



BRAHMS
Brahms Composer v1.3
User Guide

TM0305

Version 0.1 – Draft
19 May, 2003

REVISION HISTORY

Version	Contact	Action	
Version 0.1 Draft 05/19/03	Mike Scott 212/941-1845	New	Initial Version

TABLE OF CONTENTS

1.	OVERVIEW OF THE COMPOSER APPLICATION.....	6
2.	LAUNCHING THE COMPOSER APPLICATION	6
3.	CREATING A NEW MODEL.....	7
4.	OPENING AN EXISTING MODEL	9
5.	IMPORTING AN EXISTING MODEL	11
6.	THE LAYOUT OF THE COMPOSER APPLICATION WINDOW	13
7.	CREATING AND EDITING CONCEPTS USING THE VISUAL DESIGNERS	15
7.1.	CREATING CONCEPTS USING THE 3-CHARACTER KEYBOARD SHORTCUTS.....	17
7.2.	CREATING CONCEPTS USING THE MODEL PANE TOOLBAR BUTTONS.....	18
7.3.	CREATING CONCEPTS USING THE 'MODEL' MENU.....	19
7.4.	FINDING CONCEPT NODES IN THE VISUAL DESIGNER MODEL TREES	20
7.5.	EDITING CONCEPTS USING THE CONCEPT EDITOR	21
7.6.	EDITING THE INHERITANCE RELATIONS OF CONCEPTS USING DRAG AND DROP AND DELETE	22
7.7.	EDITING THE 'PARTOF' RELATION FOR OBJECTS AND CONCEPTUALOBJECTS USING DRAG AND DROP AND DELETE 23	
7.8.	EDITING THE 'PARTOF' RELATION FOR AREAS USING DRAG AND DROP AND DELETE	24
7.9.	EDITING THE 'LOCATION' PROPERTY OF AGENTS AND OBJECTS USING DRAG AND DROP AND DELETE	25
7.10.	CREATING PATHS USING DRAG AND DROP.....	25
7.11.	EDITING PATHS USING THE PATH EDITOR.....	26
8.	CREATING AND EDITING CONCEPT ELEMENTS USING THE VISUAL DESIGNERS	28
8.1.	ADDING ATTRIBUTES AND RELATIONS TO CONCEPTS USING THE KEYBOARD SHORTCUTS.....	28
8.2.	CREATING ATTRIBUTES AND RELATIONS USING THE MODEL PANE TOOLBAR BUTTONS	30
8.3.	CREATING ATTRIBUTES AND RELATIONS USING DRAG AND DROP IN THE MODEL TREES	30
8.4.	MOVING OR COPYING ATTRIBUTES AND RELATIONS USING DRAG AND DROP IN THE MODEL TREES.....	31

8.5.	EDITING ATTRIBUTES AND RELATIONS USING THE MODEL TREE CELL EDITORS.....	32
8.6.	ADDING INITIAL FACTS AND BELIEFS TO ACTIVE CONCEPTS USING THE KEYBOARD SHORTCUTS	32
8.7.	CREATING INITIAL FACTS AND BELIEFS USING THE MODEL PANE TOOLBAR BUTTONS.....	34
8.8.	CREATING INITIAL FACTS AND BELIEFS USING DRAG AND DROP IN THE MODEL TREES.....	34
8.9.	MOVING OR COPYING INITIAL STATEMENTS USING DRAG AND DROP IN THE MODEL TREES	36
8.10.	EDITING INITIAL STATEMENTS USING THE MODEL TREES.....	36
8.11.	ADDING ACTIVITIES TO ACTIVE CONCEPTS USING THE 3-CHARACTER ABBREVIATIONS IN THE MODEL TREES.....	36
8.12.	ADDING ACTIVITIES USING THE ACTIVITY MODEL PANE TOOLBAR BUTTONS	39
8.13.	MOVING OR COPYING ACTIVITIES USING DRAG AND DROP IN THE MODEL TREES.....	39
8.14.	EDITING ACTIVITIES USING THE ACTIVITY EDITOR	40
8.15.	CREATING AND EDITING ACTIVITY PARAMETERS IN THE MODEL TREES	42
8.16.	ADDING WORKFRAMES AND THOUGHTFRAMES TO ACTIVE CONCEPTS AND COMPOSITE ACTIVITIES USING THE KEYBOARD SHORTCUTS	44
8.17.	CREATING WORKFRAMES AND THOUGHTFRAMES USING THE MODEL PANE TOOLBAR BUTTONS.....	46
8.18.	MOVING OR COPYING WORKFRAMES AND THOUGHTFRAMES USING DRAG AND DROP IN THE MODEL TREES	46
8.19.	ADDING FRAME ELEMENTS TO WORKFRAMES AND THOUGHTFRAMES USING THE KEYBOARD SHORTCUTS	46
8.20.	ADDING ACTIVITY-REFS TO WORKFRAMES USING DRAG AND DROP.....	48
9.	THE COMPOSER APPLICATION DESKTOP AND SOURCE CODE EDITOR	49
9.1.	OPENING SOURCE FILES IN THE COMPOSER APPLICATION	50
9.2.	EDITING SOURCE FILES IN THE COMPOSER APPLICATION	50
9.3.	USING THE FIND AND REPLACE DIALOG.....	51
10.	USING THE PROJECT EXPLORER	53
10.1.	CREATING A NEW PACKAGE FOLDER IN THE PROJECT EXPLORER.....	54
10.2.	CREATING A NEW SOURCE FILE USING THE PROJECT EXPLORER.....	55
10.3.	CHANGING THE PACKAGE OF A SOURCE FILE USING DRAG AND DROP	56
11.	COMPILING YOUR MODEL AND USING THE TASK LIST	56
12.	RUNNING YOUR MODEL IN THE COMPOSER APPLICATION	58

Table of Figures

Figure 1.1 'Open Brahms Model' dialog window 6

Figure 2.1 New Model Dialog 7

Figure 2.2 New Model in Composer Application Window 8

Figure 3.1 'Open Existing Model' Dialog..... 9

Figure 3.2 Existing Model Opened in Composer Application..... 10

Figure 4.1 'Import Model' Dialog..... 11

Figure 4.2 Imported Model Opened in Composer Application 12

Figure 7.1 'Model' Menu 19

Figure 7.2 The Concept Editor for an Object..... 21

Figure 7.3 The Path Editor..... 26

Figure 8.1 New attribute ready for editing. 28

Figure 8.2 Creating a new attribute using drag and drop 30

Figure 8.3 New initial statement ready for editing. 33

Figure 8.4 Creating an initial statement using drag and drop 35

Figure 8.5 New primitive activity ready for editing the name. 37

Figure 8.6 The Activity Editor for a create_object activity 40

Figure 8.7 New parameter presented for editing..... 42

Figure 8.8 New workframe presented for editing in the Agent Model Tree..... 44

Figure 8.9 Editing a detectable body using the tree cell editor 48

Figure 9.1 The Composer desktop showing multiple, cascaded source code windows 49

Figure 9.2 The Find and Replace Dialog 51

Figure 10.1 The Project Explorer..... 53

Figure 10.2 The 'Add Package Folder' Dialog 54

Figure 10.3 The Project Explorer 'Add Concept' Dialog..... 55

Figure 10.4 Text of New Concept Created in Project Explorer 56

Figure 11.1 The Task List showing 2 compiler errors 57

Figure 11.2 Task List showing 1 parser error and 2 compiler errors..... 58

1. Overview of the Composer Application

The Brahms Composer is an Integrated Development Environment (IDE) for creating, editing, compiling and running Brahms models. The Composer application is included in the software packages for both Brahms Personal Agent and Brahms Professional Agent that are available for download.

