific to Process

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
Priority		The priority of the Process.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

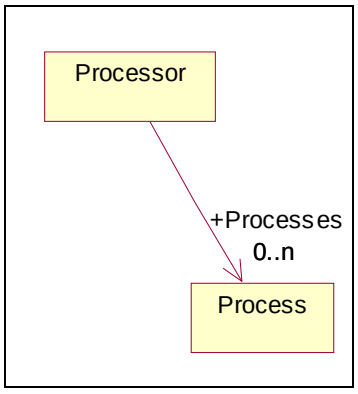
Relationships Specific to Process
This class has no relationships.

Processor
Rose Domain

A processor is a hardware component capable of executing programs.

Class Hierarchy: Item>Node>Processor

SubClasses of Processor
This class has no subclasses.



Properties Specific to Processor

Properties	Inherited From	Description
Characteristics	Node	Characteristics of the processor or device.
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Scheduling		The text in the Scheduling field of the processor specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Processor

Name	Kind	Class	Description
Processes	0..n	Process	Processes defined by this Processor.

Property
Rose Domain

A code-generation property associated with the model, a package, a subsystem, a class, an association, a has relationship, an attribute, a module, or an operation.

Class Hierarchy: Property

SubClasses of Property
This class has no subclasses.

Properties Specific to Property

Properties	Inherited From	Description
Name		Name of the property.
ParentUID		The unique id of this property's parent artifact type.
ToolName		The name of the tool, or tab, for the property, such as “cg” or “DDL.”
Value		String equivalent of the value associated with the property.

Relationships Specific to Property
This class has no relationships.

RealizeRelationship
Rose Domain

A realize relationship between a logical class and a component class shows that the component class realizes the operations defined by the logical class.

Class Hierarchy: Item>Relationship>RealizeRelationship

SubClasses of RealizeRelationship

This class has no subclasses.

Properties Specific to RealizeRelationship

Properties	Inherited From	Description
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Documentation	Item	Documentation for the item.
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.

Properties	Inherited From	Description
UniqueID	Item	The unique ID of the item.

Relationships Specific to RealizeRelationship

This class has no relationships.

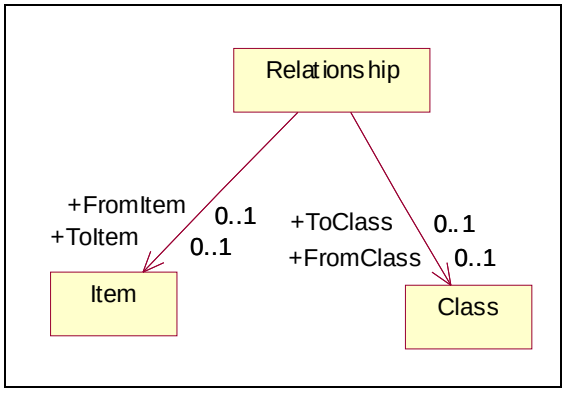
Relationship
Rose Domain

A semantic connection between two classes. Rational Rose stores relationship information in a relationship specification.

Class Hierarchy: Item>Relationship

SubClasses of Relationship

- HasRelationship
- InheritRelationship
- ObjectFlow
- PackageDependency
- RealizeRelationship
- Role
- UsesRelationship



Properties Specific to Relationship

Properties	Inherited From	Description
ClientCardinality		Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Documentation	Item	Documentation for the item.
ExportControl		Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.

Properties	Inherited From	Description
Kind		Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierCardinality		Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName		Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Relationship

Name	Kind	Class	Description
FromClass	0..1	Class	The client class. For example, if A Has a B, A is the client, or From class.
FromItem	0..1	Item	The client that owns the relationship.
ToClass	0..1	Class	The supplier class. For example, if A Has a B, B is the supplier, or To class.
ToItem	0..1	Item	The supplier of the relationship to the client.

