



Systems Modeling with Enterprise Architect version 7.1

Quick Start Guide

Version 1.0

February 12, 2009

Fundamentals of Systems Modeling Series

Pathfinder Solutions
www.pathfindersystemsmodeling.com
+1 508-568-0068

Table of Contents

OVERVIEW.....	2
HOW TO USE THIS GUIDE.....	2
AUDIENCE.....	2
CONVENTIONS.....	2
A NOTE FOR STUDENTS OF THE <i>FUNDAMENTALS OF SYSTEMS MODELING CLASS</i>	2
CREATING THE <i>SENTRYSYSTEM</i> MODEL	3
TASK 1: CREATE AN ENTERPRISE ARCHITECT SYSML PROJECT AND MODEL.....	3
<i>PROCEDURE: Use Enterprise Architect Start Page to Create a New Project</i>	3
TASK 2: CREATE USE CASE DIAGRAM.....	6
<i>PROCEDURE: Create a Use Case Diagram</i>	6
TASK 3: CREATE PACKAGES AND DIAGRAMS IN THE STRUCTURE VIEW.....	10
<i>PROCEDURE: Create Package Hierarchy in the Structure View</i>	10
<i>PROCEDURE: Complete the Block Definition Diagram capturing the Logical Architecture</i>	11
TASK 4: CREATE A LOGICAL SCENARIO MODEL.....	13
<i>PROCEDURE: Add a Sequence Diagram to Capture a new Logical Scenario</i>	13
TASK 5: CREATE NEW VALUE TYPES.....	18
<i>PROCEDURE: Add a new scalar value type and a structure block to the System DataModel</i>	18
<i>PROCEDURE: Add a new scalar value type to the GlobalGeometry Logical Component</i>	19
TASK 6: DETAILED BLOCKS IN A LOGICAL COMPONENT	20
<i>PROCEDURE: Create a BDD for the SensorResourceManagement Logical Component</i>	20
<i>PROCEDURE: Add Generalizations and Associations between SensorResourceManagement Blocks</i>	21
<i>PROCEDURE: Add Value Properties to SensorResourceManagement Blocks</i>	22
<i>PROCEDURE: Add Operations to SensorResourceManagement Blocks</i>	25
TASK 7: MODEL STATE BEHAVIOR FOR A LOGICAL BLOCK	26
<i>PROCEDURE: Create a State Machine Diagram for the Subject Block</i>	26
<i>PROCEDURE: Define Subject Block Signals</i>	27
<i>PROCEDURE: Add Signal Triggers to the Subject State Machine</i>	28
TRANSFORMING THE <i>SENTRYSYSTEM</i> MODEL.....	30
TASK 1: GENERATE HTML DOCUMENTATION FROM PROJECT.....	30
<i>PROCEDURE: Generate documentation for the SentrySystem project</i>	30

Overview

This document is provided as a first step in learning how to capture Systems Models using the Sparx Systems Enterprise Architect SysML tool.

How to Use this Guide

If you have not installed the Enterprise Architect toolset on your computer, please do this first.

Audience

The *Quick Start Guide* is for systems modelers who want to learn how use Enterprise Architect to capture systems models in SysML.

Conventions

The *Quick Start Guide* uses these conventions:

- **Bold** is for clickable buttons and menu selections.
- *Italics* is for screen text, path and file names, and other text that needs special emphasis.
- `Courier` denotes code, or text in a log or a batch file.
- A **Note** contains important information, or a timesaving tip.
- The scissors icon marks text that you copy from this document and paste elsewhere.



A Note for Students of the *Fundamentals of Systems Modeling Class*

Dear Students,

Given the time constraints of a class setting, this document has you construct a very limited subset of the SentrySystem model. Many elements are missing, or are only partially abstracted. This is especially true of model element descriptions – only a few are shown, and simply to introduce their mechanics. Please do not consider the abbreviated nature of the modeling procedures in this guide to be a suitable example of modeling practice.

Thank you for your efforts in applying your new skills, and best of luck!

Sincerely,

- *the instructors at Pathfinder Solutions*

Creating the *SentrySystem* Model

Task 1: Create an Enterprise Architect SysML Project and Model

In Enterprise Architect, all work is conducted within a project.

PROCEDURE: Use Enterprise Architect Start Page to Create a New Project

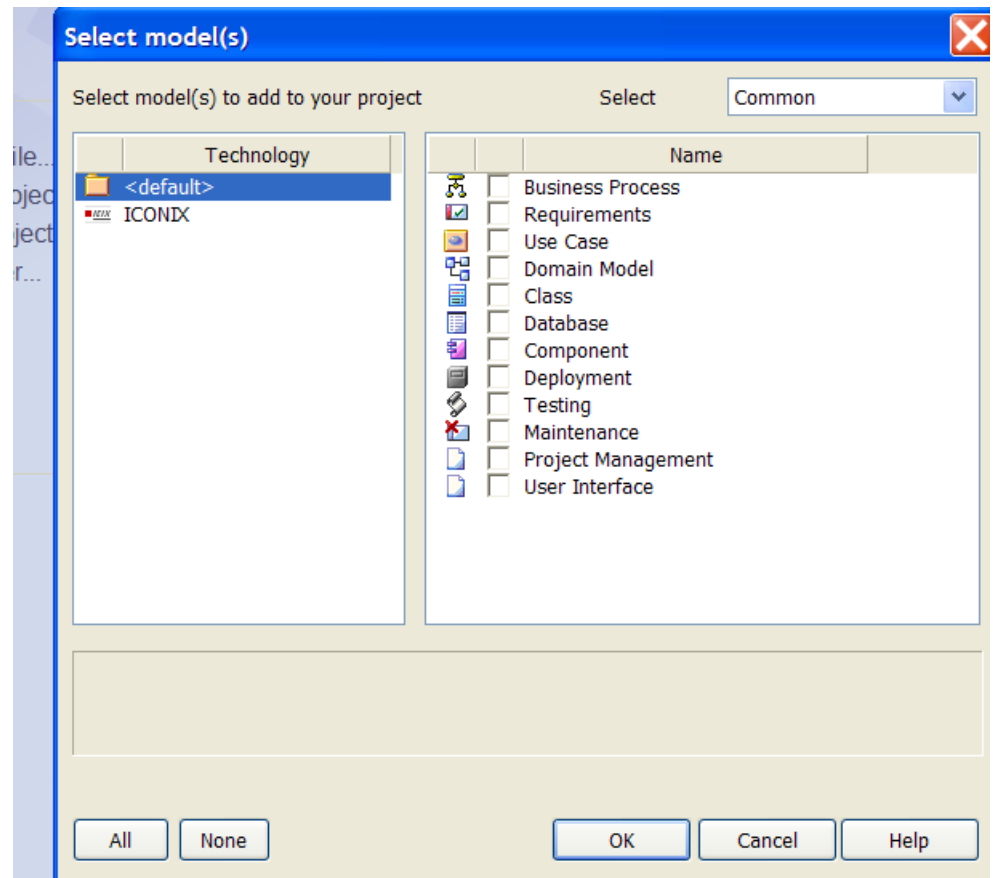
1. From the Enterprise Architect Start page click on **Create A New Project...**



The New Project dialog opens.

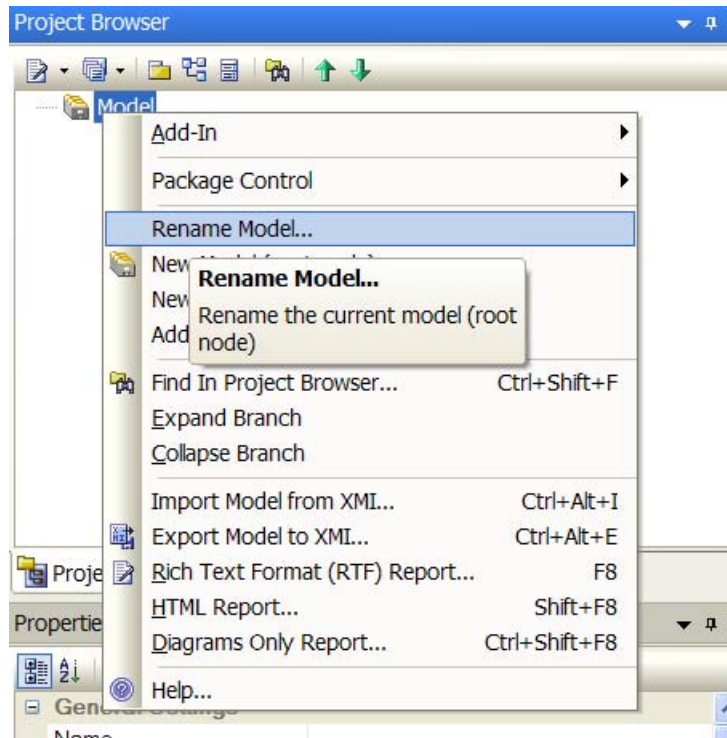
2. Select a directory and enter the filename *SentrySystem.eap*. Click **Save...**

The select model(s) dialog opens



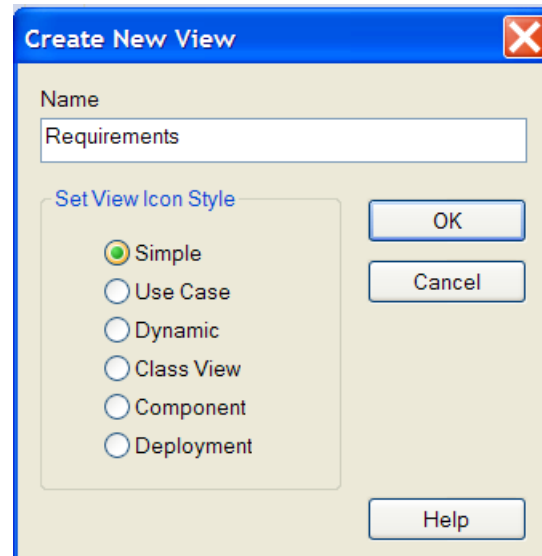
3. Click **None**, then click **OK**.

A new model is displayed in the Project Browser pane. Select the model and right click to Rename it to SentrySystem.



4. Select the model *SentrySystem*. Right click and select **New View...**

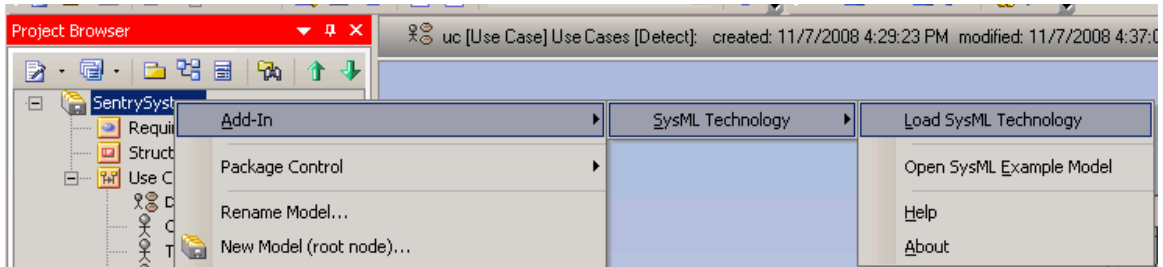
The Create New View dialog is displayed.