The Composer application has features for creating and editing all the elements of a Brahms model. There are a variety of features for accomplishing the same goals, giving the modeler a choice between creating and editing elements visually, by writing source code, or, by using a combination of both approaches.

2. Scope of this document

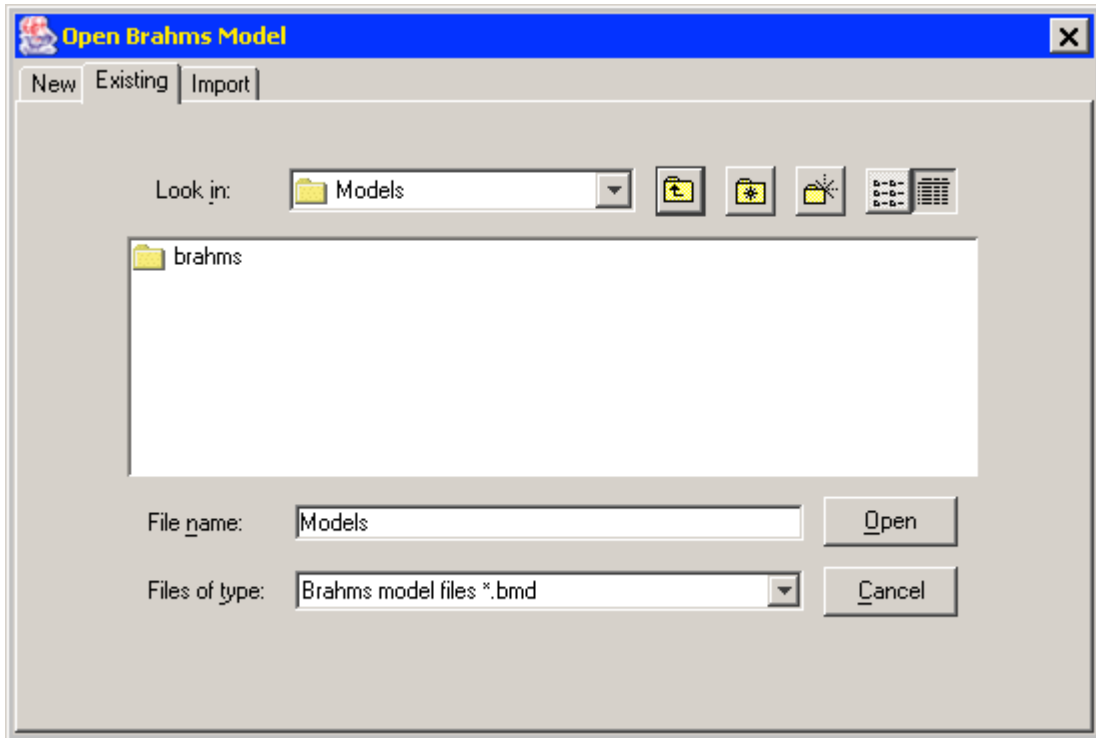
This user guide describes the features of the Composer Application and explains how a user can create new Brahms models. This document assumes familiarity with the Brahms language specification, it does not explain the Brahms language elements.

3. Launching the Composer Application

On the Windows platform select the 'Composer' application item under the Brahms Personal or Professional Agent Program Group in the Start menu. On other platforms double-click the Composer application icon in the 'bin' directory under 'ProfessionalAgent' or 'PersonalAgent' where the Brahms software was installed.

When the Composer application starts up you are presented with a dialog window allowing you to choose among tabs for opening an existing model, creating a new model or importing an existing model. The default view is the 'Existing' tab.

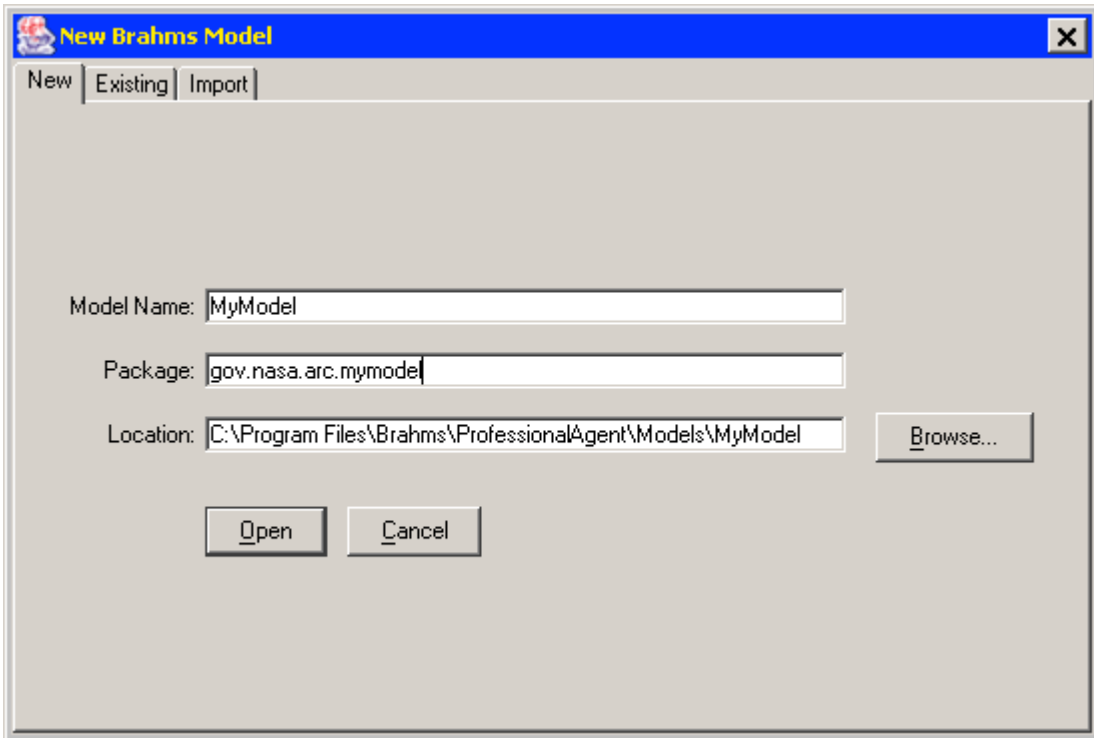
Figure 1.1 'Open Brahms Model' dialog window



4. Creating a New Model

Select the 'New' tab from the 'Open Brahms Model' dialog window. You will be presented with the new model dialog.

Figure 2.1 New Model Dialog



Enter a name for the new model in the 'Model Name' field. If you wish your model to have a package then enter the name of the package.

The default location for new models is in a root directory having the same name as the model and located within the 'Models' directory where the Brahms software was installed. You may, however, enter any location and name you wish for the root directory. You can select the directory where the model is to be created by using the Browse button.