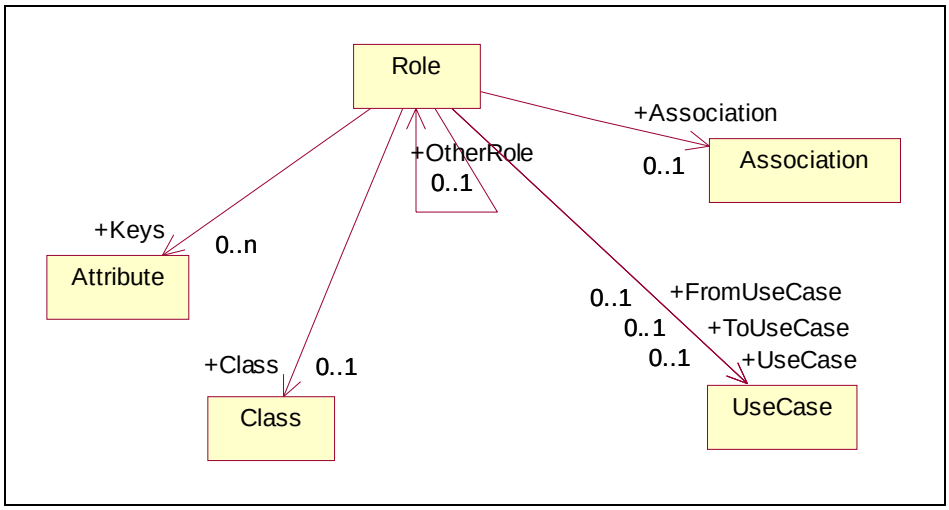
Role
Rose Domain

The purpose or capacity where one class associates with another.

Class Hierarchy: Item>Relationship>Role

SubClasses of Role

This class has no subclasses.



Properties Specific to Role

Properties	Inherited From	Description
Aggregate		Returns TRUE if the role is an aggregate relationship.
Cardinality		Cardinality of this Role.
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Constraints		Text of the Contraints field in the Role specification.
Containment		Specifies the physical containment of the role. Returns Value, Reference, or Unspecified, depending on the state of the Containment radio control on the Role specification.
Documentation	Item	Documentation for the item.

Properties	Inherited From	Description
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
Friend		Returns TRUE if the Friend check box is selected in the Role specification, otherwise FALSE.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.
Navigable		Returns TRUE if the Navigable check box is selected, otherwise FALSE.
QualifiedName	Item	Qualified name of the item.
Static		Returns TRUE if the Static check box is selected in the Role specification, otherwise FALSE.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Role

Name	Kind	Class	Description
Association	0..1	Association	The association that this Role is a part of.
Class	0..1	Class	The class associated with this Role

Name	Kind	Class	Description
FromUseCase	0..1	UseCase	The supplier use case of the Role, if it is a use case.
Keys	0..n	Attribute	Each key is an attribute that uniquely defines a single target object.
OtherRole		Role	The role at the other end of the association.
ToUseCase	0..1	UseCase	The client use case of the inherits relationship, if it is a use case.
UseCase	0..1	UseCase	The use case associated with this Role.

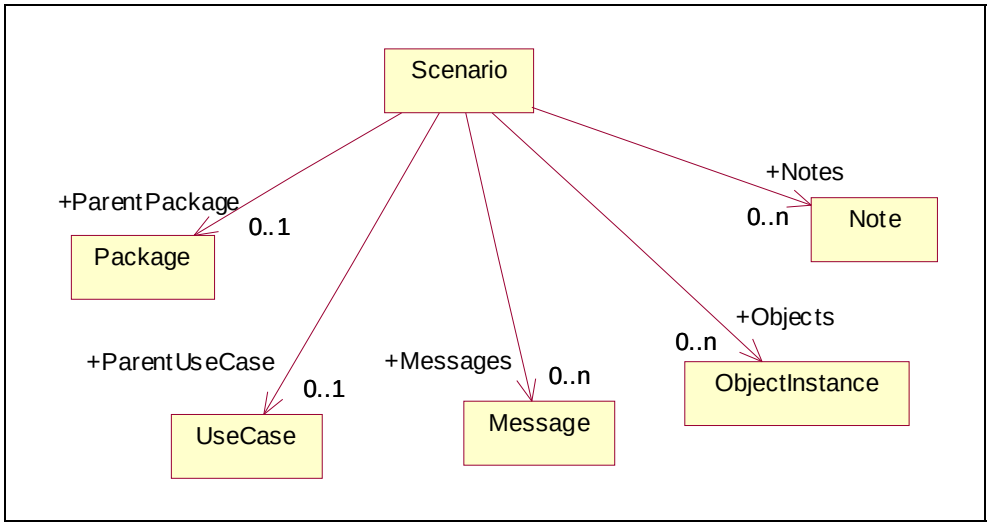
Scenario
Rose Domain

A scenario is an instance of a use case; it is an outline of events that occur during system execution.

Class Hierarchy: Diagram>Scenario

SubClasses of Scenario

This class has no subclasses.



