

Rose RealTime Domain

Introduction

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

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

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

The Rose RealTime domain uses aliases to support multiple notations. SoDA is delivered with UML aliases. To use aliases for another notation, or another language, modify the Rose RT.dom file.

Model

A Rose RealTime model file. A model file contains a Rose RealTime 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 RealTime source domain. Most templates will start with connections to a Model.

Properties Specific to Model

Properties	Description
Documentation	
DriveLetter	
Extension	
FileName	
FullName	
Name	
NameMinusExtension	
NamePrefix	
ParentDirectoryPath	
SimpleName	
Stereotype	
UniqueID	

Relationships Specific to Model

Name	Kind	Class	Description
AllAssociations	0..n	Association	All associations in the model.
AllCapsules	0..n	Capsule	
AllClasses	0..n	Class	All classes in the model, including actors.
AllClassesRecursively	0..n	Class	
AllComponentPackages	0..n	ComponentPackage	
AllComponents	0..n	Component	All components in the model (including subsystems).
AllPackages	0..n	Package	All packages in the model, including use-case packages (but not including subsystems in the Component View).
AllProperties	0..n	Property	

Name	Kind	Class	Description
AllProtocols	0..n	Protocol	
AllRelationships	0..n	Relationship	All relationships in the model.
AllUseCases	0..n	UseCase	All use cases in the model.
ComponentView	0..1	ComponentPackage	The highest-level subsystem in the model; its name is Component View. All other subsystems are nested beneath it.
DeploymentDiagram	0..1	DeploymentDiagram	The deployment diagram (process diagram) for the model.
Deployment View	0..1	DeploymentPackage	
ExternalDocuments	0..n	ExternalDocument	
LogicalView	0..1	Package	The highest-level package in the model; its name is Logical View. All other packages are nested beneath it.
Model	0..1	Model	
Properties	0..n	Property	The code-generation properties associated with the model.
UseCaseView	0..1	Package	The root use-case package in the model; its name is Use Case View. All other use-case packages are nested beneath it.

Element

Every object in a Rose RealTime model (including the model itself) is an element. And every element in a Rose RealTime model has a name and /or a unique ID. Following this logic, you can use Element Class methods to obtain the ID for any item in the current model, and from there get or set its properties and property sets.

The unique element ID also provides the most direct means of accessing an item from a collection. While you can still use GetFirst and GetNext methods to iterate through a collection, you can also use the GetwithUniqueID method to obtain the item right away, without searching through the collection.

SubClasses of Element

ModelElement, Diagram, StateMachine, Trigger

Properties Specific to Element

Properties	Description
ClassName	Class name of the Element.
Name	Name of the Element.
UniqueID	Unique identifier of the Element.

Relationships Specific to Element

Name	Kind	Class	Description
AllProperties	0..n	Property	Returns the collection of properties belonging to the specified element.
Model	0..1	Model	
Properties	0..n	Property	The code-generation properties associated with the model.

Diagram

The Diagram class exposes a set of attributes and operations, which all other diagram classes (for example, class diagrams, sequence diagrams, Collaboration diagrams, etc.) inherit. These attributes and operations determine the size and placement of a diagram on the Rose RealTime user's computer screen.

Diagram is a subclass of Element.

SubClasses of Diagram

ClassDiagram, Sequence Diagram, DeploymentDiagram, CollaborationDiagram, StateDiagram, ComponentDiagram, UseCaseDiagram

Properties Specific to Diagram

Properties	Description
Diagram graphic	
Documentation	

Relationships Specific to Diagram

Name	Kind	Class	Description
ExternalDocuments	0..n	ExternalDocument	
ModelElements	0..n	ModelElement	Specifies the collection of ModelElements belonging to the diagram.
Notes	0..n	NoteView	
ParentModelElement	0..1	ModelElement	

ClassDiagram

The class diagram class allows you to add, retrieve and delete classes and categories to and from a class diagram. The class diagram class has a set of attributes and operations that apply specifically to class diagrams. In addition, it inherits all diagram class attributes and operations.

ClassDiagram is a subclass of Diagram.

SubClasses of ClassDiagram

This class has no subclasses.

Properties Specific to ClassDiagram

This class has no properties.

Relationships Specific to ClassDiagram

Name	Kind	Class	Description
Capsules	0..n	Capsule	
Classes	0..n	Class	All of the classes that appear on the diagram.
Packages	0..n	Package	
ParentPackage	0..1	Package	Specifies the LogicalPackage that contains the class diagram.
Protocols	0..n	Protocol	
UseCases	0..n	UseCase	All of the use cases that appear on the diagram.

CollaborationDiagram

CollaborationDiagram is a subclass of Diagram.

SubClasses of CollaborationDiagram

This class has no subclasses.

Properties Specific to CollaborationDiagram

This class has no subclasses.

Relationships Specific to CollaborationDiagram

This class has no subclasses.

ComponentDiagram

A component diagram shows relationships between subsystems and components. Each component diagram provides a physical view of the current model. Each component diagram is contained by the subsystem enclosing the components it depicts.