When you press the 'Open' button a Brahms source file for the model will be created. In the example in figure 2. this file will be named 'MyModel.b' and it will be located in the directory:

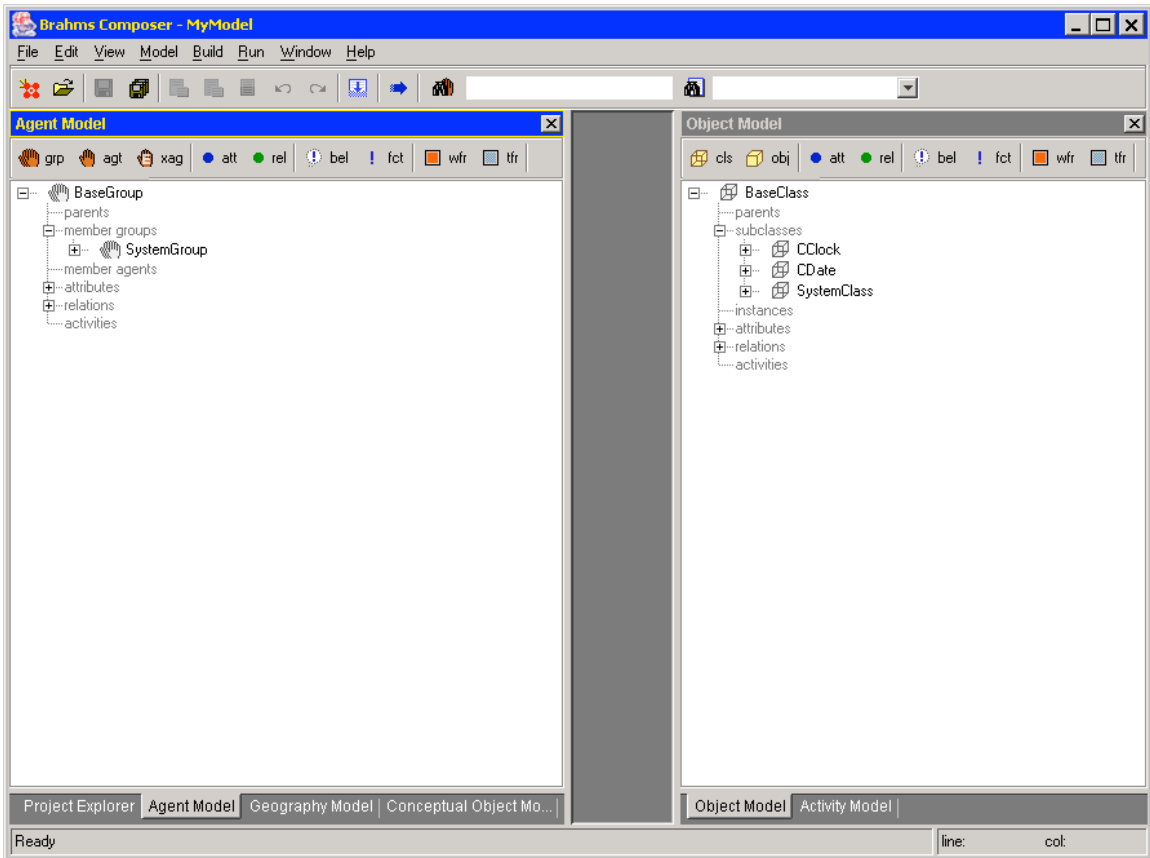
'C:\ProgramFiles\Brahms\ProfessionalAgent\Models\MyModel\gov\nasa\arc\mymodel'.

The model file created by the Composer application will contain the following text:

```
package gov.nasa.arc.mymodel;  
  
import gov.nasa.arc.mymodel.*;
```

The new model will be opened in the Composer application window.

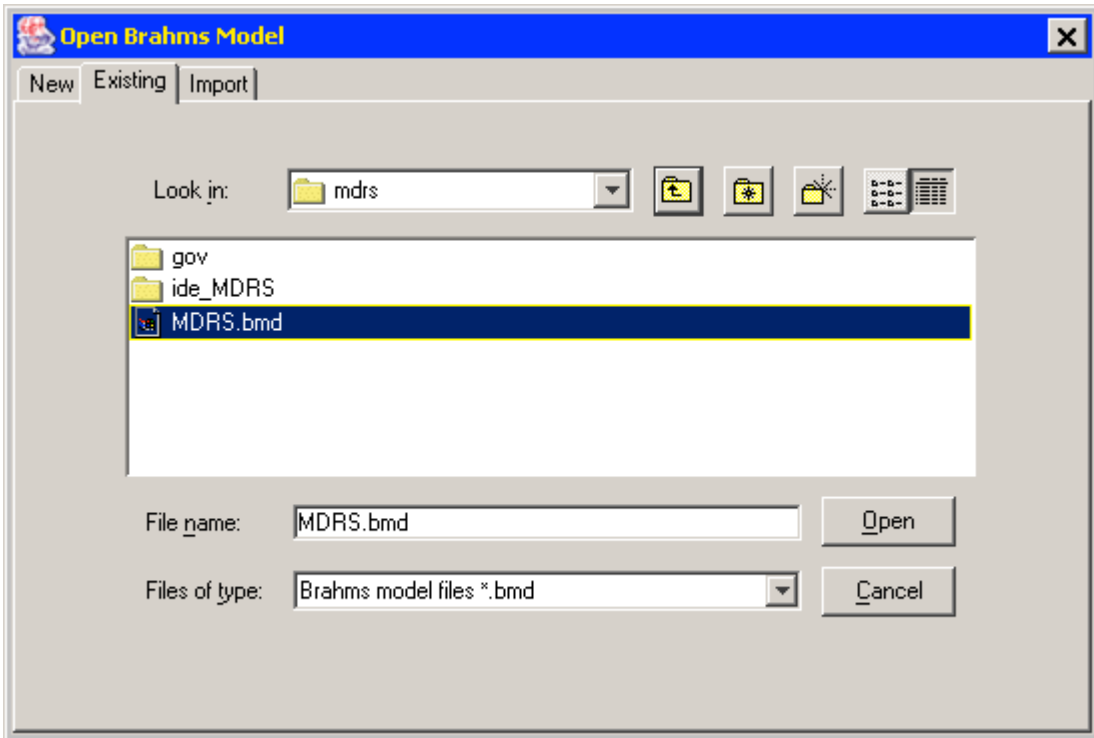
Figure 2.2 New Model in Composer Application Window



5. Opening an Existing Model

Select the 'Existing' tab from the 'Open Brahms Model' dialog window. You will be presented with the open existing model dialog.