Properties Specific to Scenario

Properties	Inherited From	Description
DiagramType		The diagram type of this scenario
Documentation	Diagram	The documentation text associated with the Diagram.
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.
Name	Diagram	Name of the Diagram.
ParentKind		The parent diagram of this scenario
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to Scenario

Name	Kind	Class	Description
Messages	0..n	Message	Messages associated with this Scenario.
Notes	0..n	Note	Notes associated with this Scenario.
Objects	0..n	ObjectInstance	Object instances associated with the Scenario.
ParentPackage	0..1	Package	Parent package of this Scenario.
ParentUseCase	0..1	UseCase	Parent use case of this Scenario.

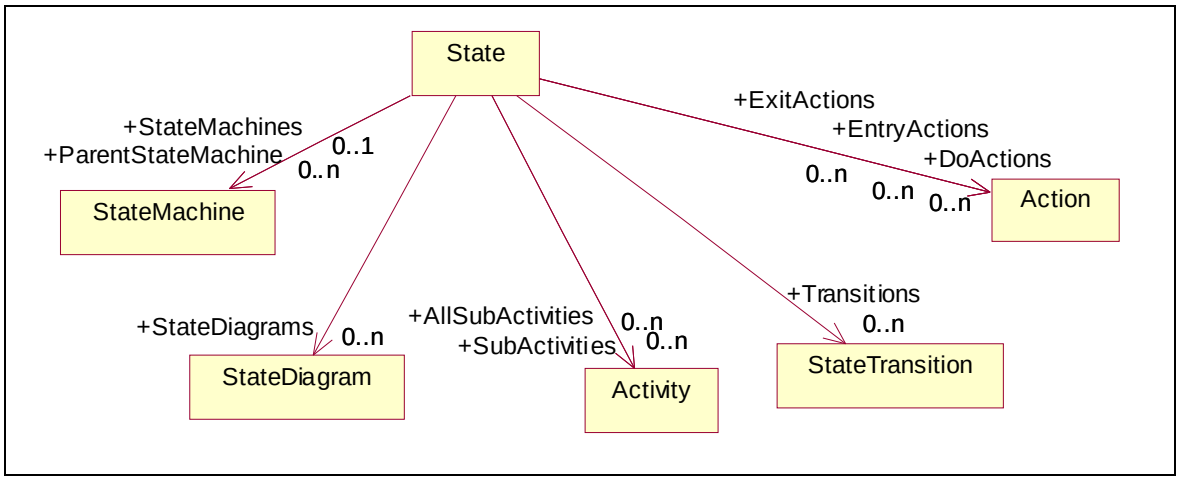
State
Rose Domain

The state of an object represents the cumulative history of its behavior. State encompasses all of the object's static properties and the current values of each property.

Class Hierarchy: Item>State

SubClasses of State

This class has no subclasses.



Properties Specific to State

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
History		The text in the History field of the state specification.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
StateKind		The kind of State. One of Start, Normal, or Stop.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to State

Name	Kind	Class	Description
AllSubActivities	0..n	Activity	All activities that are associated with this State.
DoActions	0..n	Action	The Do actions associated with this State.

Name	Kind	Class	Description
EntryActions	0..n	Action	The Entry actions associated with this State.
ExitActions	0..n	Action	The Exit actions associated with this State.
ParentStateMachine	0..1	StateMachine	The top-level state machine associated with this State.
StateDiagrams	0..n	StateDiagram	All state diagrams associated with this State.
StateMachines	0..n	StateMachine	All state machines associated with this State.
SubActivities	0..n	Activity	The activities that are part of this State.
Transitions	0..n	StateTransition	The transitions that exit from this State.

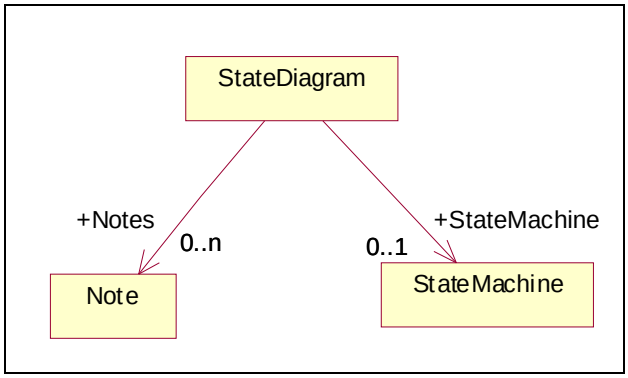
StateDiagram
Rose Domain

Depicts significant event-ordered behavior of a particular class. Each class may have one state diagram to describe its behavior.