5. Enter *Requirements* into the Name text entry field. Select **Simple** for the SetViewIcon Style, and click **OK**.

A new view named *Requirements* is added to the *SentrySystem* model.

6. Add new views named Use Cases and Structure, with Set View Icon Styles **Use Case** and **Component** respectively.
7. Select *SentrySystem* model from the Project Browser. Right click and select **Add-In -> SysML Technology -> Load SysML Technology**.



8. Select *SentrySystem* model from the Project Browser. Right click and select **Add-In -> SysML Technology -> Open SysML Example Model**

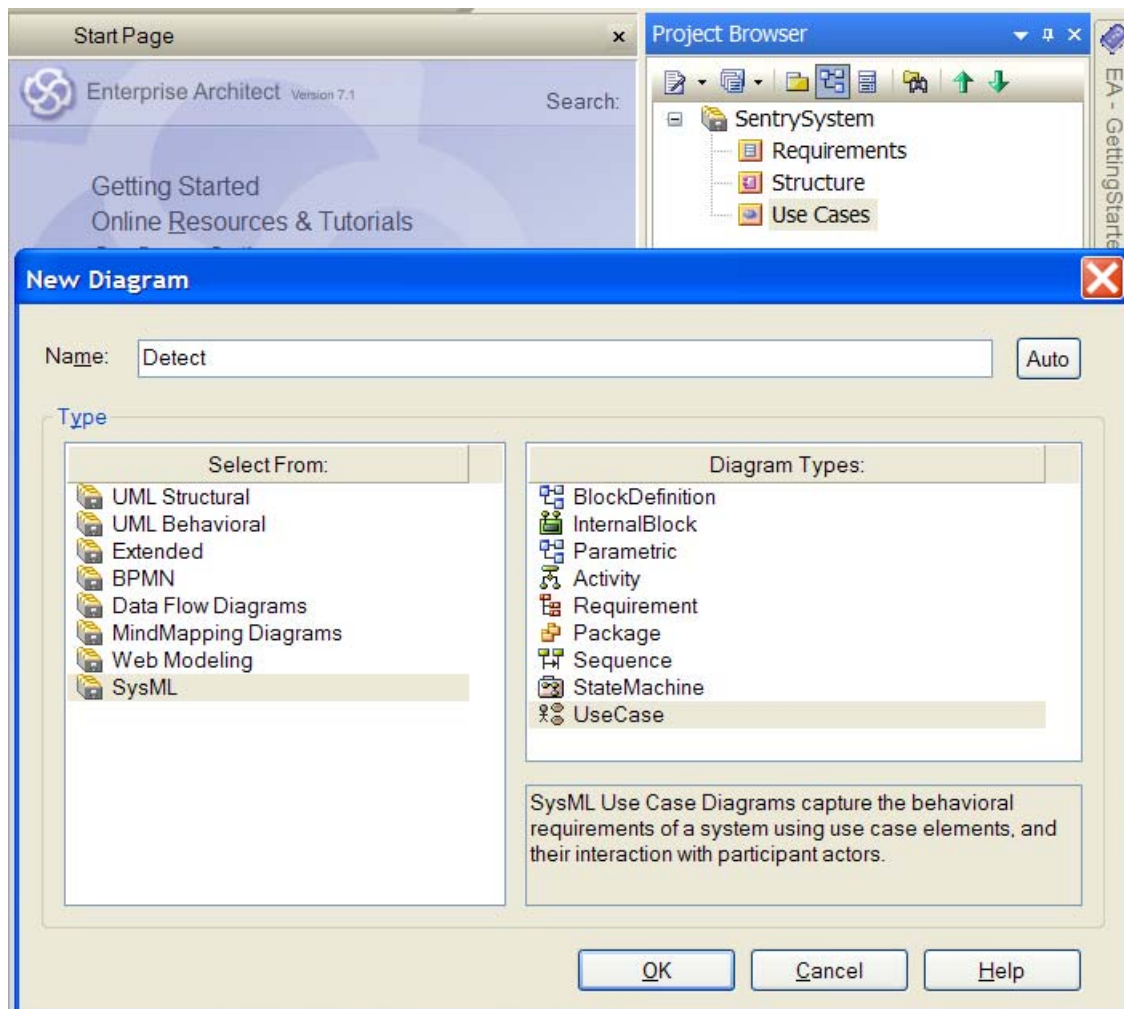
Task 2: Create Use Case Diagram

PROCEDURE: Create a Use Case Diagram

1. Open the *SentrySystem* model from the Project Browser. Select the Use Cases view in the ProjectBrowser. Click the **New Diagram** tool icon.



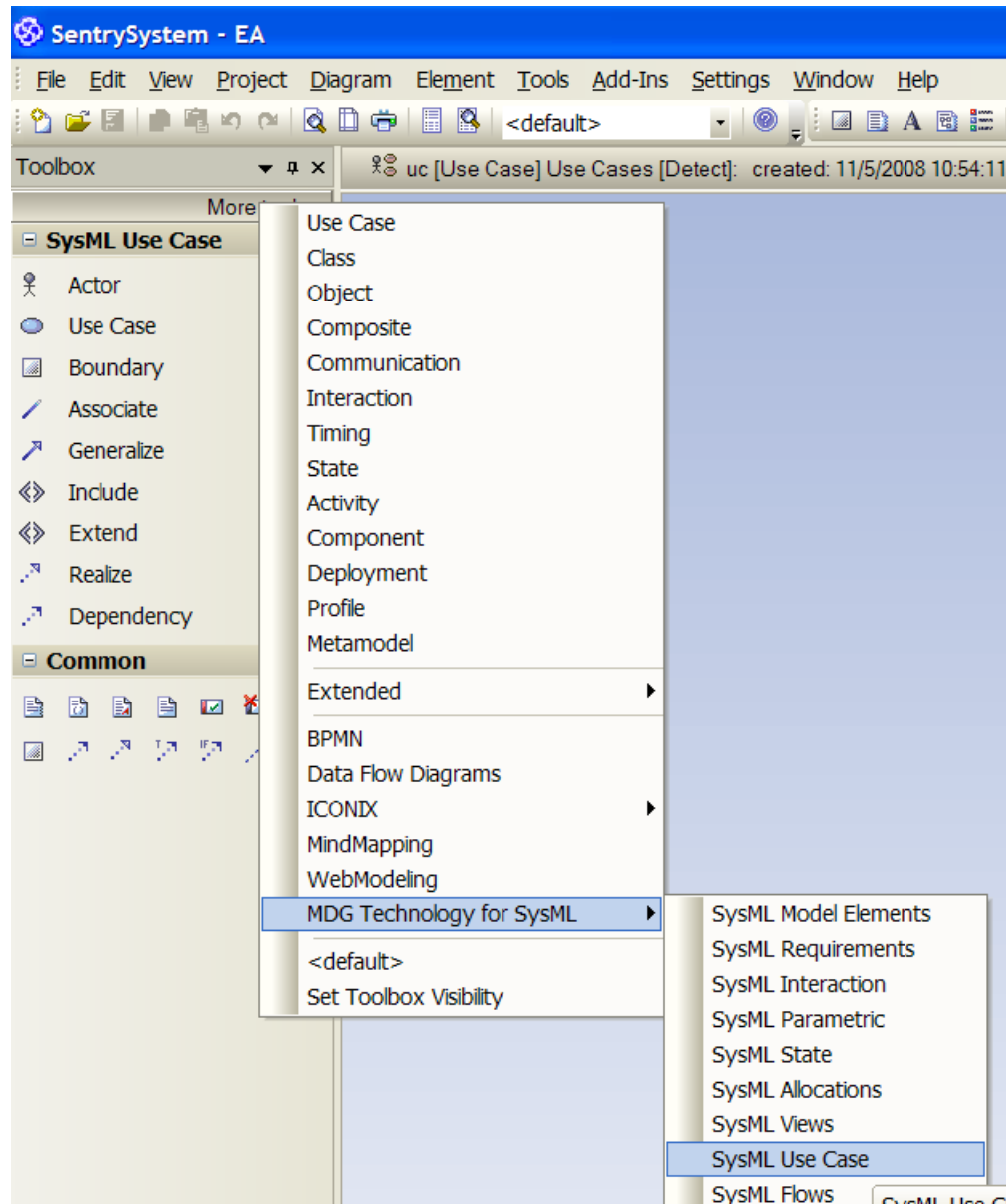
The New Diagram dialog is displayed.



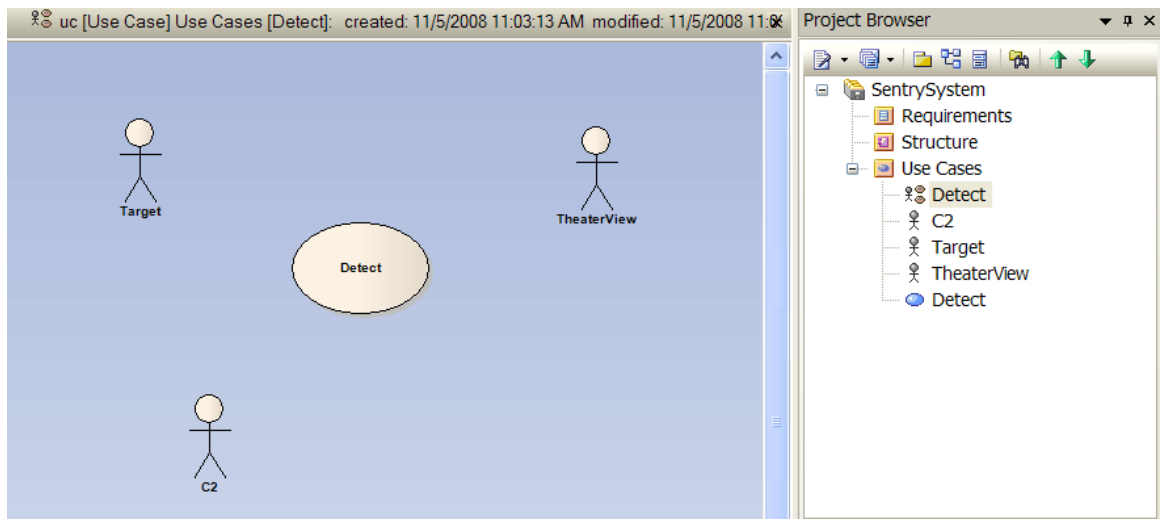
2. In the *Name* field enter *Detect*, in the *SelectFrom* list select **SysML** and from the *DiagramTypes* list select **UseCase**. Click **OK**.


The new Use Case diagram is opened and shown in the project browser.