Figure 3.1 'Open Existing Model' Dialog

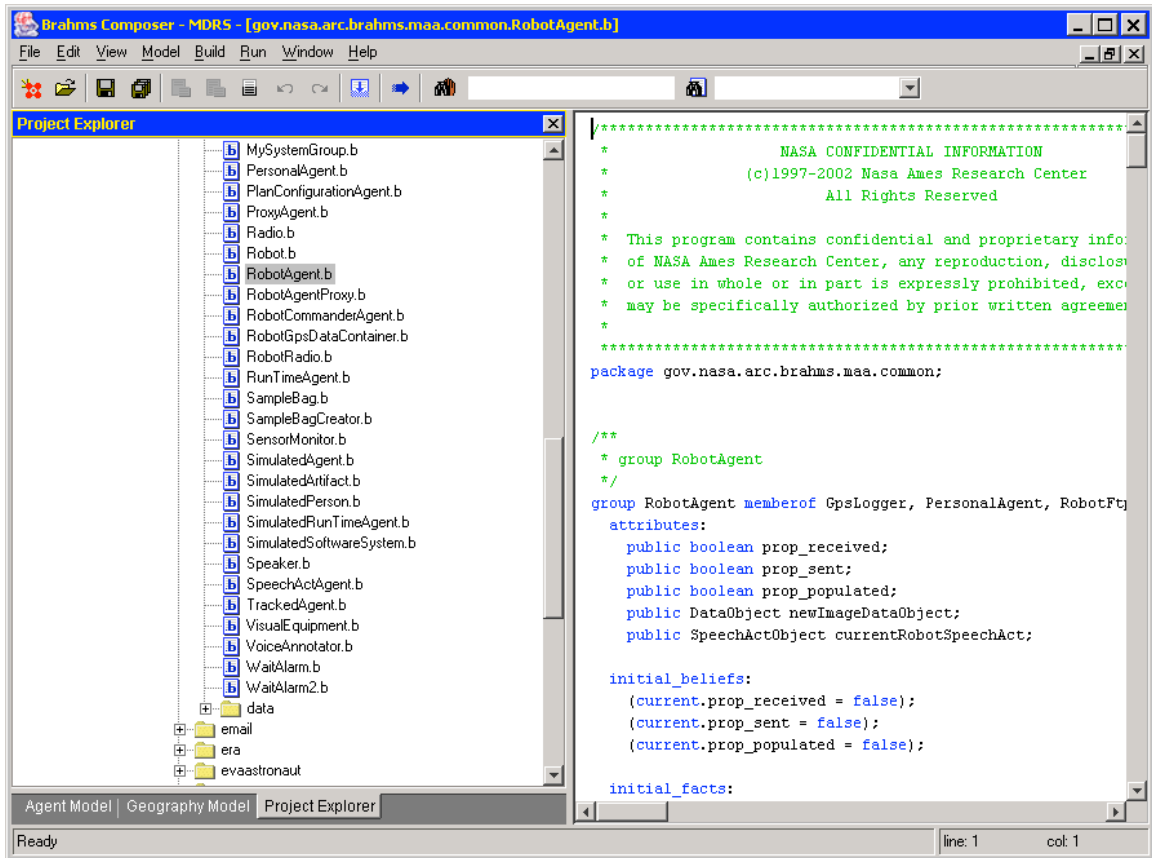


Existing models created in the Composer application will have a '.bmd' file extension. Navigate to the '.bmd' file for the model you wish to open, select it and press the 'Open Button'.

The model will be opened in the Composer application window and if none of its files were modified outside the application since it was last saved it will appear with the same layout configuration as when it was last saved in the Composer.

If the Composer encounters any syntax errors in the source code while opening a model those errors are presented in the 'Task List' pane in the application window. (See section 11 for a discussion of the Task List).

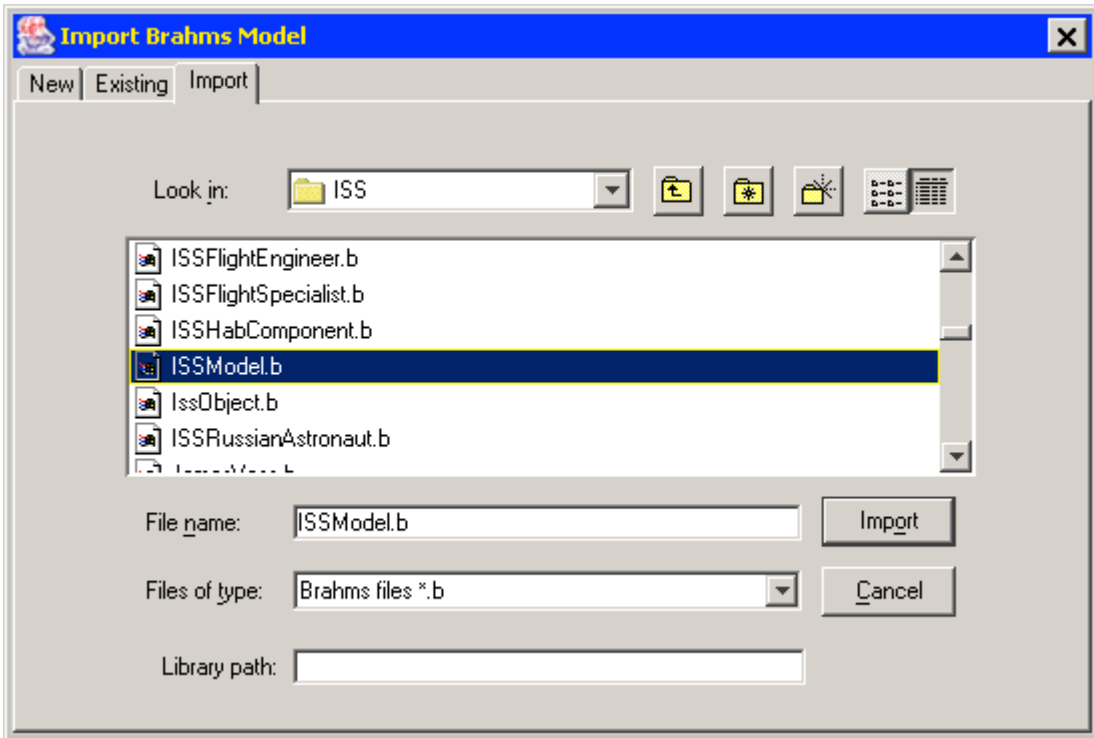
Figure 3.2 Existing Model Opened in Composer Application



6. Importing an Existing Model

Select the 'Import' tab from the 'Open Brahms Model' dialog window. You will be presented with the 'Import Model' dialog.