ComponentDiagram is a subclass of Diagram.

SubClasses of ComponentDiagram

This class has no subclasses.

Properties Specific to ComponentDiagram

This class has no properties.

Relationships Specific to ComponentDiagram

Name	Kind	Class	Description
Components	0..n	Component	The components contained in the diagram.

DeploymentDiagram

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.

DeploymentDiagram is a subclass of Diagram.

SubClasses of DeploymentDiagram

This class has no subclasses.

Properties Specific to DeploymentDiagram

This class has no properties.

Relationships Specific to DeploymentDiagram

Name	Kind	Class	Description
Processors	0..n	Processor	The processors contained in the diagram.
Devices	0..n	Device	The devices contained in the diagram.

SequenceDiagram

SequenceDiagram is a subclass of Diagram.

SubClasses of SequenceDiagram

This class has no subclasses.

Properties Specific to SequenceDiagram

This class has no properties.

Relationships Specific to SequenceDiagram

This class has no relationships.

StateDiagram

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

StateDiagram is a subclass of Diagram.

SubClasses of StateDiagram

This class has no subclasses.

Properties Specific to StateDiagram

This class has no properties.

Relationships Specific to StateDiagram

This class has no relationships.

UseCaseDiagram

UseCaseDiagram is a subclass of Diagram.

SubClasses of UseCaseDiagram

This class has no subclasses.

Properties Specific to UseCaseDiagram

This class has no properties.

Relationships Specific to UseCaseDiagram

Name	Kind	Class	Description
Capsules	0..n	Capsule	All of the capsules that appear on the diagram
Classes	0..n	Class	All of the classes that appear on the diagram
Packages	0..n	Package	All of the packages that appear on the diagram
ParentPackage	0..1	Package	Package that contains the diagram, if applicable
Protocols	0..n	Protocol	All of the protocols that appear on the diagram
UseCases	0..n	UseCase	All of the use cases that appear on the diagram

StateMachine

Defines event-ordered behavior of a class.

StateMachine is a subclass of Element.

SubClasses of StateMachine

This class has no subclasses.

Properties Specific to StateMachine

This class has no properties.

Relationships Specific to StateMachine

Name	Kind	Class	Description
AllStates	0..n	StateVertex	All states that are part of this state machine
ParentClassifier	0..1	Classifier	
StateDiagram	0..1	StateDiagram	The (first) state diagram associated with this state machine.
Top	0..1	CompositeState	

Trigger

Trigger is a subclass of Element.

SubClasses of Trigger

This class has no subclasses.

Properties Specific to Trigger

Properties	Description
Guard	

Relationships Specific to Trigger

Name	Kind	Class	Description
ParentTransition	0..1	Transition	
Ports	0..n	Port	
Signals	0..n	Signal	

ModelElement

ModelElement is a subclass of Element.

SubClasses of ModelElement

Action, Association, Attribute, Classifier, ClassifierRole, Collaboration, Component, ComponentPackage, ComponentInstance, Connector, DeploymentPackage, Device, Interaction, InteractionInstance, Message, Operation, Package, Parameter, PortRole, Processor, Relationship, Signal, StateVertex, Transition

Properties Specific to ModelElement

Properties	Description
Documentation	
Stereotype	

Relationships Specific to ModelElement

Name	Kind	Class	Description
ExternalDocuments	0..n	ExternalDocument	

Action

Actions are the things the behavior does when a transition is taken. They represent executable atomic computations that are written as statements in a detail-level programming language and incorporated into a state machine. Actions are atomic, in the sense that they cannot be interrupted by the arrival of a higher priority event. An action therefore runs to completion.

Action is a subclass of ModelElement.

SubClasses of Action

LocalState, RequestAction, ResponseAction, Coregion, CreateAction, DestroyAction, TerminateAction, UninterpretedAction

Properties Specific to Action

Properties	Description
Kind	
Time	Capture the time of the state change.

Relationships Specific to Action

Name	Kind	Class	Description
Arguments	0..n	String	Name of arguments passed to the action.
ParentMessage	0..1	Message	Message owning the Action. Nothing if the Action is owned by a State or a Transition.
ParentState	0..1	State	State owning the Action. Nothing if the Action is owned by a Message or a Transition.
ParentTransition	0..1	Transition	Transition owning the Action. Nothing if the Action is owned by a Message or a State.

LocalState

LocalState is a subclass of Action.

SubClasses of LocalState

This class has no subclasses.

Properties Specific to LocalState

This class has no properties.

Relationships Specific to LocalState

This class has no relationships.

RequestAction

RequestAction is a subclass of Action.

SubClasses of RequestAction

CallAction, SendAction

Properties Specific to RequestAction

Properties	Description
Mode	

Relationships Specific to RequestAction

Name	Kind	Class	Description
Return	0..1	ResponseAction	

CallAction

CallAction is a subclass of RequestAction.

SubClasses of CallAction

This class has no subclasses.