3. In the *Toolbox* pane select **More tools...** and then select **MDG Technology for SysML -> SysML Use Case**.



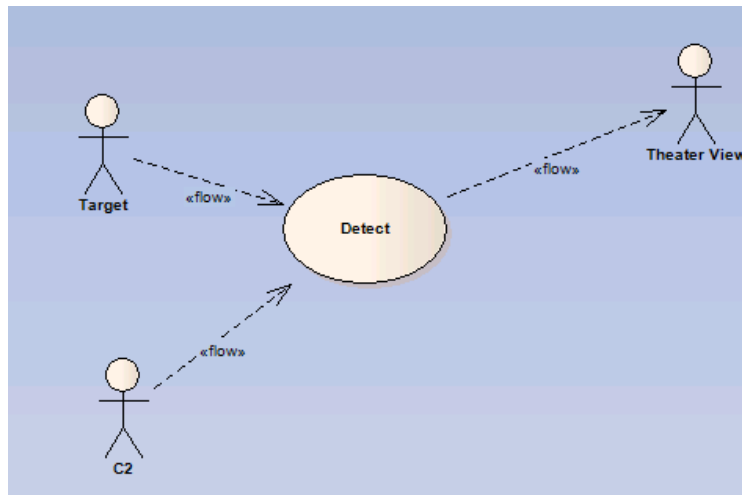
- Using the tool icons from the *Toolbox*, add a *Use Case* and name it *Detect*. Add 3 *Actors* named *Target*, *C2* and *TheaterView*.



- From the Toolbox Common tools use the **Information Flow** tool icon  to add an interaction from the actor **Target** to the use case **Detect**.

The Information Items Conveyed dialog box is displayed.


- Click **OK**.
- Add 2 more interactions, one from **Detect** to **TheaterView**, and another from **C2** to **Detect**.



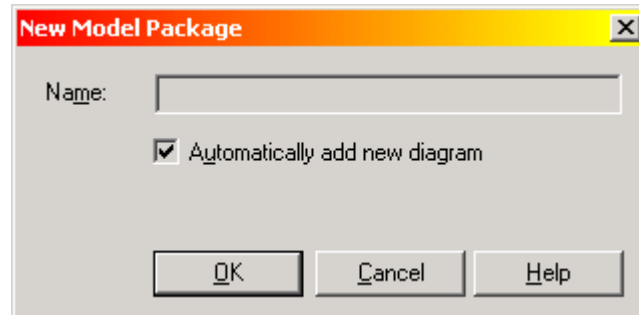
- Click **control-S** to save your changes to the model.

Task 3: Create Packages and Diagrams in the Structure View

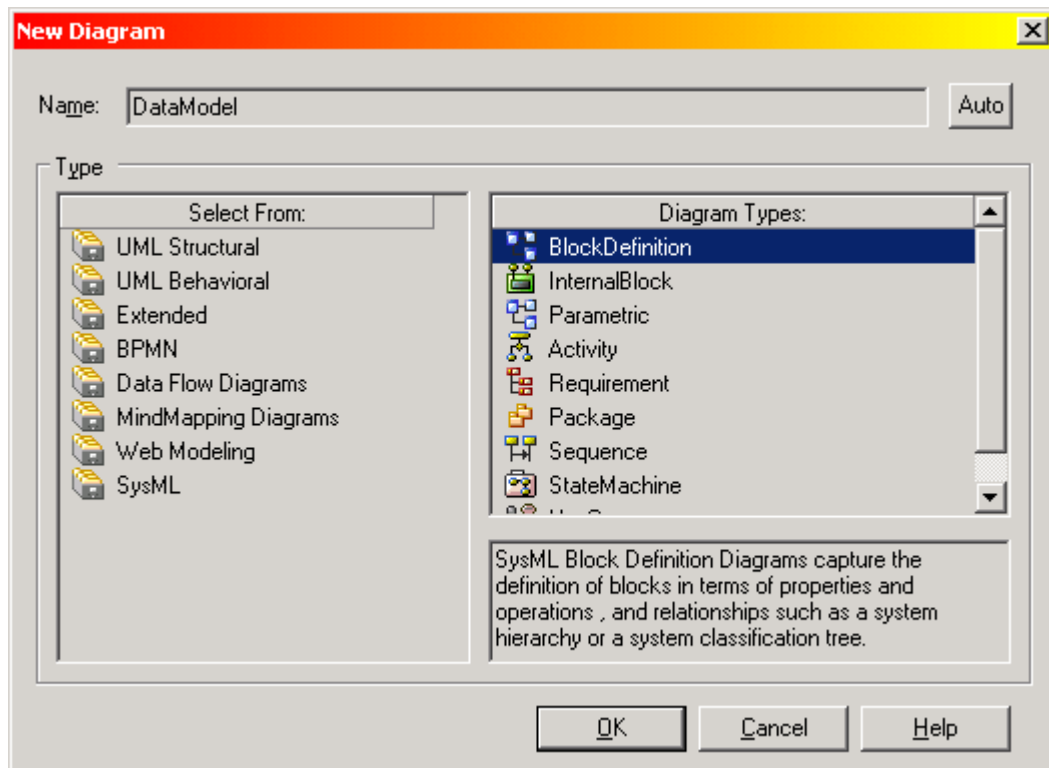
PROCEDURE: Create Package Hierarchy in the Structure View

1. Select the Structure view from the Project Browser. Use the **Package** tool icon in the Project browser  to add a new package to the Structure View.

The New Model Package dialog is displayed.



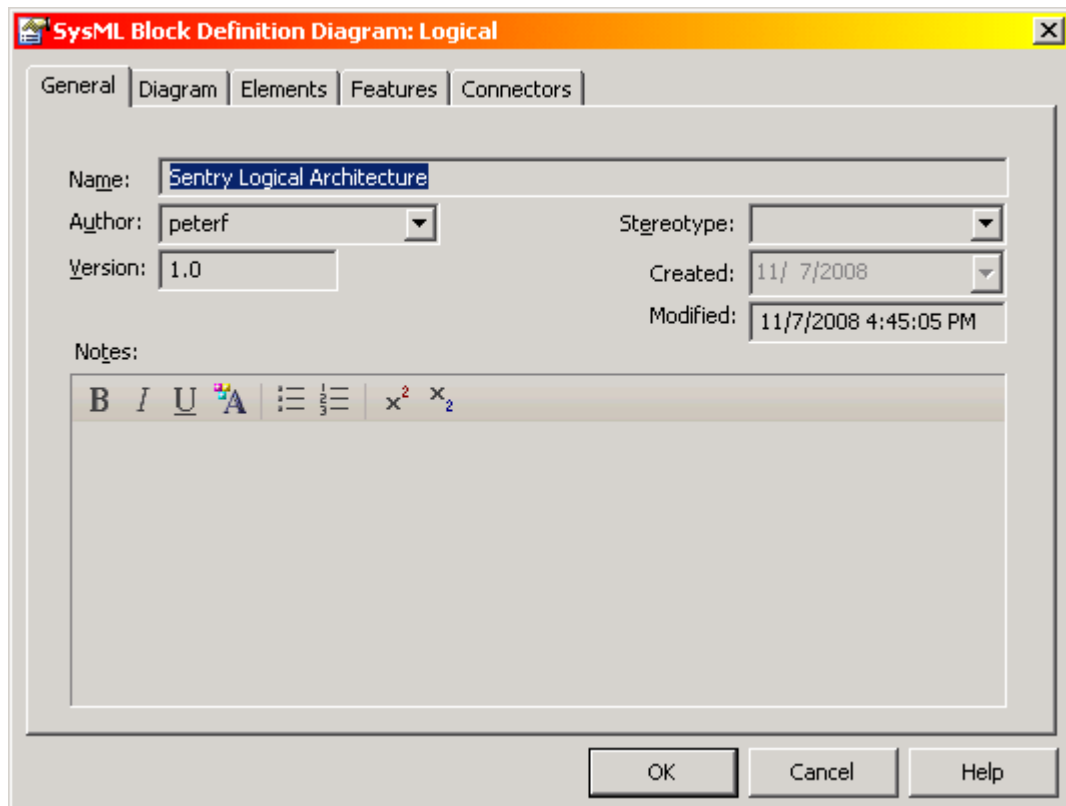
2. In the *name* field enter DataModel and click **OK**.
3. In the New Diagram dialog select **BlockDefinition** for the diagram type:




4. Add 3 more packages named Enterprise, Logical and Physical to the Structure view.

PROCEDURE: Complete the Block Definition Diagram capturing the Logical Architecture

1. Select the Block Definition Diagram (BDD) Logical in the Logical package from the Project Browser. Right click and pick **Properties**. Rename the diagram Sentry Logical Architecture, and click **OK**.

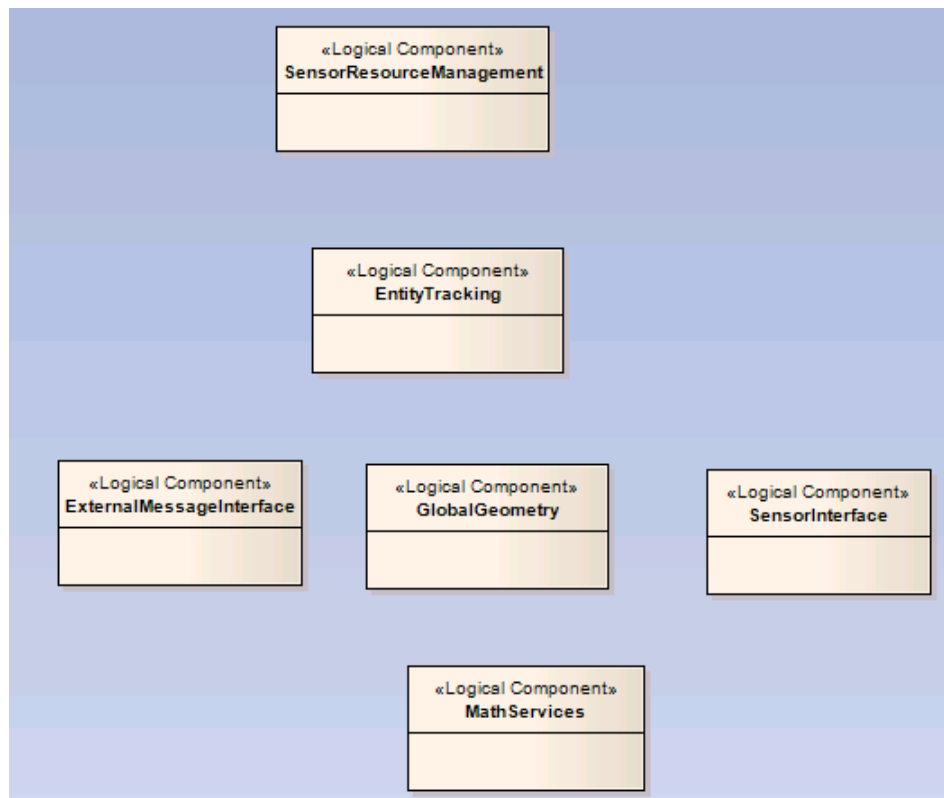



2. **Double click** the diagram Sentry Logical Architecture to open it
3. From the *Toolbox SysML Block Definition*, use the *Block* tool  to create a new block on the Sentry Logical Architecture diagram.