Figure 4.1 'Import Model' Dialog

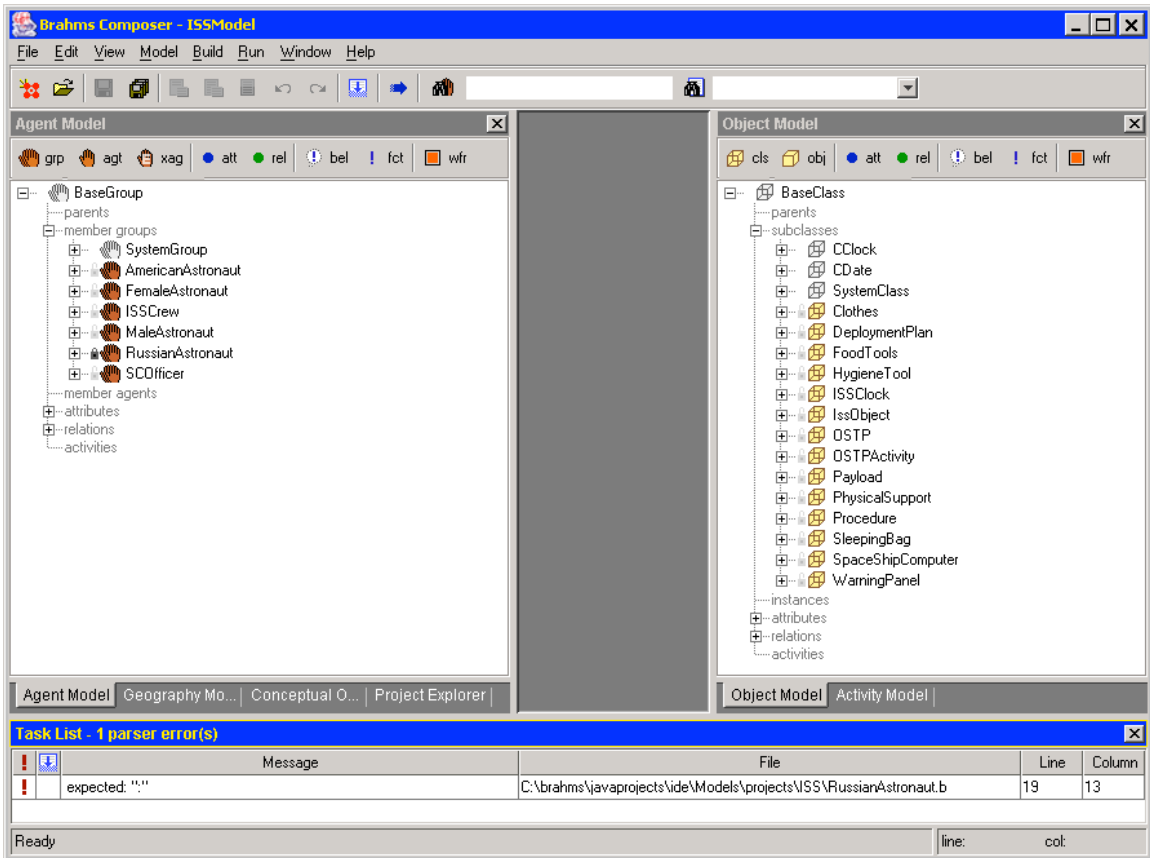


To import an existing model you must navigate to and select the model file (the main .b file for the model) for that model and press the 'Import' button. The model file for a model contains an optional package statement and an import statement for each package in the model. The Composer imports an existing model by parsing all the Brahms source files that make up the model as specified in the selected model file. The composer deduces the library path for the model based on the package statement in the model file and always includes the base model in the library path. If your model references any Brahms source files that are not in the same library path as the model you import you will need to specify that library path.

Imported models are opened in the Composer application window in the same default configuration as new models.

If the Composer encounters any syntax errors in the source code while importing a model those errors are presented in the Task List pane in the application window. (See section 11 for a discussion of the Task List).

Figure 4.2 Imported Model Opened in Composer Application



7. The Layout of the Composer Application Window

The application is laid out as a set of dockable panes that can be docked in any of eight positions within the application window: North, South, East, West, Northeast, Northwest, Southeast and Southwest. The dockable panes can be opened and closed and moved by the modeler to suit his/her preferences. Once a pane is closed it can be reopened from the 'View' menu in the application menu bar.

A dockable pane can be moved to a different docking location by pressing and holding down the left mouse button on the docked pane's title bar and dragging it to one of the sides of the application window. A valid docking position will be indicated by a rectangle indicating the docking position. Releasing the mouse button will undock the pane from its previous position and dock it in its new position.

Multiple dockable panes can be docked in the same location by dragging the mouse over the title bar of the dockable pane in the same location where you would like to dock the other window. An outline will be displayed inside of the already docked pane indicating the merge. When the mouse is released a new tab will be added to the docked pane with the name of the merged dockable pane. Selecting the tabs allows you to switch between the dockable panes.

To dock a dockable pane in a northeast, northwest position you will need to drag the mouse just under the title bar of the already docked pane. An outline will be displayed showing the docked position when the mouse is released. To dock a dockable pane in a southeast or southwest position you will need to drag the mouse just above the tab area (if there is one) or just above the status bar. An outline will also then be presented with the docked position when the mouse is released.

It is possible to reorder the tabs in a docked pane by dragging a tab to the desired position within the docked pane. A dockable pane that is represented as a tab can be docked into a different position by either dragging its title bar or by dragging its tab to the desired docking location.

The close button 'x' can be used to close the currently active docked pane (if multiple dockable panes are merged in the same location). If no other dockable panes are docked in the same location then the close button will close the entire docked pane. The dockable panes can be re-opened through the View menu.

The Center area of the application window is a multi-document desktop used for displaying and editing source files.

Nine Dockable Panes

In version 1.3 of the application there are nine dockable panes. Five of them are visual designers for creating and editing Brahms model elements.

The Composer application represents a Brahms model as five individual models, four Concept Models and One Activity Model, each in its own pane. The five models that make up a Brahms model are:

- The Agent Model
- The Object Model
- The Geography Model
- The Conceptual Object Model
- The Activity Model

Each concept model is presented as a tree structure with the inheritance hierarchy of the concepts in the model reflected in the tree hierarchy.

Agent Model Pane

The Agent Model pane gives a visual representation of all the Groups and Agents in the model and their inheritance relations. It also presents all the sub-elements of each concept. All features of the Agent Model can be created and edited visually within the Agent Model pane.

Object Model Pane

The Object Model pane gives a visual representation of all the Classes and Objects in the model and their inheritance relations as well as the conceptual-object-membership

relations of Objects. It also presents all the sub-elements of each concept. All features of the Object Model can be created and edited visually within the Object Model pane.

Geography Model Pane

The Geography Model pane gives a visual representation of all the AreaDefs and Areas in the model and their inheritance relations as well as all Paths between Areas. It gives a representation of the aggregate relations among Areas. It also presents all the sub-elements of each concept. All features of the Geography Model can be created and edited visually within the Geography Model pane.

Conceptual Object Model Pane

The Conceptual Object Model pane gives a visual representation of all the ConceptualClasses and ConceptualObjects in the model and their inheritance relations as well as the conceptual-object-membership relations of Objects and Conceptual Objects. It also presents all the sub-elements of each concept. All features of the Conceptual Object Model can be created and edited visually within the Conceptual Object Model pane.

Activity Model Pane

The Activity Model pane gives a visual representation of all the Active Concepts in the model that contain activities. It allows new activities to be created and edited visually within the Activity Model Pane.

Project Explorer Pane

The Project Explorer pane gives a visual representation of the file and package structure of the model. It allows new files and packages to be created. It also allows the package structure of the model to be edited.

Task List Pane

Syntax errors and compiler errors are presented in a pane entitled 'Task List'.

Find and Replace Pane

The Find and Replace pane contains tools for searching for and replacing text in the source files that make up the model.

Output Pane

The Output Pane is displayed when a model is run in the Brahms virtual machine from within the Composer. It displays the Brahms virtual machine's console output, i.e. log messages.

8. Creating and Editing Concepts Using the Visual Designers

Concept Nodes

The Composer application represents all concepts in a Brahms Model as tree nodes in one of the four Concept Model Trees – the Agent Model Tree, the Object Model Tree, the Geography Model Tree or the Conceptual Object Model Tree.

Each concept node is created with a set of sub-nodes that either represent its relations with other concepts or that list the model elements defined for the concept.

Concept Sub-nodes

All concept nodes are created with the following sub-nodes:

- **parent(s)** - The parent(s) node lists the parent or parents of the concept.
- **attributes** – The attributes node lists all attributes defined for the concept as well as all attributes inherited from the concept’s ancestors.
- **relations** – The relations node lists all relations defined for the concept as well as all relations inherited from the concept’s ancestors.