Class Hierarchy: Diagram>StateDiagram

SubClasses of StateDiagram

This class has no subclasses.



Properties Specific to StateDiagram

Properties	Inherited From	Description
Documentation	Diagram	The documentation text associated with the Diagram.
HasStateMachine		Returns TRUE if the diagram includes a state activity model.
IsActivityDiagram		Returns TRUE if the StateDiagram is an activity diagram
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.
Name	Diagram	Name of the Diagram.
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to StateDiagram

Name	Kind	Class	Description
Notes	0..n	Note	Notes that appear in the diagram.
StateMachine	0..1	StateMachine	The top-level state machine associated with this diagram.

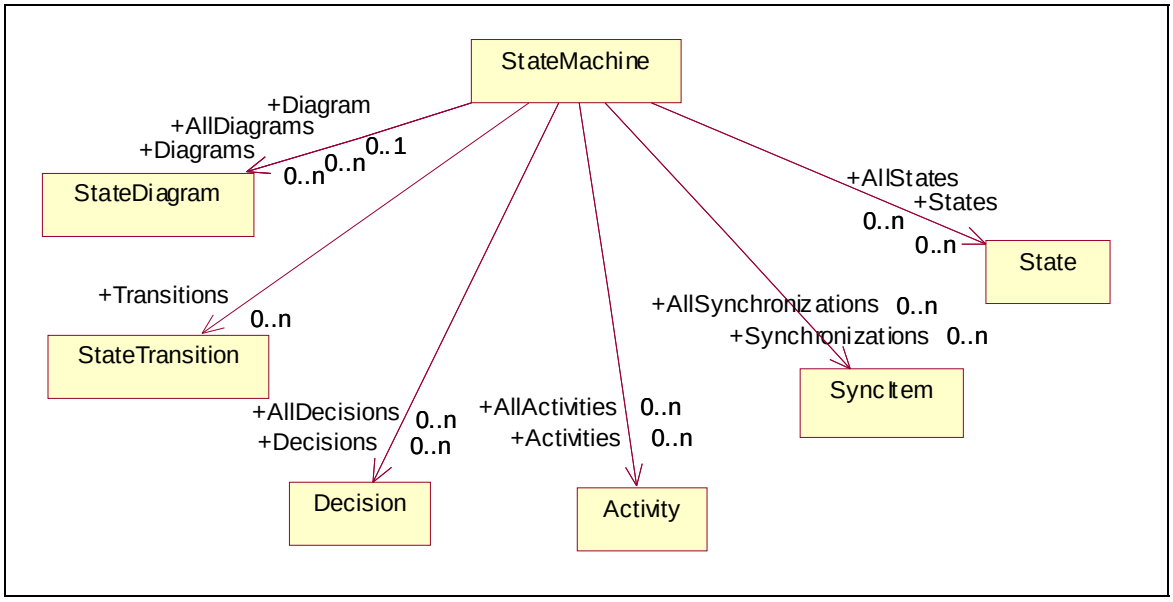
StateMachine
Rose Domain

A state machine can be defined as a behavior that specifies the valid sequences of activities that an object or interaction goes through during its life in response to events, together with its responses and actions.

Class Hierarchy: StateMachine

SubClasses of StateMachine

This class has no subclasses.



Properties Specific to StateMachine

Properties	Inherited From	Description
Documentation		The documentation for the StateMachine.
HasDiagram		True if the state activity model has at least one state or activity diagram.
Name		Name of the state activity model.
Stereotype		The stereotype of the StateMachine.
UniqueID		The internal unique identifier of the state activity model.