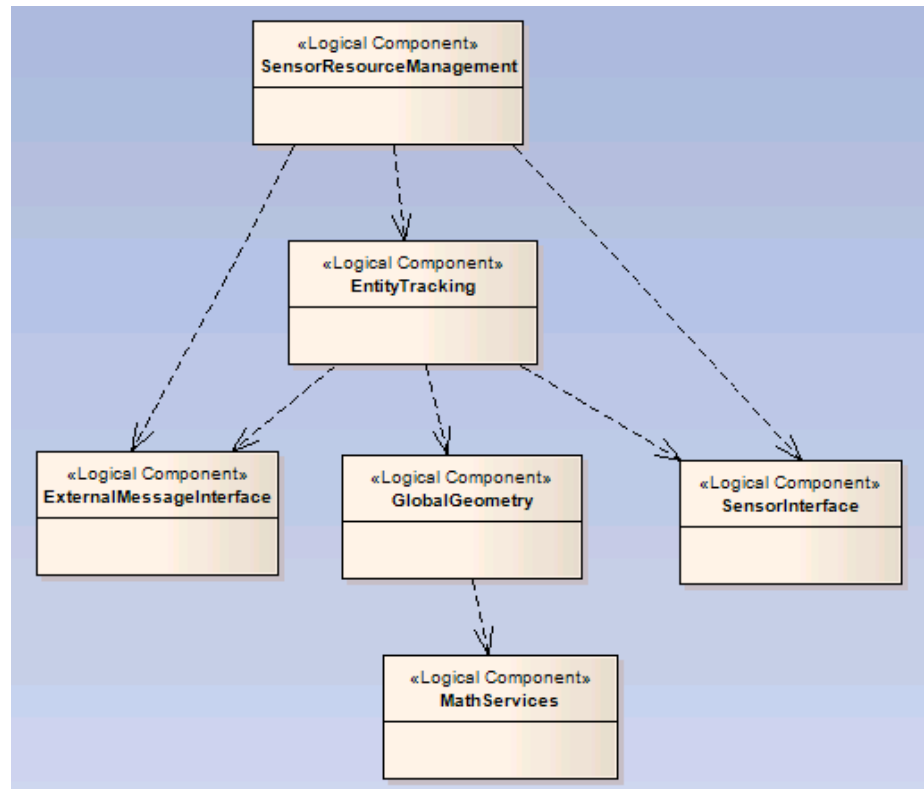
A Block dialog is displayed.

4. In the *Name* text entry box enter the name EntityTracking. In the *Stereotype* box type Logical Component (or select it from the list). Click **OK**.
5. Repeat steps 3 and 4 to add more blocks named ExternalMessageInterface, GlobalGeometry, MathServices, SensorInterface, and

SensorResourceManagement to the Sentry
Logical Architecture diagram as shown below.





6. Use the Dependency tool icon  Dependency to add dependencies as shown below.




7. Click **control-S** to save your changes to the model.

Task 4: Create a Logical Scenario Model

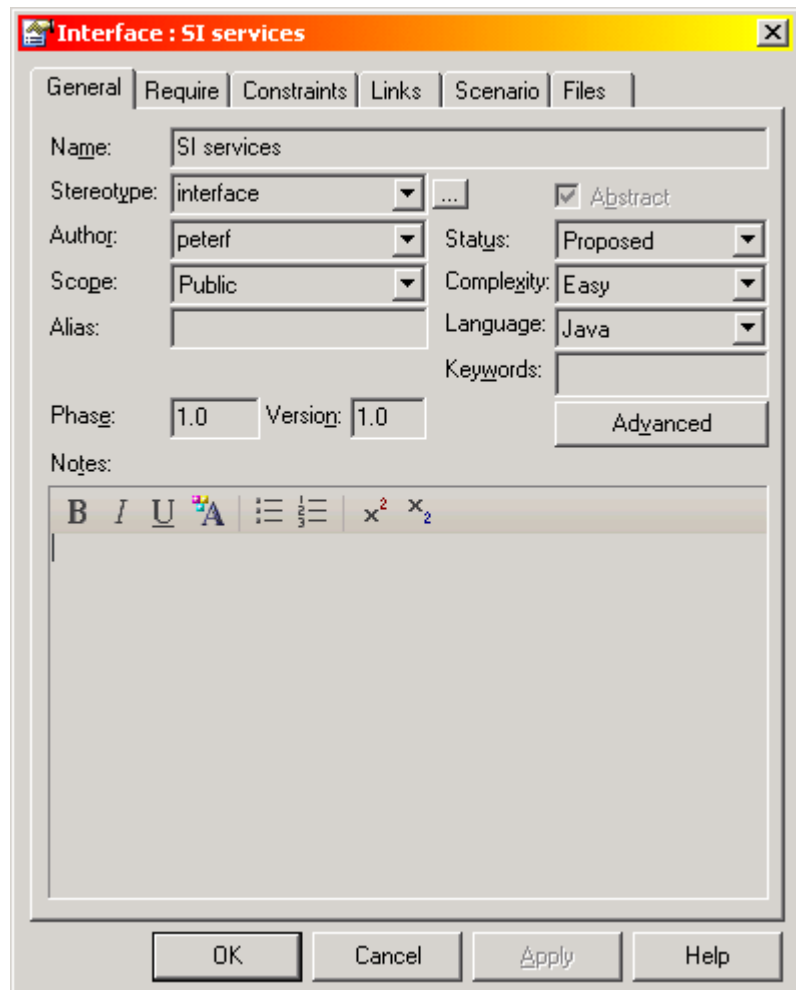
PROCEDURE: Add a Sequence Diagram to Capture a new Logical Scenario

1. Select the Logical package from the Project Browser and use the **Add a Package** tool  at the top of the Project Browser to create a new Package with the name Logical Scenario Models. Deselect *Automatically add new diagram* and click **OK**.
2. Select the *Logical Scenario Models* package from the Project Browser and use the **New Diagram** tool  at the top of the Project Browser to create a new diagram with type *Sequence* with the name NewSubject.

The new Sequence Diagram is opened.

3. In the Project Browser select the SensorInterface Block and use the **Create Element** tool  at the top of the Project Browser to create a new element of type *Interface* with the name SI services. Click **OK** in the New Element Dialog.

The Interface dialog appears:



4. Click **OK** in the Interface dialog.

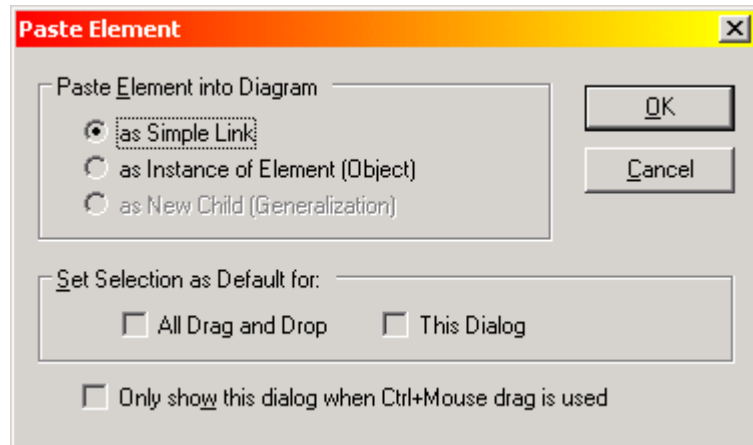
A name warning dialog appears:



5. Click **Yes**.

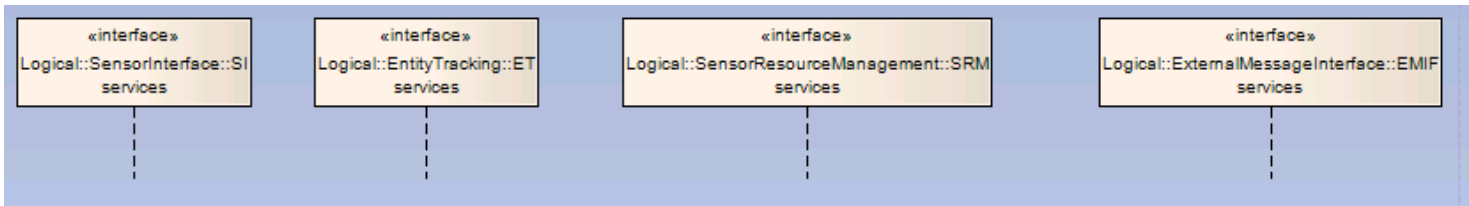
6. Drag the new Interface *SI services* from the Project Browser to the Sequence Diagram and place it in the upper left corner.


The Paste Element dialog appears:



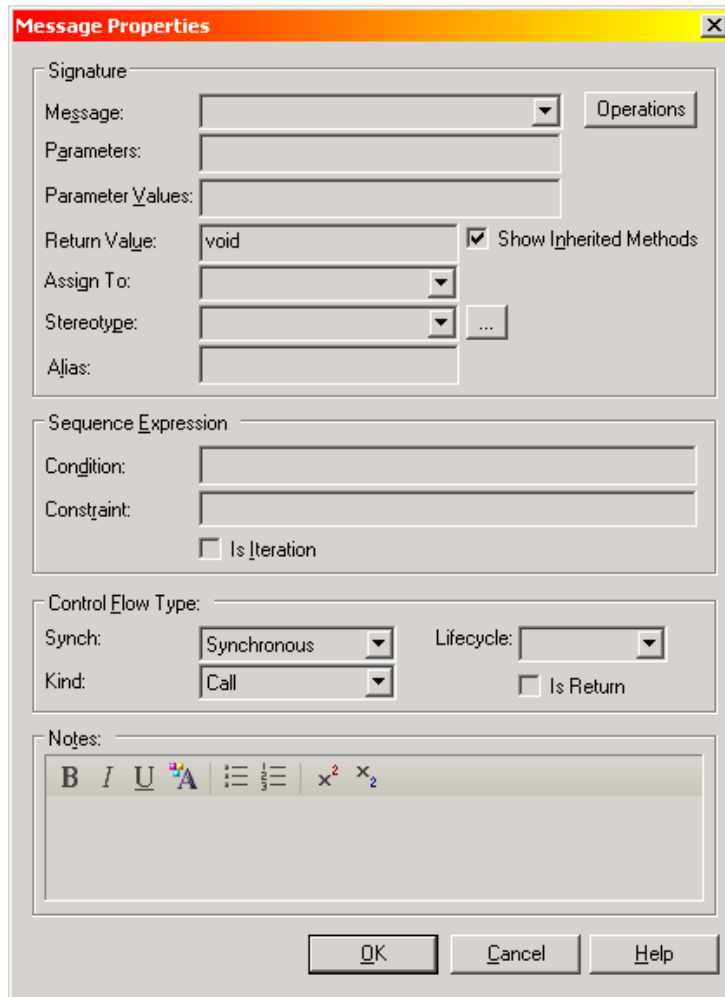
7. Click **OK**.
8. Repeat steps 3 through 5 to add interfaces to all other Logical Component Blocks in the *Logical Package*. Name the new interfaces *ET services*, *EMIF services*, *GG services*, *MS services*, and *SRM services*,
9. Repeat steps 6 and 7 to place the Interfaces (in order from left to right) *ET services*, *SRM services*, and *EMIF Services*.