All active concept nodes (groups, agents, classes, and objects) are created with the following additional sub-nodes:

- **initial facts and beliefs** - The initial facts and beliefs node lists all initial facts and initial beliefs defined for the active concept as well as all initial facts and beliefs inherited from the active concept’s ancestors.
- **activities** - The activities node lists all activities defined for the active concept.
- **workframes** - The workframes node lists all workframes defined for the active concept.
- **thoughtframes** - The thoughtframes node lists all thoughtframes defined for the active concept.

All Group nodes are created with the following additional sub-nodes:

- **member groups** – The member groups node lists all the Groups that are members of this Group.
- **member agents** – The member agents node lists all the Agents that are members of this Group.

All class-type concept nodes (Class nodes, ConceptualClass nodes, and AreaDef nodes) are created with the following additional sub-nodes:

- **subclasses** – The subclasses node lists all class-type concepts that extend this class-type concept.
- **instances** - The instances node lists all instance-type concepts that are instances of this class-type concept.

All Object nodes are created with the following additional sub-node:

- **partof** – The part of node lists all ConceptualObjects that this object is part of.

All Conceptual Object nodes are created with the following additional sub-nodes:

- **partof** – The partof node lists all ConceptualObjects that this ConceptualObject is part of.
- **parts** – The parts node lists all Objects and ConceptualObjects that are part of this ConceptualObject.

All AreaDef and Area nodes are created with the following additional sub-node:

- **initial facts** – The initial facts node lists all initial facts for this AreaDef or Area as well as all initial facts inherited from the concept’s ancestors.

All Area nodes are created with the following additional sub-nodes:

- **partof** – The part of node lists all Areas that this Area is part of.
- **parts** – The parts node lists all Areas that are part of this Area.
- **paths** – The paths node lists all Paths for which this area is a terminus.
- **inhabitants** – The inhabitants node lists all Agents and Objects that are located in this Area.

8.1. Creating Concepts Using the 3-Character Keyboard Shortcuts

Brahms concepts for a given Concept Model can be created in the Composer application by typing a 3-character keyboard shortcut while navigating within the Concept Model Tree.

The three-character codes for each concept type are:

- Group ‘grp’
- Agent ‘agt’
- ExternalAgent ‘xag’
- Class ‘cls’
- Object ‘obj’
- AreaDef ‘adf’
- Area ‘are’
- ConceptualClass ‘coc’
- ConceptualObject ‘cob’

The three character shortcuts are printed as the button text for the create concept buttons in the toolbar of each model pane. (see 7.2 below)

If no tree node is selected in the Model Tree when the three characters are typed then the new concept will be created as a child of the base library concept for that particular model. For example, if you type ‘grp’ in the Agent Model tree while no nodes are selected, then a new group will be created as a child of the library concept ‘BaseGroup’.

If a node in the tree is selected when the three characters are typed then the Composer application will find the first valid parent concept up the hierarchy from the selected node and will create the new concept as a child of that parent concept.

Note that ExternalAgents are created at the top level of the Agent Model tree since they have no parents.

The newly created concept will be presented as a tree node with a generic concept name ready for editing so that you may immediately give the new concept a name. When you have entered a valid name and pressed the ENTER key a file for the new concept will be created with the name '<concept_name>.b'. The new file will have the same package statement as the main model file and will be located in the file system in the same directory as the main model file. The new file will contain an empty definition for the new concept. The new concept will have the default properties as defined in the Brahms Language Specification.

All concept properties can be edited as described below in this user guide.

8.2. Creating Concepts Using the Model Pane Toolbar Buttons

Each of the panes for the Agent Model, the Object Model, the Geography Model and the Conceptual Object Model contains a toolbar at the top of the pane with buttons for creating concepts for that model.

If no tree node is selected in the model tree when a new concept button is pressed then the new concept will be created as a child of the base library concept for that particular model. For instance, if you press the 'Create New Group' button in the Agent Model pane toolbar while no nodes are selected in the model tree then a new group will be created as a child of the library concept 'BaseGroup'.

If a node in the tree is selected when the button is pressed then the Composer application will find the first valid parent concept up the hierarchy from the selected node and will create the new concept as a child of that parent concept.

Note that ExternalAgents are created at the top level of the Agent Model tree since they have no parents.

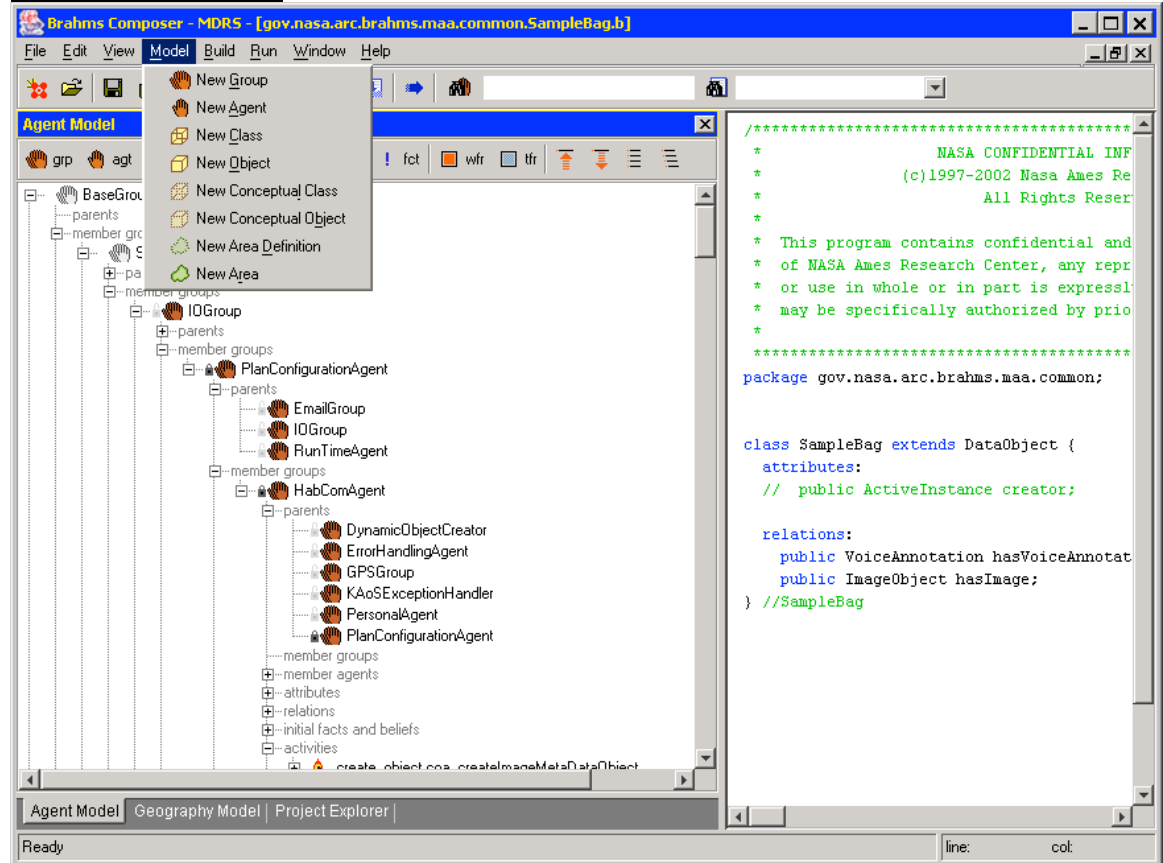
The newly created concept will be presented as a tree node with a generic concept name ready for editing so that you may immediately give the new concept a name. When you have entered a valid name and pressed the ENTER key a file for the new concept will be created with the name '<concept_name>.b'. The new file will have the same package statement as the main model file and will be located in the file system in the same directory as the main model file. The new file will contain an empty concept definition for the new concept. The new concept will have the default properties as defined in the Brahms Language Specification.