Relationships Specific to StateMachine

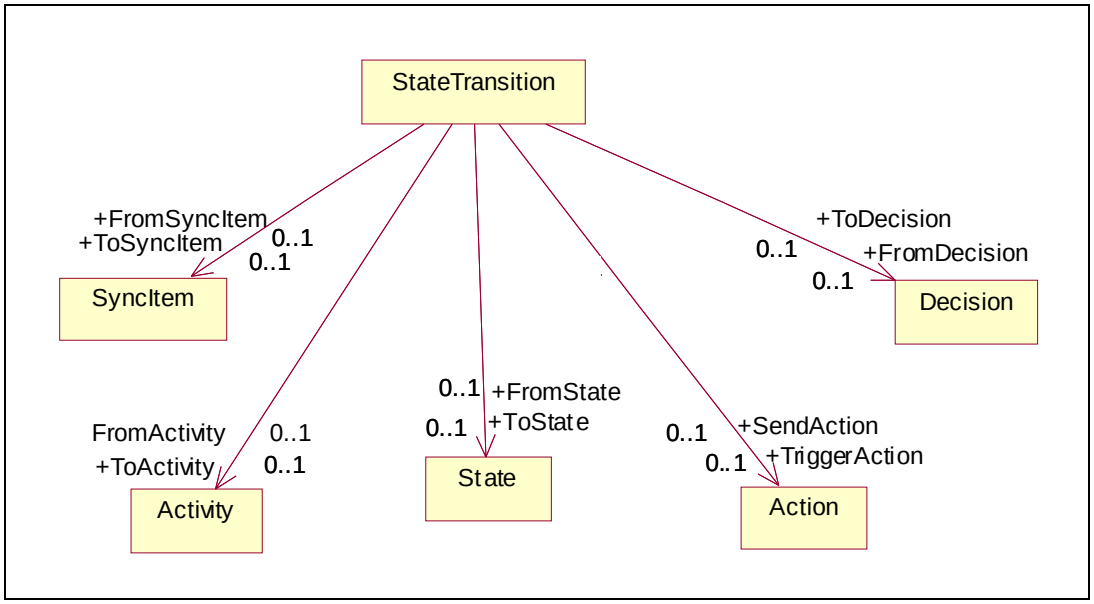
Name	Kind	Class	Description
Activities	0..n	Activity	The activities defined in this state activity model.
AllActivities	0..n	Activity	The activities defined in both this state activity model and all nested state activity models.
AllDecisions	0..n	Decision	Decisions defined in both this state activity model and all nested state activity models.
AllDiagrams	0..n	StateDiagram	Diagrams defined in both this state activity model and all nested state activity models.
AllStates	0..n	State	All states that are associated with this state activity model.
AllSynchronizations	0..n	SyncItem	Synchronizations defined in both this state activity model and all nested state activity models.
Decisions	0..n	Decision	Decisions defined in this state activity model.
Diagram	0..1	StateDiagram	The (first) state or activity diagram associated with this state activity model.
Diagrams	0..n	StateDiagram	The state or activity diagrams associated with this state activity model.
States	0..n	State	States that are part of this state activity model.
Synchronizations	0..n	SyncItem	Synchronizations defined in this state activity model.
Transitions	0..n	StateTransition	Transitions that are part of this state activity model.

StateTransition
Rose Domain

A state transition is a change of state caused by an event. Use state transitions to connect two states in a state diagram or show state transitions from a state to itself.

Class Hierarchy: Item>StateTransition

SubClasses of StateTransition
This class has no subclasses.



Properties Specific to StateTransition

Properties	Inherited From	Description
CausingArguments		Arguments that accompany the causing event.
CausingEventName		Name of the event that causes this transition.
Documentation	Item	Documentation for the item.
GuardCondition		The guard condition associated with the causing event.
KindofFromItem		Returns a string for the artifact type of the 'From', which is set to either State, or Activity or SyncItem.
KindofToItem		Returns a string for the artifact type of the 'To', which is set to either State, or Activity or SyncItem.
Name	Item	Name of the item.

Properties	Inherited From	Description
QualifiedName	Item	Qualified name of the item.
SendArguments		Arguments that accompany the trigger event.
SendEventName		Name of the event triggered by the transition.
SendTarget		Name of the object that will receive the transition event.
Stereotype	Item	Stereotype of the item.
SupplierName		Name of the object that supplies the transition event.
UniqueID	Item	The unique ID of the item.

Relationships Specific to StateTransition

Name	Kind	Class	Description
FromActivity	0..1	Activity	The Activity that this transition emanates from.
FromDecision	0..1	Decision	The Decision that this transition emanates from.
FromState	0..1	State	The State that this transition emanates from.
FromSyncItem	0..1	SyncItem	The SyncItem that this transition emanates from.
SendAction	0..1	Action	Send action of this transition.
ToActivity	0..1	Activity	The Activity that this transition leads to.
ToDecision	0..1	Decision	The Decision that this transition leads to.
ToState	0..1	State	The State that this transition leads to.
ToSyncItem	0..1	SyncItem	The SyncItem that this transition leads to.
TriggerAction	0..1	Action	The Action that triggers this transition.