The Sequence Diagram should look like



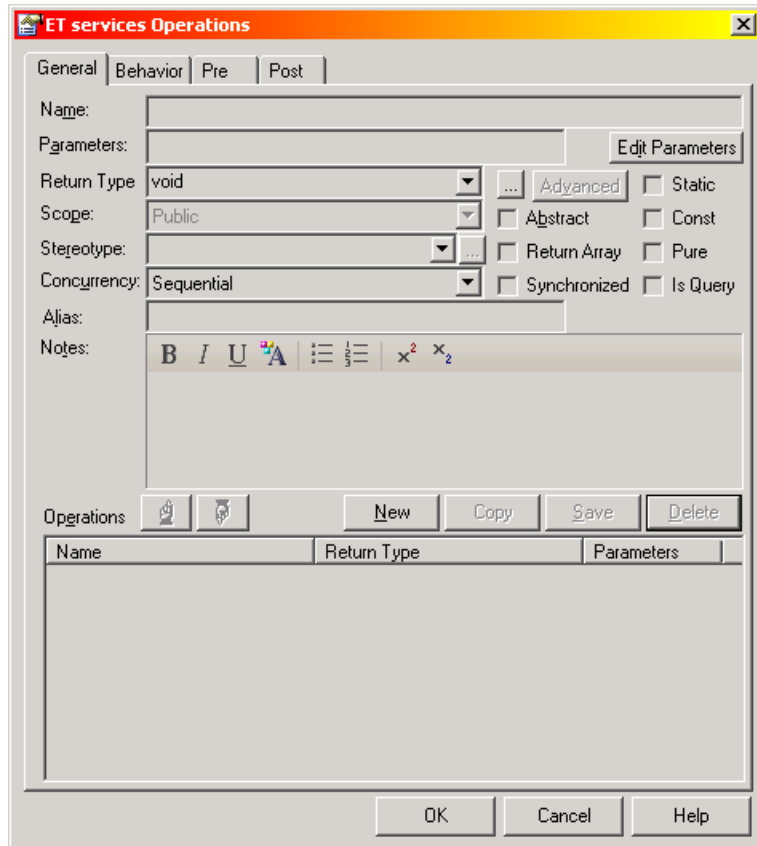
10. From the SysML Interaction tools in the Toolbox, use the **Message**  tool to draw a message line from the *SI services* Lifeline to the *ET services* Lifeline.

The Message Properties dialog appears:



11. Click the **Operations** button.

The Operations dialog appears for *ET services*:



Name	Return Type	Parameters


12. Enter `NewObservation` in the Name field.

13. Enter the following description into the Notes field:



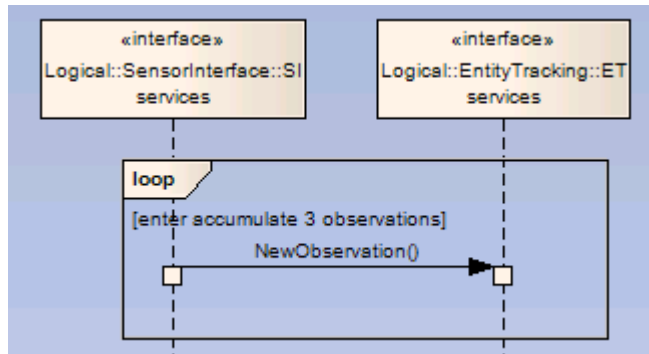
Reports a new observation. The observation could correlate to an existing entity, or indicate a new entity.

14. Click **Save, OK, OK**.

15. Use the **Fragment**  tool to draw a Fragment box over the `NewObservation` message on the diagram. Select a Type of **loop** and for the Condition enter `accumulate 3 observations`.

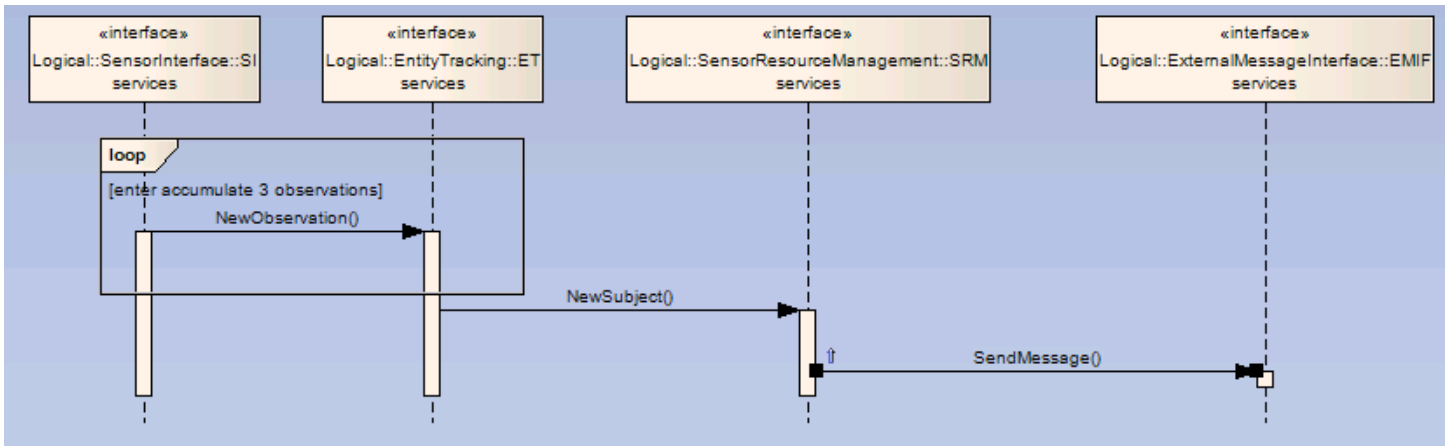
16. Click **Save** and **OK**.

Stretch and reposition elements like:



17. Repeat steps 10-14 to place a message below the loop block, from *ET services* to *SRM services* named *NewSubject*.
18. Repeat steps 10-14 to place a message below the *NewSubject* message, from *SRM services* to *EMIF services* named *SendMessage*.

Your sequence diagram should look something like:



19. Click **control-S** to save your changes to the model.

Task 5: Create New Value Types

PROCEDURE: Add a new scalar value type and a structure block to the System DataModel

1. Select the *DataModel* package from the Project Browser and use the **New Diagram** tool at the top of the Project Browser to create a new diagram with type *BlockDefinition* with the name *SystemTypes*.

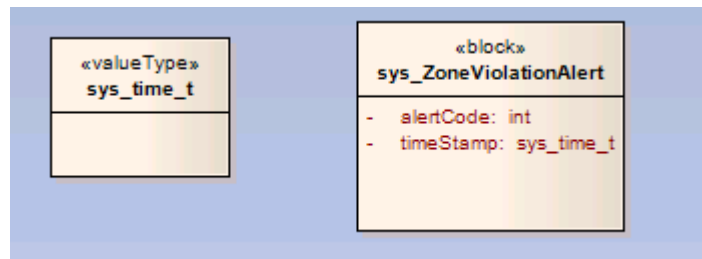
The new block definition diagram is opened.

2. From the *Toolbox SysML Block Definition*, use the **ValueType** tool icon to create a new ValueType on the *SystemTypes* diagram.


A ValueType dialog is displayed.

3. In the *Name* text entry box enter the name `sys_time_t`. Click **OK**.
4. Use the **Block** tool icon to add a structure block to the diagram. In the Block dialog enter the name `sys_ZoneViolationAlert`.
5. Select the **Details** tab and click the **Attributes** button. Enter the name `alertCode` (the type is `int` – default) and click the **Save** button.
6. Click the **New** button, enter the name `timeStamp`, enter `sys_time_t` in the Type field and click **Save**, then **Close**.
7. On the Block Definition Diagram right-click on the `sys_ZoneViolationAlert` Block and select **Feature Visibility**. Under Attribute Visibility select **All** and click **OK**.

Your DataModel value types should look like this:



PROCEDURE: Add a new scalar value type to the GlobalGeometry Logical Component

1. In the Project Browser select the *GlobalGeometry* block under the *Logical* package and use the **New Diagram** tool  at the top of the Project Browser to create a new *BlockDefinition* diagram with the name *GlobalGeometry*.

The new Block Definition Diagram is opened.

2. From the *Toolbox SysML Block Definition*, use the **ValueType** tool icon to create a new ValueType on the *GlobalGeometry* diagram.
3. In the *Name* text entry box enter the name `gg_space_t`. Click **OK**.
4. Click **control-S** to save your changes to the model.

Task 6: Detailed Blocks in a Logical Component

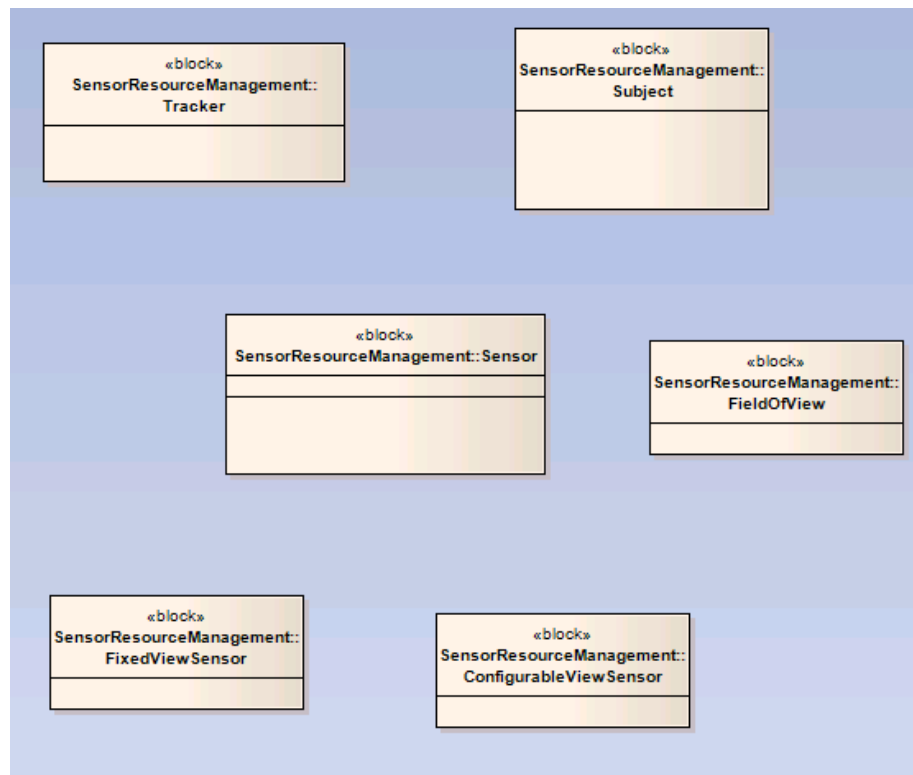
PROCEDURE: Create a BDD for the SensorResourceManagement Logical Component

1. In the Project Browser under the *Logical* package select the block *SensorResourceManagement*.

Using the **New Diagram** tool  in the Project Browser create a new *Block Definition* diagram named *SensorResourceManagement*.