Properties Specific to CallAction

Properties	Description
Operation	

Relationships Specific to CallAction

This class has no relationships.

SendAction

SendAction is a subclass of RequestAction.

SubClasses of SendAction

This class has no subclasses.

Properties Specific to SendAction

Properties	Description
DeliveryTime	
Priority	
ReceiverPort	
SenderPort	
Signal	

Relationships Specific to SendAction

This class has no relationships.

ResponseAction

ResponseAction is a subclass of Action.

SubClasses of ResponseAction

ReturnAction, ReplyAction

Properties Specific to ResponseAction

This class has no properties.

Relationships Specific to ResponseAction

Name	Kind	Class	Description
Request	0..1	RequestAction	

ReturnAction

ReturnAction is a subclass of ResponseAction.

SubClasses of ReturnAction

This class has no subclasses.

Properties Specific to ReturnAction

This class has no properties.

Relationships Specific to ReturnAction

This class has no relationships.

ReplyAction

ReplyAction is a subclass of ResponseAction.

SubClasses of ReplyAction

This class has no subclasses.

Properties Specific to ReplyAction

Properties	Description
Data	
Signal	

Relationships Specific to ReplyAction

This class has no relationships.

Coregion

Coregion is a subclass of Action.

SubClasses of Coregion

This class has no subclasses.

Properties Specific to Coregion

This class has no properties.

Relationships Specific to Coregion

Name	Kind	Class	Description
Messages	0..n	Message	

CreateAction

CreateAction is a subclass of Action.

SubClasses of CreateAction

This class has no subclasses.

Properties Specific to CreateAction

Properties	Description
Operation	

Relationships Specific to CreateAction

This class has no relationships.

DestroyAction

DestroyAction is a subclass of Action.

SubClasses of DestroyAction

This class has no subclasses.

Properties Specific to DestroyAction

This class has no properties.

Relationships Specific to DestroyAction

This class has no relationships.

TerminateAction

TerminateAction is a subclass of Action.

SubClasses of TerminateAction

This class has no subclasses.

Properties Specific to TerminateAction

This class has no properties.

Relationships Specific to TerminateAction

This class has no relationships.

UninterpretedAction

UninterpretedAction is a subclass of Action.

SubClasses of UninterpretedAction

This class has no subclasses.

Properties Specific to UninterpretedAction

Properties	Description
Code	
Effect	

Relationships Specific to UninterpretedAction

This class has no relationships.

Association

An association is a connection, or a link, between classes. The association class exposes a set of attributes and operations that:

- Determine the characteristics of associations between classes
- Allow you to retrieve associations from a model

Check the lists of attributes and operations for complete information.

Association is a subclass of ModelElement.

SubClasses of Association

AssociationRole

Properties Specific to Association

Properties	Description
IsDerived	True if the association is derived; otherwise False.

Relationships Specific to Association

Name	Kind	Class	Description
AssociationClass	0..1	Class	Class holding attributes and operations of an Association Class. May point to nothing if the Association is not an Association Class.
EndA	0..1	AssociationEnd	The first role defined in the association.
EndB	0..1	AssociationEnd	The second role defined in the association.

AssociationRole

AssociationRole is a subclass of Association.

SubClasses of AssociationRole

This class has no subclasses.

Properties Specific to AssociationRole

Properties	Description
BaseName	
Multiplicity	

Relationships Specific to AssociationRole

Name	Kind	Class	Description
Base	0..1	Association	
EndRoleA	0..1	AssociationEndRole	The first role defined in the association.
EndRoleB	0..1	AssociationEndRole	The second role defined in the association.
ParentCollaboration	0..1	Collaboration	

AssociationEndRole

AssociationEndRole is a subclass of AssociationEnd.

SubClasses of AssociationEndRole

This class has no subclasses.

Properties Specific to AssociationEndRole

Properties	Description
Multiplicity	

Relationships Specific to AssociationEndRole

Name	Kind	Class	Description
AssociationRole	0..1	AssociationRole	
Base	0..1	AssociationEnd	

Attribute

Attributes are data members of a class whose type is not another class.

Attribute is a subclass of ModelElement.

SubClasses of Attribute

This class has no subclasses.

Properties Specific to Attribute

Properties	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.
InitialValue	The initial value of the attribute.
Derived	True if the Derived check box is selected in the attribute specification, otherwise False.
Scope	
Type	The type of the attribute.
Visibility	

Relationships Specific to Attribute

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

Classifier

A classifier is a base class that describes behavioral and structural features (attributes and operations). Classifier Class serves 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 RealTime 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.

Classifier is a subclass of ModelElement.

SubClasses of Classifier

Capsule, Class, Protocol, UseCase

Properties Specific to Classifier

Properties	Description
IsAbstract	True if the classifier is an abstract classifier.
HasStateDiagram	
IsSystemClass	
Language	
QualifiedName	
Visibility	

Relationships Specific to Classifier

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.
AllCollaborations	0..n	Collaboration	
AllOperations	0..n	Operation	All operations of this class, including those inherited from other classes.