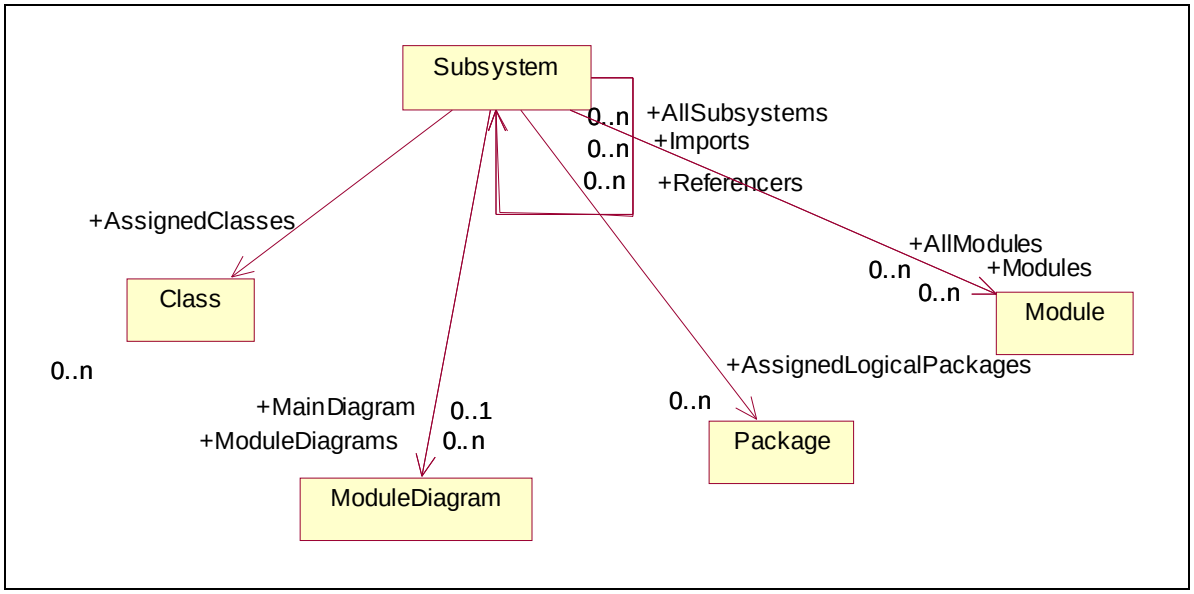
Subsystem
Rose Domain

Subsystems represent clusters of logically related components. They parallel the role played by packages for class diagrams, allowing you to partition the physical model of the system.
Each subsystem can contain components and other subsystems. Each module in your system must reside in a single subsystem or at the Component View of the model.

Class Hierarchy: Item>Subsystem

SubClasses of Subsystem

This class has no subclasses.



Properties Specific to Subsystem

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Subsystem

Name	Kind	Class	Description
AllModules	0..n	Module	All modules associated with this Subsystem.
AllSubsystems		Subsystem	All subsystems associated with this Subsystem.

Name	Kind	Class	Description
AssignedClasses	0..n	Class	The classes assigned to this Subsystem.
AssignedLogicalPackages	0..n	Package	The logical packages assigned to this Subsystem.
Imports		Subsystem	All other Subsystems that this Subsystem directly depends on. Does not include indirect dependencies. For example if A imports B and B imports C, A does not directly import C.
MainDiagram	0..1	ModuleDiagram	All main module diagram contained in this Subsystem.
ModuleDiagrams	0..n	ModuleDiagram	All module diagrams contained in this Subsystem.
Modules	0..n	Module	The modules contained in this Subsystem.
Referencers		Subsystem	All other Subsystems that directly depend on this Subsystem. Does not include indirect referencers. For example if A imports B and B imports C, A is not a direct referencer of C.

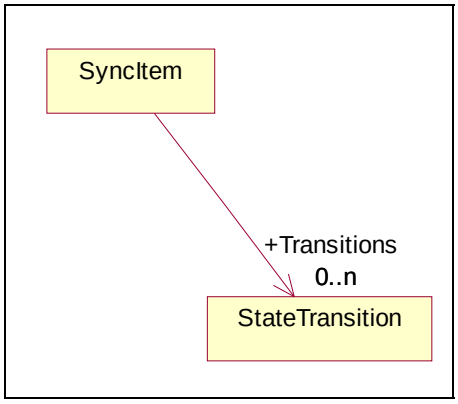
SyncItem
Rose Domain

The SyncItem class is an abstract class that exposes the Rose synchronization functionality in the extensibility interface.