The new block definition diagram is opened.

2. Using the **Block** tool from the SysML Block Definition Toolbox, create blocks named *Subject*, *Tracker*, *Sensor*, *FieldOfView*, *FixedViewSensor*, and *ConfigurableViewSensor*.




3. Select the *Subject* block, right click and select **Properties...** Enter the following description into the Notes field




An object which is being tracked. It can be friend or foe.

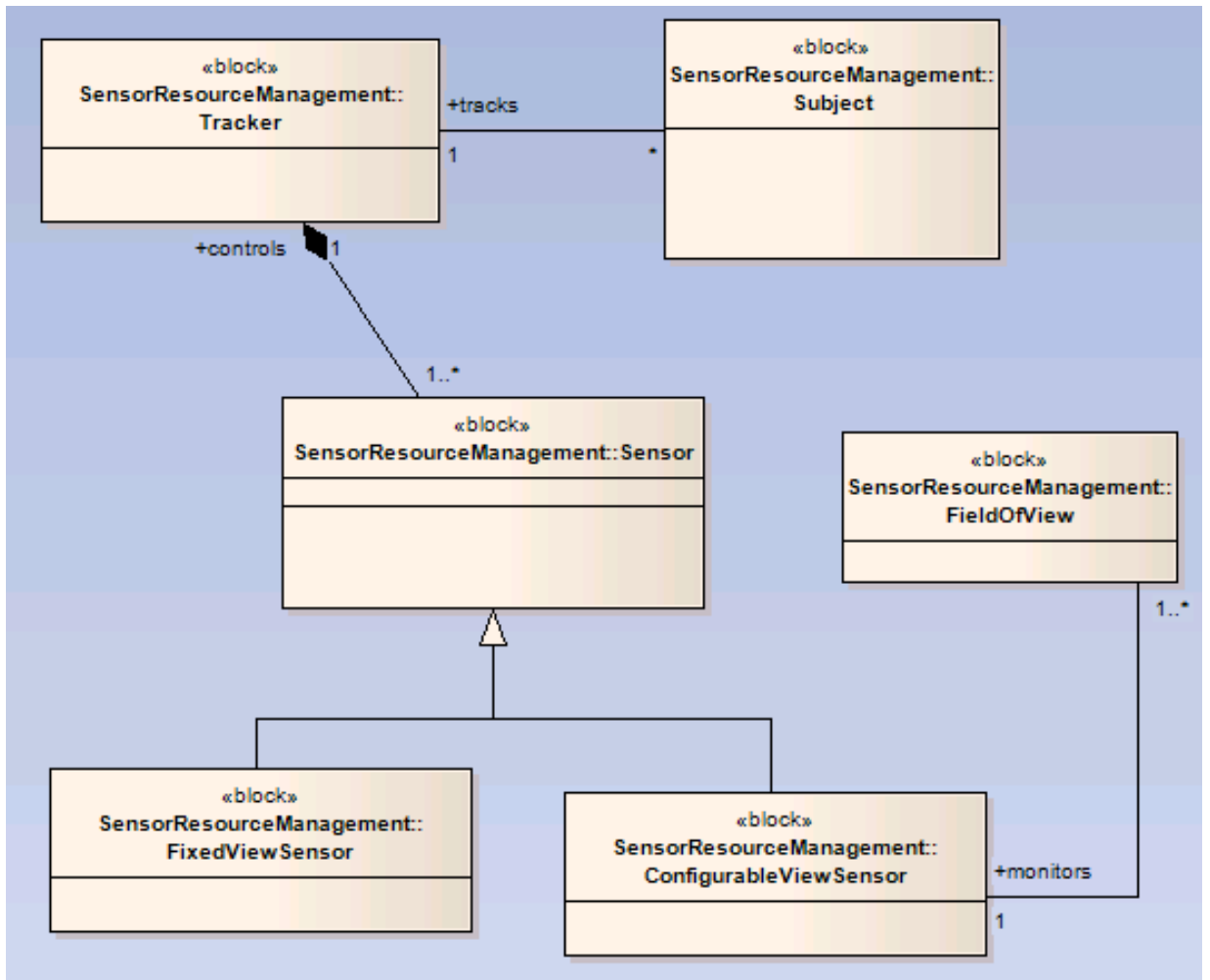
4. Click **OK**.

PROCEDURE: Add Generalizations and Associations between SensorResourceManagement Blocks

1. Using the **Generalize** tool in the Toolbox draw a generalization relationship from *FixedViewSensor* to *Sensor*, and from *ConfigurableViewSensor* to *Sensor*.
2. Select one of the generalizations, right click and select **LineStyle -> Tree Style-Vertical**. Repeat for the second generalization.
3. Use the **ReferenceAssociation** tool  to draw an association from *Tracker* to *Subject*.
4. Right click on the new association and select **Association Properties...**

The Association Properties Dialog is opened.

5. Click on the *Source Role* tab, and in the *Tracker Role* text entry field enter `tracks`. From the *Multiplicity* pull down, select 1.
6. Click on the *Target Role* tab, and from the *Multiplicity* pull down, select `*`. Click **OK**.
7. Use the **ReferenceAssociation** tool to draw an association from *ConfigurableViewSensor* to *FieldOfView*, and use the **PartAssociation** tool  to draw a Compose association from *Sensor* to *Tracker*, and set the role names and cardinality as shown in the diagram below:



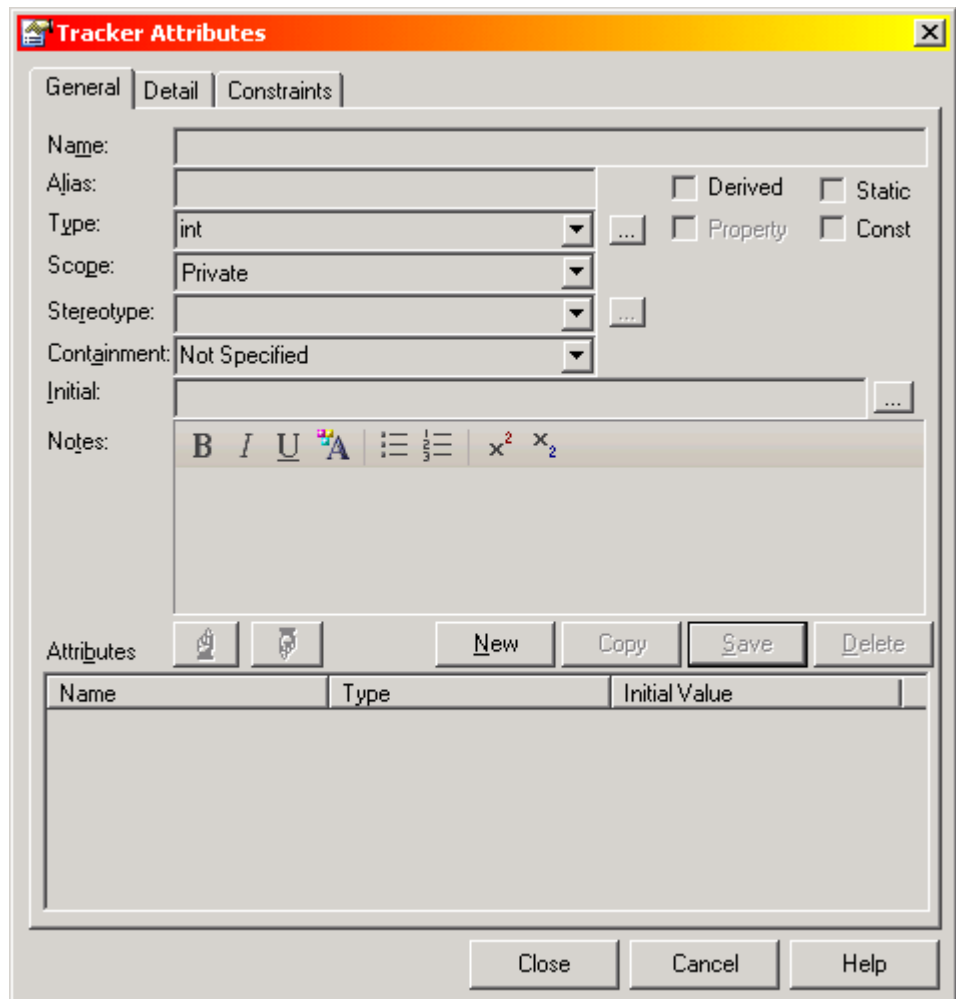
8. Click **control-S** to save your changes to the model.

PROCEDURE: Add Value Properties to SensorResourceManagement Blocks

Value properties of blocks are abstracted as attributes.

1. In the Project Browser, select the block *Tracker*, right click and select **Attributes...**

A Dialog named *Tracker Attributes* is opened.



The dialog box is titled "Tracker Attributes" and has three tabs: "General", "Detail", and "Constraints". The "General" tab is selected. It contains the following fields and options:

- Name: [Text Entry Field]
- Alias: [Text Entry Field]
- Type: [Dropdown Menu] (set to "int")
- Scope: [Dropdown Menu] (set to "Private")
- Stereotype: [Dropdown Menu]
- Containment: [Dropdown Menu] (set to "Not Specified")
- Initial: [Text Entry Field]
- Notes: [Rich Text Editor]

Checkboxes on the right side of the dialog include:

- Derived
- Static
- Property
- Const

Buttons at the bottom of the dialog include: Close, Cancel, Help, and a set of buttons for "Attributes": New, Copy, Save, and Delete.

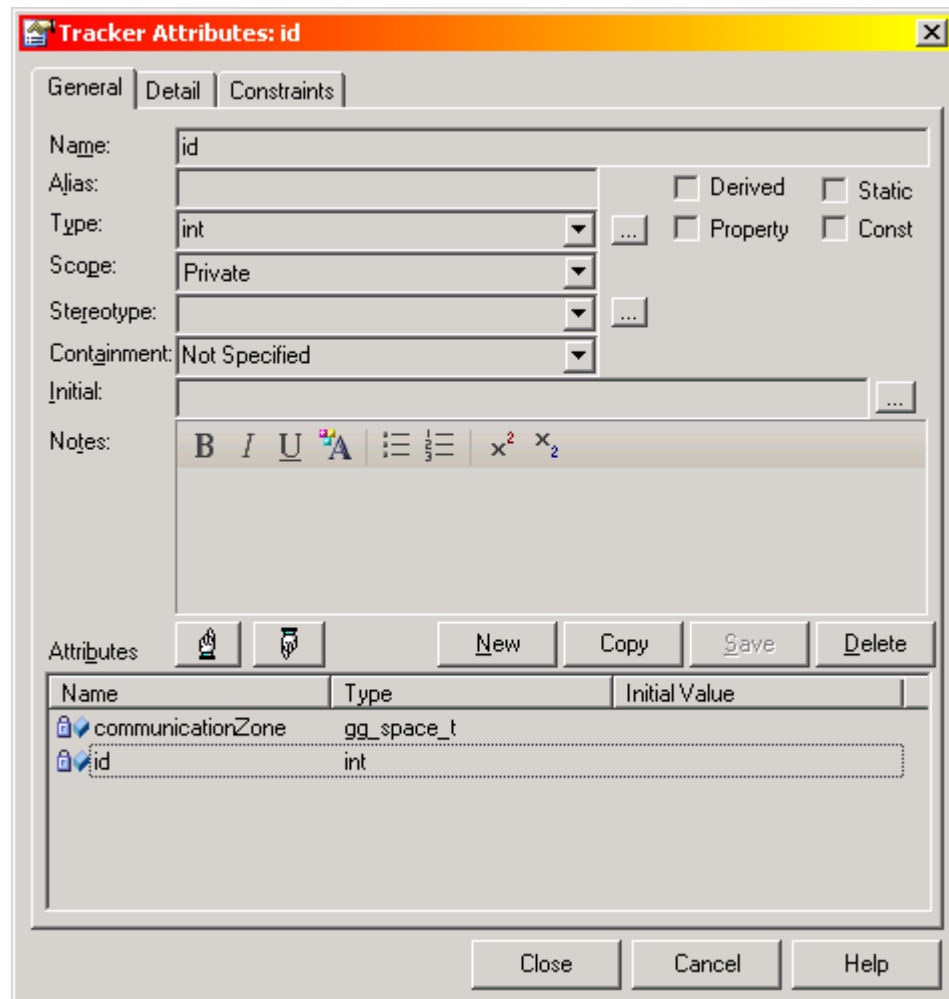
Name	Type	Initial Value

- In the *Name* text entry field enter *communicationZone*. In the *type* selection list choose *gg_space_t* and then click **Save**.