Name	Kind	Class	Description
AllRelationships	0..n	Relationship	All relationships of this class, including those inherited from other classes.
AllSubClasses	0..n	Classifier	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	0..n	Classifier	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.
Associations	0..n	Association	The associations where this class plays a role.
Attributes	0..n	Attribute	Causes the classifier to inherit all of the attributes of a specified attribute collection.
Collaborations	0..n	Collaboration	Collaborations that belong to this classifier.
Instances	0..n	InteractionInstance	The instances of this class.
Operations	0..n	Operation	Causes the classifier to inherit all of the operations of a specified operation collection.
ParentPackage	0..1	Package	The enclosing package.
RealizeRelationships	0..n	RealizeRelationship	
Relationships	0..n	Relationship	
StateDiagram	0..1	StateDiagram	
StateMachine	0..1	StateMachine	Specifies the state machine that belongs to the classifier. A state machine defines all of the state information, including states, transitions, and state diagrams, defined for a given classifier. A classifier can have zero or one state machine.
SubClasses	0..n	Classifier	
SuperClasses	0..n	Classifier	

Capsule

Capsule is a subclass of Classifier.

SubClasses of Capsule

This class has no subclasses.

Properties Specific to Capsule

This class has no properties.

Relationships Specific to Capsule

Name	Kind	Class	Description
CapsuleRoles	0..n	CapsuleRole	
Connectors	0..n	Connector	
Diagram	0..1	CollaborationDiagram	
Ports	0..n	Port	
Structure	0..1	CapsuleStructure	

Class

The Class class allows you to get and set the characteristics and relationships of specific classes in a model.

Some of the questions answered by class properties are:

- Is this an abstract class?
- Is this class a fundamental type?
- Is this class persistent?
- Can this class be concurrent with any other classes?
- What set of attributes and operations belong to this class?
- What relationships are defined between this class and other objects in the model?

Class operations allow you to get and set this information for the classes in the model.

Class is a subclass of Classifier.

SubClasses of Class

ParameterizedClass, InstantiatedClass, ClassUtility, ParameterizedClassUtility, InstantiatedClassUtility, MetaClass

Properties Specific to Class

Properties	Description
Concurrency	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
IsFundamentalType	True if this class is a fundamental type.
IsNestedClass	True if the class is nested.
Multiplicity	Multiplicity of the Class.
Persistence	Returns Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
Type	The ClassKind property is a rich data type that determines the type of the class.
Space	The string in the Space field of the More dialog of the class specification.

Relationships Specific to Class

Name	Kind	Class	Description
AllNestedClasses		Class	
AppearsIn	0..n	ClassDiagram	The class diagrams where this class appears.
InstancesAppearIn	0..n	Interaction	The interaction diagrams that include instances of this class.
InstantiateRelationships	0..n	InstantiateRelationship	Returns the collection of Instantiate Relations that belong to a class.
NestedClasses	0..n	Class	The classes that are nested within this class.
ParentClass	0..1	Class	The parent class of this class, if it is nested.

ParameterizedClass

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.

ParameterizedClass is a subclass of Class.

SubClasses of ParameterizedClass

This class has no subclasses.

Properties Specific to ParameterizedClass

This class has no properties.

Relationships Specific to ParameterizedClass

Name	Kind	Class	Description
FormalArguments	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.

InstantiatedClass

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.

InstantiatedClass is a subclass of Class.

SubClasses of InstantiatedClass

This class has no subclasses.

Properties Specific to InstantiatedClass

This class has no properties.

Relationships Specific to InstantiatedClass

This class has no relationships.

ClassUtility

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.

ClassUtility is a subclass of Class.

SubClasses of ClassUtility

This class has no subclasses.

Properties Specific to ClassUtility

This class has no properties.

Relationships Specific to ClassUtility

This class has no relationships.

ParameterizedClassUtility

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.

ParameterizedClassUtility is a subclass of Class.

SubClasses of ParameterizedClassUtility

This class has no subclasses.

Properties Specific to ParameterizedClassUtility

This class has no properties.

Relationships Specific to ParameterizedClassUtility

Name	Kind	Class	Description
FormalArguments	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.

InstantiatedClassUtility

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.

InstantiatedClassUtility is a subclass of Class.

SubClasses of InstantiatedClassUtility

This class has no subclasses.

Properties Specific to InstantiatedClassUtility

This class has no properties.

Relationships Specific to InstantiatedClassUtility

This class has no relationships.

MetaClass

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 will be required by all objects of a class. Smalltalk and CLOS support the use of metaclasses. C++ does not directly support metaclasses.

MetaClass is a subclass of Class.

SubClasses of MetaClass

This class has no subclasses.

Properties Specific to MetaClass

This class has no properties.

Relationships Specific to MetaClass

This class has no relationships.

Protocol

Protocol is a subclass of Classifier.

SubClasses of Protocol

This class has no subclasses.