Class Hierarchy: Item>SyncItem

SubClasses of SyncItem

This class has no subclasses.



Properties Specific to SyncItem

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to SyncItem

Name	Kind	Class	Description
Transitions	0..n	StateTransition	The state transitions associated with this SyncItem.

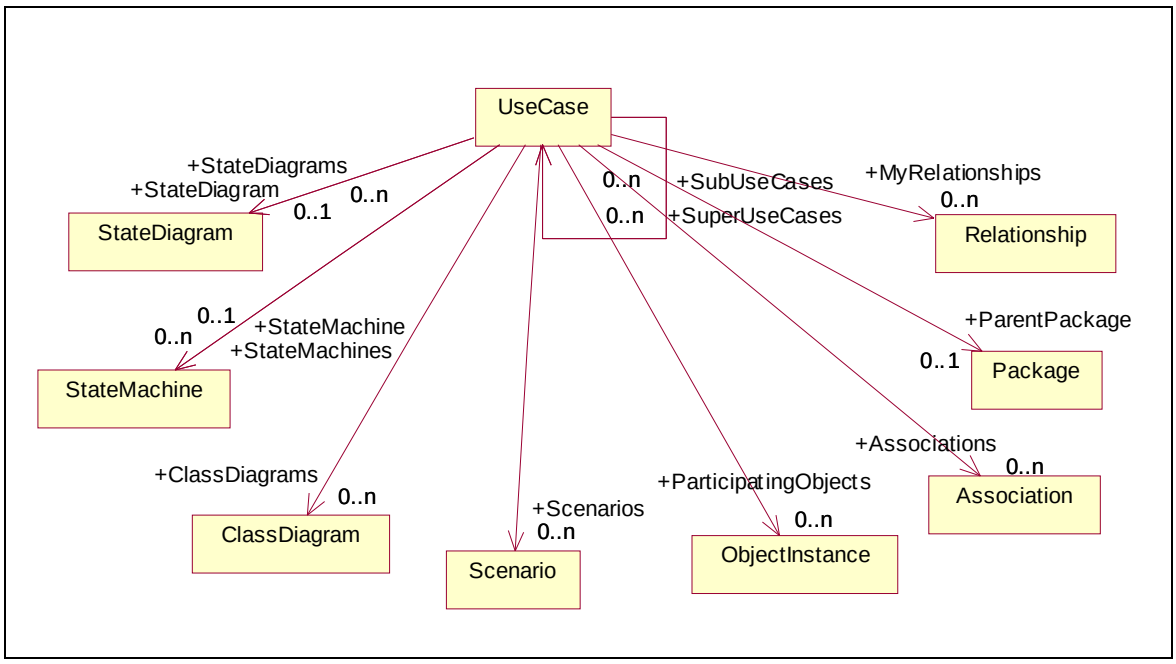
UseCase
Rose Domain

A sequence of transactions performed by a system in response to a triggering event initiated by an actor to the system. A full use case should provide a measurable value to an actor when the actor is performing a certain task. A use case contains all the events that can occur between an actor-use case pair, not necessarily the ones that will occur in any particular scenario. A use case contains a set of scenarios that explain various sequences of interaction within the transaction.

Class Hierarchy: Item>UseCase

SubClasses of UseCase

This class has no subclasses.



Properties Specific to UseCase

Properties	Inherited From	Description
Abstract		True if the abstract check box is checked.
Documentation	Item	Documentation for the item.
HasStateDiagram		True if the use case has an associated state diagram.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Rank		The rank of the UseCase.
RequisiteProDocName		The associated ReqPro ReqDocument name.

Properties	Inherited From	Description
RequisiteProProjectPath		The associated ReqPro Project path.
RequisiteProReqtGUID		The associated ReqPro Requirement GUID.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to UseCase

Name	Kind	Class	Description
Associations	0..n	Association	The associations where this UseCase plays a role.
ClassDiagrams	0..n	ClassDiagram	The class diagrams included in this UseCase.
MyRelationships	0..n	Relationship	The inherits and role relationships defined by this UseCase.
ParentPackage	0..1	Package	The enclosing package.
ParticipatingObjects	0..n	ObjectInstance	The objects included in scenarios defined by this UseCase.
Scenarios	0..n	Scenario	The scenarios by this UseCase.
StateDiagram	0..1	StateDiagram	The top-level state diagram associated with this UseCase.
StateDiagrams	0..n	StateDiagram	All state diagrams associated with this UseCase.
StateMachine	0..1	StateMachine	The top-level state machine associated with this UseCase.
StateMachines	0..n	StateMachine	All state machines associated with this UseCase.
SubUseCases		UseCase	The UseCases that inherit from this UseCase.
SuperUseCases		UseCase	The UseCases that this UseCase inherits from directly.