A new attribute named *communicationZone* is listed at the bottom of the dialog.

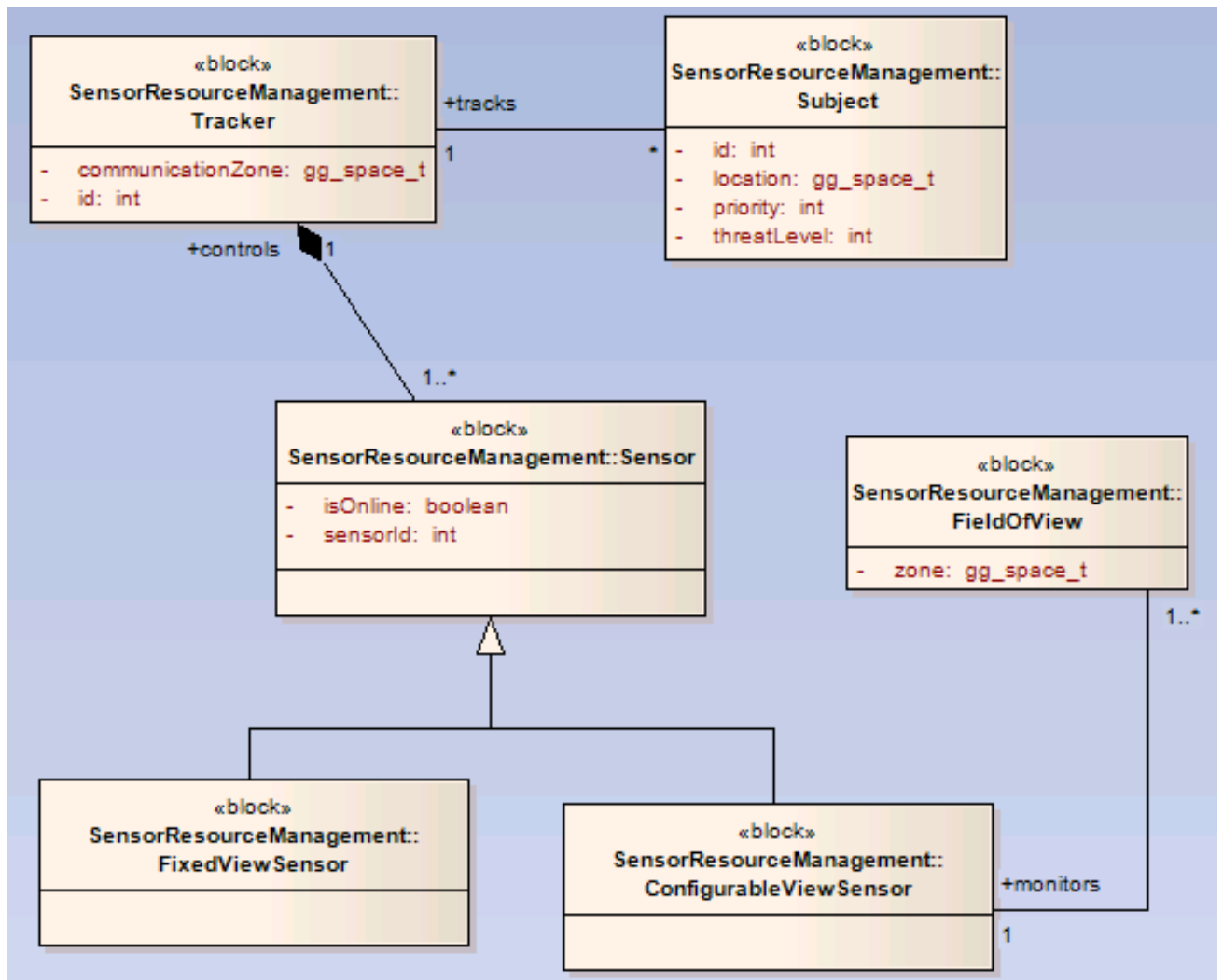
- Click **New**. Create a new attribute with name *id*, and type *int*, and click **Save**.

Tracker Attributes dialog with two new attributes:



4. Click **Close**.
5. On the BDD right-click on the *Tracker* Block and select **Feature Visibility**. Under Attribute Visibility select **All** and click **OK**.

6. Add Value Properties to other blocks as shown:



7. Click **control-S** to save your changes to the model.

PROCEDURE: Add Operations to SensorResourceManagement Blocks

1. In the Project Browser, select the block *Sensor*, right-click and pick **Operations**.
2. In Name enter *getActiveFieldOfView* and pick a Return Type of *gg_space_t*.
3. Enter the description below in the Notes field:



Return the currently active field of view monitored by this sensor.

4. Click **Save** and **Close**.
5. Add a *getActiveFieldOfView* operation with a return type of *gg_space_t* to the *FixedViewSensor* and enter the description below in the Notes field:

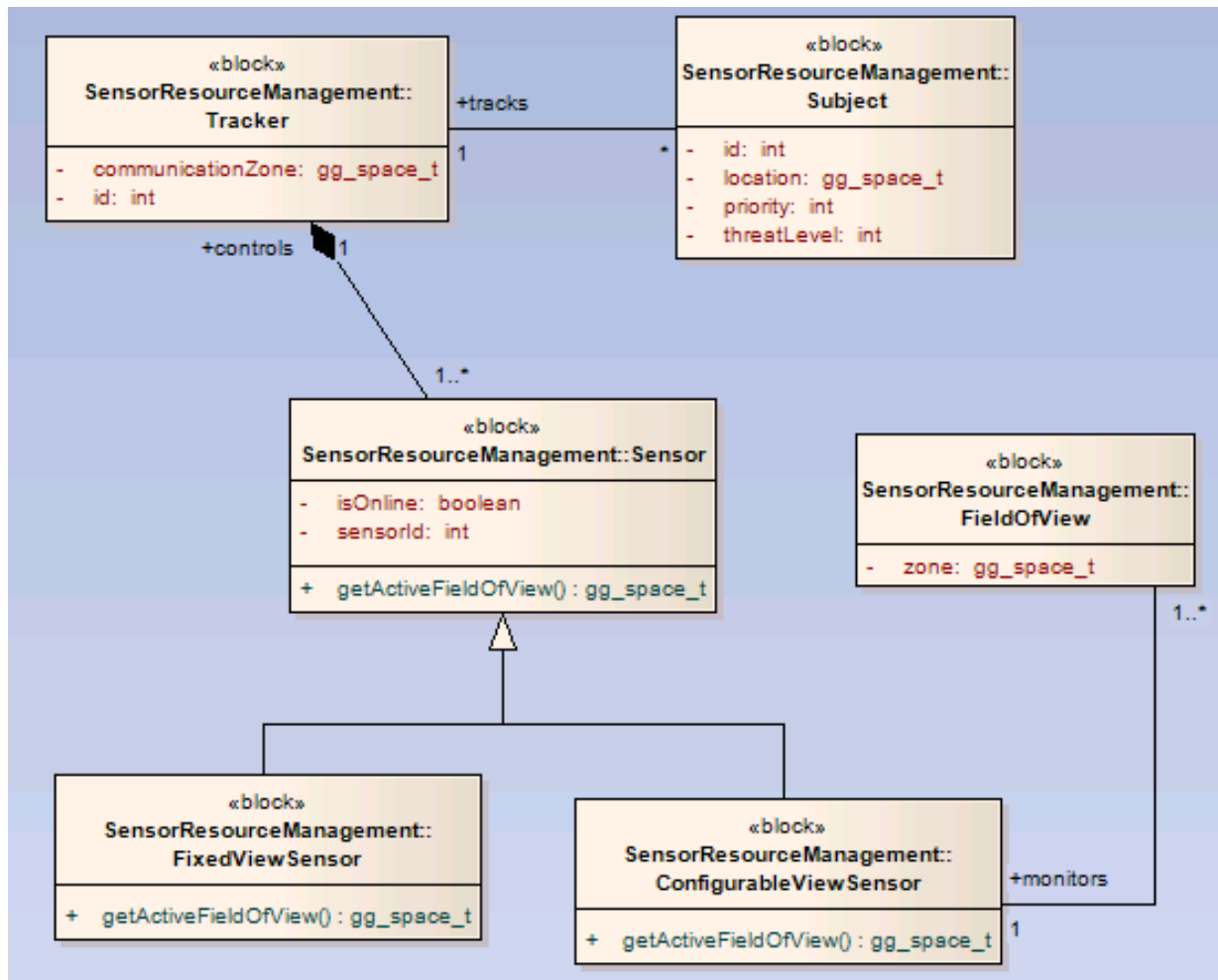


Override supertype behavior to return a fixed field of view for this sensor.

6. Add a *getActiveFieldOfView* operation with a return type of *gg_space_t* to the ConfigurableViewSensor and enter the description below in the Notes field:




Override supertype behavior to return a summation of the associated fields of view for this sensor.






7. Click **control-S** to save your changes to the model.

Task 7: Model State Behavior for a Logical Block

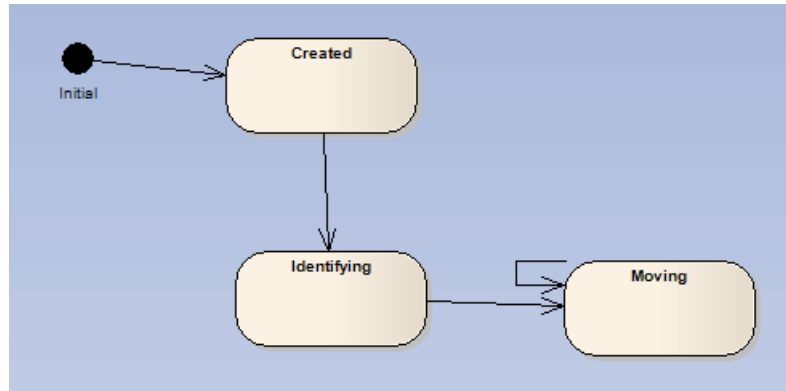
PROCEDURE: Create a State Machine Diagram for the Subject Block

1. In the Project Browser, select the block Subject.
Using the **New Diagram** tool  in the Project Browser create a new StateMachine diagram named Subject state machine.