Properties Specific to Protocol

This class has no properties.

Relationships Specific to Protocol

Name	Kind	Class	Description
InSignals	0..n	Signal	
Interactions	0..n	Interaction	
OutSignals	0..n	Signal	

UseCase

A use case is 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.

UseCase is a subclass of Classifier.

SubClasses of UseCase

This class has no subclasses.

Properties Specific to UseCase

Properties	Description
Rank	The rank of the use case.
RequisiteProDocName	
RequisiteProProjectPath	
RequisiteProReqtGUID	

Relationships Specific to UseCase

Name	Kind	Class	Description
ClassDiagrams	0..n	ClassDiagram	The class diagrams included in this use case.
SuperUseCases	0..n	UseCase	The use cases that this use case inherits from directly.
UseCaseDiagrams	0..n	UseCaseDiagram	The use-case diagrams associated with this use case.

ClassifierRole

A classifier role is a specific role played by a participant in collaboration. It specifies a restricted view of a classifier, defined by what is required in the collaboration.

ClassifierRole is a subclass of ModelElement.

SubClasses of ClassifierRole

CapsuleRole

Properties Specific to ClassifierRole

Properties	Description
ClassifierName	Name of the Classifier the ClassifierRole is a projection of.
Multiplicity	The number of Classifier playing this role in a Collaboration.

Relationships Specific to ClassifierRole

Name	Kind	Class	Description
Classifier	0..1	Classifier	Classifier the ClassifierRole is a projection of.
ParentCollaboration	0..1	Collaboration	Collaboration that owns the ClassifierRole.

CapsuleRole

CapsuleRole is a subclass of ClassifierRole.

SubClasses of CapsuleRole

This class has no subclasses.

Properties Specific to CapsuleRole

Properties	Description
Cardinality	
Genericity	
IsSubstitutable	

Relationships Specific to CapsuleRole

Name	Kind	Class	Description
Capsule	0..1	Capsule	
PortRoles	0..n	PortRole	

Port

Port is a subclass of ClassifierRole.

SubClasses of Port

This class has no subclasses.

Properties Specific to Port

Properties	Description
Cardinality	
IsConjugated	
IsEndPort	
IsNotified	
IsPublished	
IsWired	
RegistrationMode	
RegistrationString	
Visibility	

Relationships Specific to Port

Name	Kind	Class	Description
Protocol	0..1	Protocol	

Collaboration

Collaboration is a subclass of ModelElement.

SubClasses of Collaboration

This class has no subclasses.

Properties Specific to Collaboration

This class has no properties.

Relationships Specific to Collaboration

Name	Kind	Class	Description
AssociationRoles	0..n	AssociationRole	
ClassifierRoles	0..n	ClassifierRole	
Connectors	0..n	Connector	
Diagram	0..1	CollaborationDiagram	
Interactions	0..n	Interaction	
ParentClassifier	0..1	Classifier	
ParentLogicalPackage	0..1	Package	

CapsuleStructure

CapsuleStructure is a subclass of Collaboration.

SubClasses of CapsuleStructure

This class has no subclasses.

Properties Specific to CapsuleStructure

This class has no properties.

Relationships Specific to CapsuleStructure

Name	Kind	Class	Description
CapsuleRoles	0..n	CapsuleRole	
Ports	0..n	Port	

Component

A building block for the physical structure of a system. A component can be one of the following: Main Program, Package Body, Subprogram, Package, Task Body, Generic Package, Task, Subprogram Body.

Component is a subclass of ModelElement.

SubClasses of Component

This class has no subclasses.

Properties Specific to Component

Properties	Description
Environment	
Type	

Relationships Specific to Component

Name	Kind	Class	Description
ClassifierReferences	0..n	Classifier	
PackageReferences	0..n	Package	
ParentComponentPackage	0..1	ComponentPackage	
Relationship	0..n	Relationship	

ComponentPackage

ComponentPackage is a subclass of ModelElement.

SubClasses of ComponentPackage

This class has no subclasses.

Properties Specific to ComponentPackage

Properties	Description
IsRootPackage	

Relationships Specific to ComponentPackage

Name	Kind	Class	Description
AllComponents	0..n	Component	
AllComponentPackages	0..n	ComponentPackage	
ComponentDependencies	0..n	ComponentDependency	
Components	0..n	Component	
ComponentDiagrams	0..n	ComponentDiagram	
ComponentPackages	0..n	ComponentPackage	
ParentComponentPackage	0..1	ComponentPackage	
VisibleComponentPackages	0..n	ComponentPackage	

ComponentInstance

ComponentInstance is a subclass of ModelElement.

SubClasses of ComponentInstance

This class has no subclasses.

Properties Specific to ComponentInstance

Properties	Description
AttachTargetObservability	
ConsolePort	
LoadDelay	
LoadOrder	
LogsPort	
OperationMode	
TargetObservabilityPort	
UserParameters	

Relationships Specific to ComponentInstance

Name	Kind	Class	Description
Component	0..1	Component	
Processor	0..1	Processor	

Connector

Connector is a subclass of ModelElement.