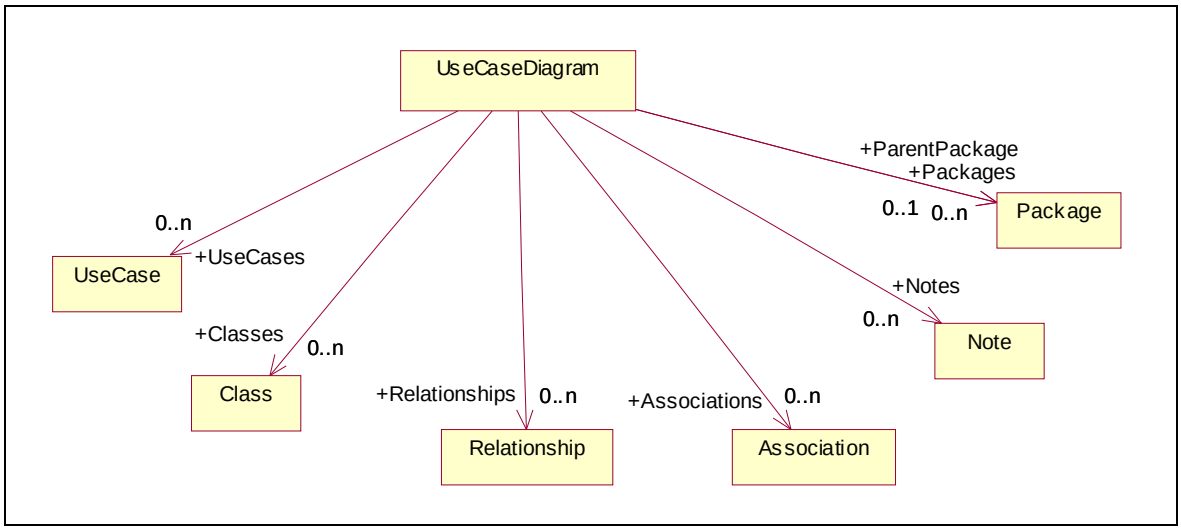
UseCaseDiagram
Rose Domain

A use case diagram shows the relationships between use cases and actors. Use case diagrams can be considered as filtered views into the model. They do not necessarily depict all the use cases or relationships in the model. For example, iterating over all the use cases in the main diagram of a package will not necessarily return all the use cases defined in that package.

Class Hierarchy: Diagram>UseCaseDiagram

SubClasses of UseCaseDiagram

Package



Properties Specific to UseCaseDiagram

Properties	Inherited From	Description
Documentation	Diagram	The documentation text associated with the Diagram.
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.
Name	Diagram	Name of the Diagram.
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to UseCaseDiagram

Name	Kind	Class	Description
Associations	0..n	Association	The associations where this use case diagram plays a role.
Classes	0..n	Class	All of the classes that appear on the diagram.
Notes	0..n	Note	All of the notes associated with the diagram.
Packages	0..n	Package	All of the packages that appear on the diagram.
ParentPackage	0..1	Package	The package that contains the diagram, if applicable.
Relationships	0..n	Relationship	All of the relationships that appear on the diagram.
UseCases	0..n	UseCase	All of the use cases that appear on the diagram.

UsesRelationship
Rose Domain

Indicates that the client class depends on the supplier class to provide certain services, such as:
The client class accesses a value (constant or variable) defined in the supplier class.
Operations of the client class invoke operations of the supplier class.
Operations of the client class have signatures whose return class or arguments are instances of the supplier class.

Class Hierarchy: Item>Relationship>UsesRelationship

SubClasses of UsesRelationship

This class has no subclasses.

Properties Specific to UsesRelationship

Properties	Inherited From	Description
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Documentation	Item	Documentation for the item.
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
InvolvesFriendship		Indicates whether the supplier class grants rights to the client class to access its non-public parts. Returns TRUE, if the Friendship required check box is checked on the relationship specification. Otherwise, returns FALSE.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.

Properties	Inherited From	Description
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to UsesRelationship

This class has no relationships.