The new state machine diagram is opened.


2. Use the **Initial** tool  to create an Initial Pseudostate. Click OK in the State dialog.
3. Use the **State** tool  to create States named *Created*, *Identifying* and *Moving*.
4. Use the Transition  tool to add Transitions from the Initial Pseudostate to *Created*, from *Created* to *Identifying*, from *Identifying* to *Moving*, and from *Moving* to *Moving* (click twice within *Moving*).

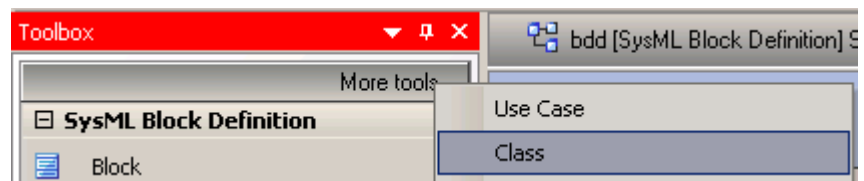
Arrange your diagram like this:




5. Select the self Transition (transition to itself) of the *Moving* state, right click and select **Line Style**→**Bezier**.

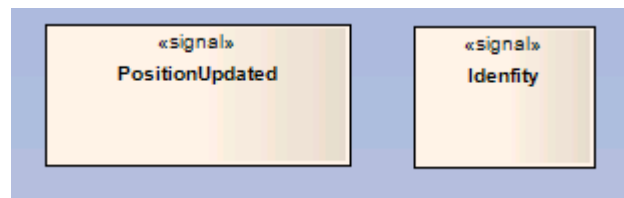
PROCEDURE: Define Subject Block Signals

1. In the Project Browser, select the block *Subject*. Use the **New Diagram** tool  in the Project Browser to create a new BlockDefinition diagram named *Subject definition*.
2. At the top of the Toolbox click **More Tools**, then click **Class**:



3. Use the **Signal** tool  to place two Signals on the BDD, named *Identify* and *PositionUpdated*.

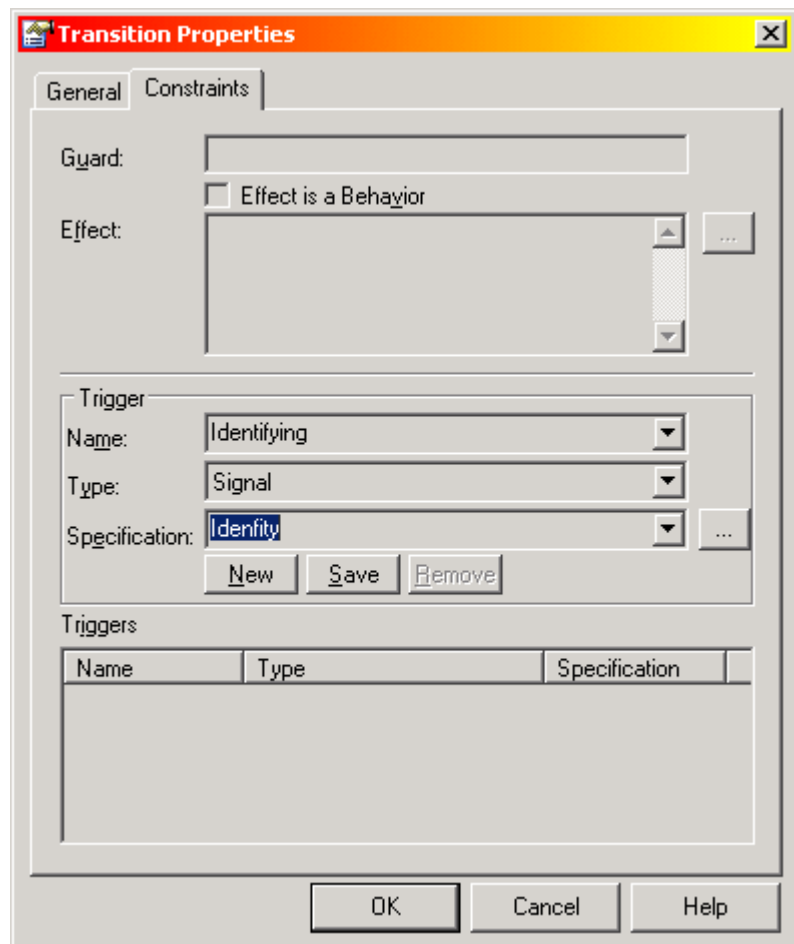
The Subject BDD looks like:



- Right-click the Signal *PositionUpdated* and select **Attributes**. Add the attribute *new_position* with Type *gg_space_t*. Click **Save** and **Close**.

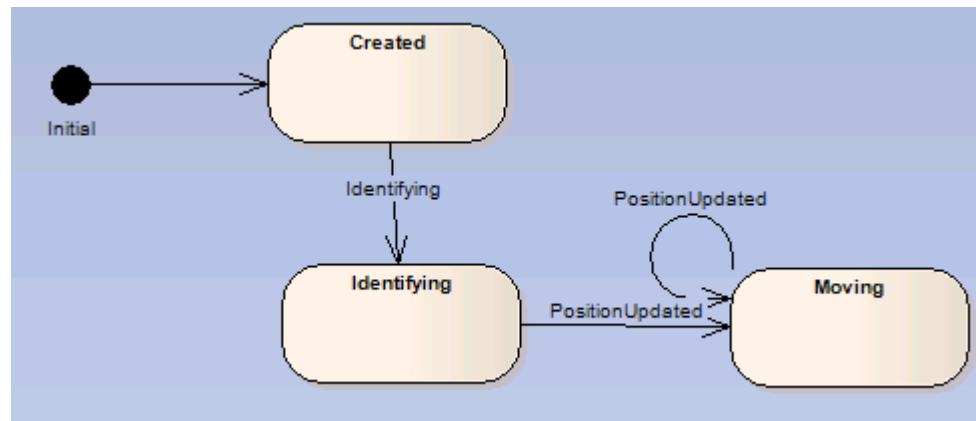
PROCEDURE: Add Signal Triggers to the Subject State Machine

- At the bottom of the main diagram pane, select the *Subject state machine* State Machine.
- Double-click on the transition from *Created* to *Identifying*.
- In the Transition dialog on the **Constraints** tab in the Trigger region enter *Identify* in the *Name* field, select **Signal** for *Type* and select the *Identify* signal under *Specification*.



- Click **Save** and **OK**.
- Repeat this to bind the *PositionUpdated* Signal as the Trigger for the two Transitions into the *Moving* State.

The Subject BDD looks like:



6. Click **control-S** to save your changes to the model.

Congratulations! You have constructed the SentrySystem Model with Enterprise Architect.

Transforming the *SentrySystem* Model

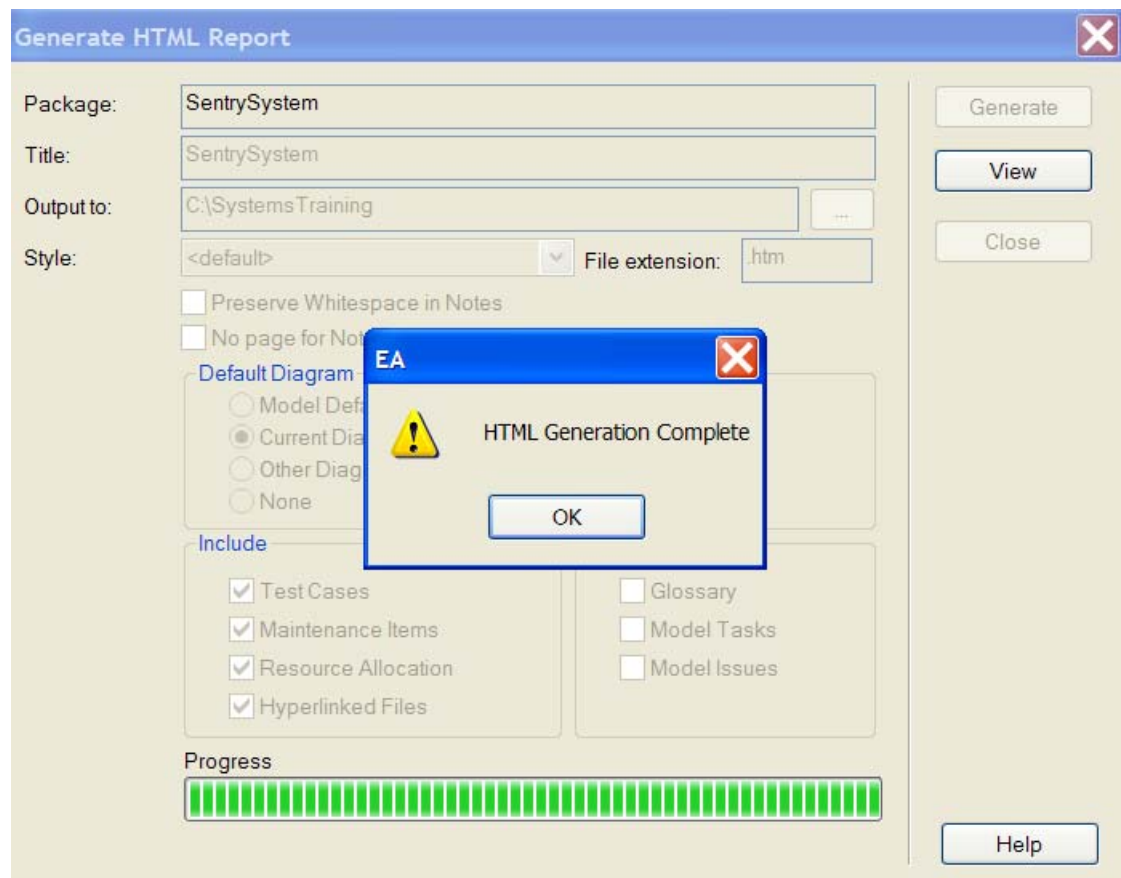
Task 1: Generate HTML documentation from Project

PROCEDURE: Generate documentation for the *SentrySystem* project.

1. In the Project Browser, select the project *SentrySystem*. From the Enterprise Architect menu, select **Project->Documentation->HTML Report...**

The Generate HTML Report dialog is opened.

2. Enter *SentrySystem* QSG report in the *Title* text entry field, and in the *Output to* field use the browse (...) button to choose a target directory. Click **Generate...**



3. Click **OK**. Click **View**.

Congratulations! You transformed your Systems Model to documentation.