SubClasses of Connector

This class has no subclasses.

Properties Specific to Connector

Properties	Description
Cardinality	
Delay	

Relationships Specific to Connector

Name	Kind	Class	Description
Port1	0..1	Port	
Port2	0..1	Port	
PortRole1	0..1	PortRole	
PortRole2	0..1	PortRole	

DeploymentPackage

DeploymentPackage is a subclass of ModelElement.

SubClasses of DeploymentPackage

This class has no subclasses.

Properties Specific to DeploymentPackage

This class has no properties.

Relationships Specific to DeploymentPackage

Name	Kind	Class	Description
AllDevices	0..n	Device	
AllProcessors	0..n	Processor	
DeploymentDiagrams	0..n	DeploymentDiagram	
DeploymentPackages	0..n	DeploymentPackage	
ParentDeploymentPackage	0..1	DeploymentPackage	

Device

A device is a hardware component with no computing power.

Device is a subclass of ModelElement.

SubClasses of Device

This class has no subclasses.

Properties Specific to Device

Properties	Description
Characteristics	

Relationships Specific to Device

Name	Kind	Class	Description
ConnectedDevices	0..n	Device	
ConnectedProcessors	0..n	Processor	

Interaction

Interaction is a subclass of ModelElement.

SubClasses of Interaction

This class has no subclasses.

Properties Specific to Interaction

This class has no properties.

Relationships Specific to Interaction

Name	Kind	Class	Description
Instances	0..n	InteractionInstance	
Messages	0..n	Message	
ParentCollaboration	0..1	Collaboration	
ParentProtocol	0..1	Protocol	
SequenceDiagram	0..1	SequenceDiagram	

InteractionInstance

InteractionInstance is a subclass of ModelElement.

SubClasses of InteractionInstance

Environment

Properties Specific to InteractionInstance

This class has no properties.

Relationships Specific to InteractionInstance

Name	Kind	Class	Description
Messages	0..n	Message	
Path	0..n	ClassifierRole	
ParentInteraction	0..1	Interaction	

Environment

Environment is a subclass of InteractionInstance.

SubClasses of Environment

This class has no subclasses.

Properties Specific to Environment

This class has no properties.

Relationships Specific to Environment

This class has no relationships.

Message

Any message associated with an object.

Message is a subclass of ModelElement.

SubClasses of Message

This class has no subclasses.

Properties Specific to Message

This class has no properties.

Relationships Specific to Message

Name	Kind	Class	Description
Action	0..1	Action	
Activator	0..1	Message	
ParentInteraction	0..1	Interaction	
Receiver	0..1	InteractionInstance	The object that receives the message.
Sender	0..1	InteractionInstance	The object that sends the message.

Operation

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 RealTime stores operation information in an operation specification. You can access operation specifications only through the class specification.

Operation is a subclass of ModelElement.

SubClasses of Operation

This class has no subclasses.

Properties Specific to Operation

Properties	Description
AdalImage	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 AdalImage 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.
CplusplusImage	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 CplusplusImage 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.
Code	
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.
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.
IsAbstract	
IsQuery	
IsVirtual	
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	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.
ReturnType	For operations that are functions, ReturnType specifies the type of data returned by the function.
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.
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.
Visibility	

Relationships Specific to Operation

Name	Kind	Class	Description
Parameters	0..n	Parameter	
ParentClassifier	0..1	Classifier	

Package

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 RealTime 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.

Package is a subclass of ModelElement.

SubClasses of Package

This class has no subclasses.

Properties Specific to Package

Properties	Description
HasAssignedComponentPackage	True if the package has a subsystem associated with it, otherwise False.
IsGlobal	
IsRootPackage	
IsUseCasePackage	True if the package is a descendent of the Use Case View package, otherwise False.

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.
AllCapsules	0..n	Capsule	All capsules owned by this package and any of its subpackages.
AllClassDiagrams	0..n	ClassDiagram	Diagrams for all of the classes belonging to this package.
AllClasses	0..n	Class	All classes that are defined in this package, or in any nested packages.

Name	Kind	Class	Description
AllProtocols		Protocol	All protocols owned by this package and any of its subpackages.
AllUseCaseDiagrams	0..n	UseCaseDiagram	All diagrams associated with use cases owned by the package and its subpackages
AllUseCases	0..n	UseCase	All use cases that are defined in this package, or in any nested packages.
AssignedComponentPackage	0..1	ComponentPackage	
Associations	0..n	Association	
Capsules	0..n	Capsule	
ChildPackages	0..n	Package	The packages owned by 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.
Collaborations	0..n	Collaboration	
DependedOnBy	0..n	Package	All logical packages that depend on this logical package.
DependsOn	0..n	PackageDependency	All package dependencies owned by the package and any of its subpackages.
Generalizations	0..n	Generalization	The set of generalizations that this package is a client of.
Imports	0..n	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.
PackageDependencies	0..n	PackageDependency	
MainDiagram	0..1	ClassDiagram	The diagram specifically called "Main".
NestedSubPackages	0..n	Package	All packages that are descendents of this package.