All concept properties can be edited as described below in this user guide.

8.3. Creating Concepts Using the ‘Model’ Menu

A new concept of any type can be created in the appropriate model by selecting the menu item for the desired concept type from the ‘Model’ menu in the application menu bar.

Figure 7.1 ‘Model’ Menu



When a new concept item is selected from the ‘Model’ menu the Model Tree pane for the model that will contain the concept will be docked if it is not currently docked and brought to the front of the application window.

If no tree node is selected in the appropriate model tree when a new concept menu item is selected then the new concept will be created as a child of the base library concept for that particular model. For instance, if you select the ‘New Group’ menu item while no nodes are selected in the Agent Model tree, then a new group will be created as a child of the library concept ‘BaseGroup’.

If a node in the tree is selected when the menu item is selected then the Composer application will find the first valid parent concept up the hierarchy from the selected node and will create the new concept as a child of that parent concept.

Note that ExternalAgents are created at the top level of the Agent Model tree since they have no parents.

The newly created concept will be presented as a tree node with a generic concept name ready for editing so that you may immediately give the new concept a name. When you have entered a valid name and pressed the ENTER key a file for the new concept will be created with the name '<concept_name>.b'. The new file will have the same package statement as the main model file and will be located in the file system in the same directory as the main model file. The new file will contain an empty concept definition for the new concept. The new concept will have the default properties as defined in the Brahms Language Specification.

All concept properties can be edited as described below in this user guide.

8.4. Finding Concept Nodes in the Visual Designer Model Trees

The Find Concept Toolbar Button

The Composer application toolbar contains a 'Find Concept' button and text field. When the cursor is placed in the text field a pop-up list of all concepts in the model will appear. As you enter text into the field the list will be scrolled to select the first concept in the list that matches the text being entered (Brahms concept names are case-sensitive). When the concept you are searching for is selected you may press the ENTER key and the first tree node representing that concept on the model's inheritance hierarchy will be located and selected in the appropriate Model Tree. You may also scroll the list manually or using the arrow keys and double-click the concept you wish to locate in the Model Trees.

Double-Clicking Unpopulated Concept Nodes

The Model Trees contain some concept nodes that have no sub-nodes such as the nodes representing 'parent' concepts or 'inhabitant' concepts. Double-clicking one of the unpopulated concept nodes will locate and select the first populated node representing the same concept in the model's inheritance hierarchy. For example, if you double click a tree node representing an Object 'inhabitant' of an Area in the Geography Model Tree then the Composer application will find the tree node highest up in the inheritance hierarchy representing that Object in the Object Model Tree.

Using the Model Pane Toolbar's 'Tree View Mode' Buttons

The Model Panes for the Agent Model, the Object Model, the Geography Model and the Conceptual Object Model contain toolbar buttons for toggling the Model Tree view mode between 'Hierarchical' and 'Flat'. 'Hierarchical' is the default view for new models.

In the 'Hierarchical' view the tree structure represents the inheritance structure of the concepts in the model, with oldest ancestors furthest out in the tree structure

and youngest children deepest into the tree structure. Toggling the view to ‘Flat’ will present all concepts in the model listed in alphabetical order at the top level of the tree. The descendants of each concept can then be found below its top level node. Toggling to the ‘Flat’ view offers a quick way to find a concept node in a Model Tree.

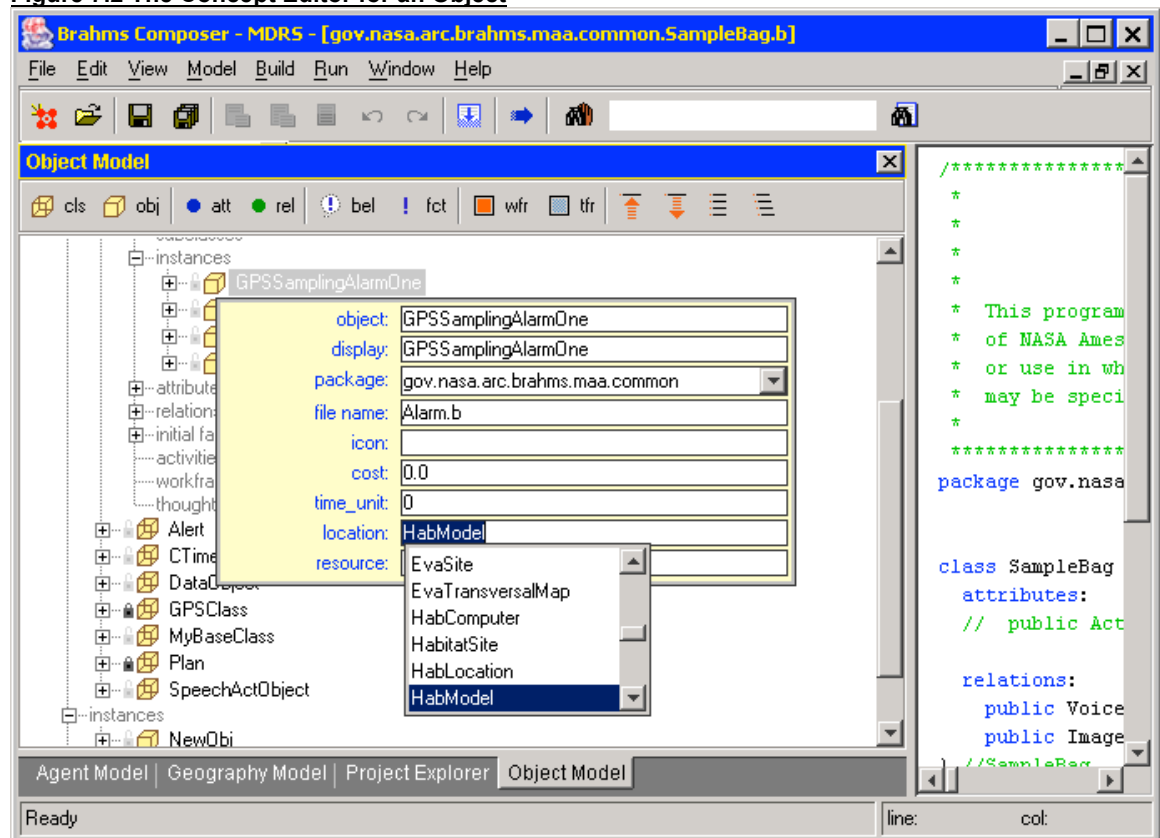
8.5. Editing Concepts Using the Concept Editor

Concept properties can be edited using the Concept Editor from within the Model Tree. Selecting a concept node in a Model Tree and pressing CTRL-E will open the concept editor panel positioned beside the selected node.

The Concept Editor can be used to edit the concept’s name, the source file where it is located, the source file’s package and the built-in properties defined for the particular concept type being edited.

Note that when you edit a concept’s name using the Concept Editor, all references to that name in any nodes in the model trees are updated to reflect the change. Also, all source code files that contain references to that concept are auto-edited to reflect the change (provided the source code file is not open on the desktop).