Name	Kind	Class	Description
ParentPackage	0..1	Package	The enclosing package. This relationship will result in an error if applied to the TopLevelCategory.
Protocols	0..n	Protocol	
Referencers	0..n	Package	All packages that import this package. Does not include indirect referencers.
Relationships	0..n	Relationship	Relationships involving any class or use case within the package.
SubPackages	0..n	Package	The packages that are derived from this package.
SuperPackages	0..n	Package	The packages from which this package is derived.
UseCaseDiagrams	0..n	UseCaseDiagram	Diagrams associated with use cases owned by this package.
MyUseCases	0..n	UseCase	

Parameter

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

Parameter is a subclass of ModelElement.

SubClasses of Parameter

This class has no subclasses.

Properties Specific to Parameter

Properties	Description
InitValue	The initial value of the parameter
IsConst	True if the parameter is constant; otherwise False
Type	The type of the parameter.

Relationships Specific to Parameter

This class has no relationships.

PortRole

PortRole is a subclass of ModelElement.

SubClasses of PortRole

This class has no subclasses.

Properties Specific to PortRole

This class has no properties.

Relationships Specific to PortRole

Name	Kind	Class	Description
ParentCapsuleRole	0..1	CapsuleRole	
Port	0..1	Port	

Processor

A processor is a hardware component capable of executing programs.

Processor is a subclass of ModelElement.

SubClasses of Processor

This class has no subclasses.

Properties Specific to Processor

Properties	Description
Address	
CPU	
OS	
ServerAddress	
UserScriptDirectory	

Relationships Specific to Processor

Name	Kind	Class	Description
ComponentInstances	0..n	ComponentInstance	
ConnectedDevices	0..n	Device	
ConnectedProcessors	0..n	Processor	

Relationship

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

Relationship is a subclass of ModelElement.

SubClasses of Relationship

UsesRelationship, RealizeRelationship, InstantiateRelationship, Generalization, PackageDependency, ComponentDependency, ComponentAggregation, AssociationEnd

Properties Specific to Relationship

Properties	Description
Kind	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship or UsesRelationship.
ToName	

Relationships Specific to Relationship

Name	Kind	Class	Description
FromModelElement	0..1	ModelElement	The client class. For example, if A Has a B, A is the client, or FromModelElement class.
ToModelElement	0..1	ModelElement	The supplier class. For example, if A Has a B, B is the supplier, or ToModelElement class.

AssociationEnd

AssociationEnd is a subclass of Relationship.

SubClasses of AssociationEnd

This class has no subclasses.

Properties Specific to AssociationEnd

Properties	Description
Constraints	
Containment	
IsAggregate	
IsFriend	
IsNavigable	
IsStatic	
Multiplicity	
Visibility	

Relationships Specific to AssociationEnd

Name	Kind	Class	Description
Association	0..1	Association	
Classifier	0..1	Classifier	
FromClassifier	0..1	Classifier	
FromElement	0..1	ModelElement	
Keys	0..n	Attribute	
OtherAssociationEnd	0..1	AssociationEnd	
ToClassifier	0..1	Classifier	
UseCase	0..1	UseCase	

ComponentAggregation

ComponentAggregation is a subclass of Relationship.

SubClasses of ComponentAggregation

This class has no subclasses.

Properties Specific to ComponentAggregation

This class has no properties.

Relationships Specific to ComponentAggregation

Name	Kind	Class	Description
FromComponent	0..1	Component	
ToComponent	0..1	Component	

ComponentDependency

ComponentDependency is a subclass of Relationship.

SubClasses of ComponentDependency

This class has no subclasses.

Properties Specific to ComponentDependency

This class has no properties.

Relationships Specific to ComponentDependency

Name	Kind	Class	Description
FromComponent	0..1	Component	
FromClass	0..1	Class	
FromComponentPackage	0..1	ComponentPackage	
ToComponent		Component	
ToClass	0..1	Class	
ToComponentPackage	0..1	ComponentPackage	

Generalization

Generalization is a subclass of Relationship.

SubClasses of Generalization

This class has no subclasses.

Properties Specific to Generalization

Properties	Description
FriendshipRequired	
Visibility	

Relationships Specific to Generalization

Name	Kind	Class	Description
FromClassifier	0..1	Classifier	
ToClassifier	0..1	Classifier	

InstantiateRelationship

InstantiateRelationship is a subclass of Relationship.

SubClasses of InstantiateRelationship

This class has no subclasses.

Properties Specific to InstantiateRelationship

This class has no properties.

Relationships Specific to InstantiateRelationship

Name	Kind	Class	Description
FromClass	0..1	Class	
ToClass	0..1	Class	

PackageDependency

PackageDependency is a subclass of Relationship.

SubClasses of PackageDependency

This class has no subclasses.

Properties Specific to PackageDependency

This class has no properties.

Relationships Specific to PackageDependency

Name	Kind	Class	Description
FromPackage	0..1	Package	
ToPackage	0..1	Package	

RealizeRelationship

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

RealizeRelationship is a subclass of Relationship.

SubClasses of RealizeRelationship

This class has no subclasses.

Properties Specific to RealizeRelationship

This class has no properties.

Relationships Specific to RealizeRelationship

Name	Kind	Class	Description
FromCapsule	0..1	Class	
FromClass	0..1	Class	
FromProtocol	0..1	Class	
ToClass	0..1	Class	
ToUseCase	0..1	Class	

UsesRelationship

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

UsesRelationship is a subclass of Relationship.

SubClasses of UsesRelationship

This class has no subclasses.

Properties Specific to UsesRelationship

Properties	Description
FromCardinality	
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.
ToCardinality	

Relationships Specific to UsesRelationship

Name	Kind	Class	Description
FromClassifier	0..1	Classifier	
ToClassifier	0..1	Classifier	

Signal

Signal is a subclass of ModelElement.

SubClasses of Signal

This class has no subclasses.

Properties Specific to Signal

Properties	Description
DataClassName	
IsInSignal	

Relationships Specific to Signal

Name	Kind	Class	Description
DataClass	0..1	Class	
ParentProtocol	0..1	Protocol	

StateVertex

StateVertex is a subclass of ModelElement.

SubClasses of StateVertex

State, Initial Point, JunctionPoint, ChoicePoint, FinalState

Properties Specific to StateVertex

Properties	Description
FullName	
StateKind	

Relationships Specific to StateVertex

Name	Kind	Class	Description
IncomingTransitions	0..n	Transition	
OutgoingTransitions	0..n	Transition	
ParentState	0..1	CompositeState	
ParentStateMachine	0..1	StateMachine	

State

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.

State is a subclass of StateVertex.

SubClasses of State

This class has no subclasses.

Properties Specific to State

This class has no properties.

Relationships Specific to State

Name	Kind	Class	Description
EntryAction	0..1	UninterpretedAction	
ExitAction	0..1	UninterpretedAction	
States	0..n	StateVertex	
SubDiagram	0..1	StateDiagram	The sub-diagram associated with a CompositeState (alias State)
Transitions	0..n	Transition	The transitions that exit from this state.

InitialPoint

InitialPoint is a subclass of StateVertex.

SubClasses of InitialPoint

This class has no subclasses.

Properties Specific to InitialPoint

This class has no properties.

Relationships Specific to InitialPoint

This class has no relationships.

JunctionPoint

JunctionPoint is a subclass of StateVertex.

SubClasses of JunctionPoint

This class has no subclasses.

Properties Specific to JunctionPoint

Properties	Description
Continuation	
IsEntry	
IsExit	
IsExternallyVisible	

Relationships Specific to JunctionPoint

This class has no relationships.

ChoicePoint

ChoicePoint is a subclass of StateVertex.

SubClasses of ChoicePoint

This class has no subclasses.

Properties Specific to ChoicePoint

Properties	Description
Condition	

Relationships Specific to ChoicePoint

Name	Kind	Class	Description
FALSETransition	0..1	Transition	
InTransition	0..1	Transition	
TRUETransition	0..1	Transition	

FinalState

FinalState is a subclass of StateVertex.

SubClasses of FinalState

This class has no subclasses.

Properties Specific to FinalState

This class has no properties.

Relationships Specific to FinalState

This class has no relationships.

Transition

Transition is a subclass of ModelElement.

SubClasses of Transition

This class has no subclasses.

Properties Specific to Transition

Properties	Description
IsInternal	
SourceRegion	

Relationships Specific to Transition

Name	Kind	Class	Description
Action	0..1	UninterpretedAction	
ParentState	0..1	CompositeState	
ParentStateMachine	0..1	StateMachine	
Source	0..1	StateVertex	
Target	0..1	StateVertex	
Triggers	0..n	Trigger	

Property

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

SubClasses of Property

This class has no subclasses.

Properties Specific to Property

Properties	Description
IsStructured	
Name	The name of the property.
PropertyType	
ToolName	The name of the tool, or tab, for the property, such as "cg" or "DDL".
Type	
Value	The string equivalent of the value associated with the property.

Relationships Specific to Property

This class has no relationships.

NoteView

SubClasses of NoteView

This class has no subclasses.

Properties Specific to NoteView

Properties	Description
Text	
Type	

Relationships Specific to NoteView

Name	Kind	Class	Description
ParentDiagram	0..1	Diagram	

File

SubClasses of File

This class has no subclasses.

Properties Specific to File

Properties	Description
IsURL	
Value	

Relationships Specific to File

Name	Kind	Class	Description
ParentLogicalPackage	0..1	Package	

String

The Rose RealTime String class is used to store the names of external documents.

SubClasses of String

This class has no subclasses.

Properties Specific to String

Properties	Description
Value	

Relationships Specific to String

This class has no relationships.