Figure 7.2 The Concept Editor for an Object



When editing the concept's name in the Concept Editor the 'display' property field and the file name field will automatically change to match the new concept name unless the 'display' property or file name have been explicitly edited to something different from the concept name.

As shown in figure 7.2, placing the cursor in some fields in the Concept Editor will pop-up a list of choices that are valid values for that field. In the example in figure 7.2 a list of all Areas in the model has been presented for editing the 'location' property for an Object. Double-clicking an item in the list will set the text in the field being edited to the selected item. Also, selecting an item in the list using the mouse or arrow keys and then tabbing to the next field in the Concept Editor will set the text in the field to the selected item.

Pressing the ENTER key will save the edits made in the Concept Editor and close the panel. Pressing the ESCAPE key or clicking outside the panel will close the editor panel without saving any edits.

All edits made and saved in the Concept Editor are automatically written to the source code file where the concept is defined. If you edit the file name or package for the file then the application will move the file in the file system to the new location. If you edit the file name and package to match a file that already exists in the model then the source code for the concept being edited will be cut from its present file and appended to the end of the newly selected file. The original file will not be deleted automatically. If you wish to delete the file you must use the Project Explorer.

NOTE: You cannot open a Concept Editor for a concept whose source file is open on the desktop. In this case you will see a 'lock' icon next to the concept icon in the tree node. You must first close the source file before attempting to edit the concept using the Concept Editor.

8.6. Editing the Inheritance Relations of Concepts Using Drag and Drop and Delete

The inheritance relations among concepts can be edited by dragging and dropping concept nodes in the Model Trees onto one another.

For example, dragging an Agent node and dropping it onto the node representing a Group that is not the Agent's current parent will move the Agent node and all its sub-nodes out of the 'member agents' node of the Agent's current parent and into the 'member agents' node of the target parent. The source code file where the Agent is defined will be auto-edited to reflect the new inheritance for the Agent.

For concepts that allow multiple inheritance such as Groups and Agents, using CTRL-drag to move a child node onto a new parent will copy the child node and all its sub-nodes into the 'children' node of the new parent. The source code file for the child concept will have its inheritance statement edited to add the new parent to the list of parents that the child inherits from.

To remove one of multiple inheritance relations you simply select the node representing the child concept you wish to remove in the 'children' node of the parent concept you wish to remove it from and press the DELETE key. Alternately, you could select the node representing the parent concept in the 'parents' node of the child concept you wish to remove and press the DELETE key. The source code for the child concept will be auto-edited to remove the inheritance statement.

Attempting to create invalid inheritance relations will result in an error message and no edits will take place.

NOTE: You cannot drag and drop nodes for concepts whose source files are open on the desktop. In this case you will see a 'lock' icon next to the concept icon in the tree node. You must first close the source file before attempting to edit the inheritance relations using drag and drop.

8.7. Editing the 'partof' relation for Objects and ConceptualObjects Using Drag and Drop and Delete

The 'partof' relation for Objects and ConceptualObjects can be edited by dragging and dropping an Object node or ConceptualObject node in the Model Trees onto another ConceptualObject node or onto the 'parts' node of another ConceptualObject node. Alternately, A ConceptualObject node may be dragged and dropped onto the 'partof' node of an Object node or another ConceptualObject node.

For example, dragging an Object node from the Object Model Tree and dropping it onto the node representing a ConceptualObject in the Conceptual Object Model Tree will place a copy of the Object node into the 'parts' node of the ConceptualObject node. If the Object was currently 'partof' another ConceptualObject then that relation will be removed and the Object node copy in that former ConceptualObject node's 'parts' node will be deleted. The source code file where the Object is defined will be auto-edited to change the 'partof' statement defining the relation.

Dropping the Object node onto the 'parts' node of the ConceptualObject node would achieve the same result. Dragging the ConceptualObject node and dropping it onto the 'partof' node of the Object node would also achieve the same result.

In order to make an Object or a ConceptualObject partof multiple ConceptualObjects hold down the CTRL key while dragging and dropping.

To remove a 'partof' relation you simply select the node representing the Object or ConceptualObject you wish to remove in the 'parts' node of the ConceptualObject you wish to remove it from and press the DELETE key. Alternately, you could select the node representing the ConceptualObject in the 'partof' node of the Object or ConceptualObject you wish to remove and press the DELETE key. The source code for the Object or ConceptualObject will be auto-edited to remove the partof statement for the deleted 'partof' relation.

Attempting to create invalid 'partof' relations will result in an error message and no edits will take place.

NOTE: You cannot drag and drop or delete nodes for concepts whose source files are open on the desktop. When the file is open you will see a 'lock' icon next to the concept icon in the tree node. You must first close the source file before attempting to edit a concepts in the model trees.

8.8. Editing the 'partof' Relation for Areas Using Drag and Drop and Delete

The 'partof' relation for Areas can be edited by dragging and dropping one Area node onto another Area node or onto the 'parts' node of another Area node in the Geography Model Tree. This will result in the dragged Area becoming part of the Area it was dropped onto. Alternately, an Area node may be dragged and dropped onto the 'partof' node of an Area node. This will result in the dragged Area becoming the containing Area for the Area it was dropped onto.

A copy of the containing Area's node will be created in the 'partof' node of the part Area and a copy of the part Area's node will be created in the 'parts' node of the containing Area.

If the dragged Area is currently 'partof' another Area then that relation will be removed and the copy of the dragged Area's node in the original containing Area's 'parts' node will be deleted.

The source code file where the part Area is defined will be auto-edited to change the 'partof' statement defining the relation.

To remove a 'partof' relation for an Area you simply select the node representing the part Area you wish to remove in the 'parts' node of the containing Area you wish to remove it from and press the DELETE key. Alternately, you could select the node representing the containing Area in the 'partof' node of the part Area

you wish to remove and press the DELETE key. The source code for the part Area will be auto-edited to remove the partof statement.

Attempting to create invalid partof relations will result in an error message and no edits will take place.

NOTE: You cannot drag and drop or delete nodes for concepts whose source files are open on the desktop. When the file is open you will see a 'lock' icon next to the concept icon in the tree node. You must first close the source file before attempting to edit a concepts in the model trees.

8.9. Editing the 'location' Property of Agents and Objects Using Drag and Drop and Delete

An Agent or Object can have its location property set to a specific Area by dragging the node representing the Agent from the Agent Model Tree or the node representing the Object from the Object Model Tree and dropping it onto an Area node or onto the 'inhabitants' node of an Area in the Geography Model Tree or by dragging the node representing the Area and dropping onto the node representing the Agent or Object you wish to locate in that Area.

This will result in a copy of the dragged node being created in the 'inhabitants' node of the Area onto which the dragged node was dropped. The source code for the dragged Agent or Object will be auto-edited to reflect the change in the location property.

Alternately, if the Agent or Object is currently located in an Area, you may drag the copy of the Agent or Object node from the 'inhabitants' node of that Area and drop it onto another Area node or onto the 'inhabitants' node of another Area. The copy of the dragged node will be moved from the original Area's 'inhabitants' node to the new Area's 'inhabitants' node. The source code for the dragged Agent or Object will be auto-edited to reflect the change in the location property.

Selecting an Agent or Object node from the 'inhabitants' node of an Area and pressing the DELETE key will delete the node and set the location property of the Agent or Object to the default value. The source code for the Agent or Object will be auto-edited to reflect the change in the location property.

8.10. Creating Paths Using Drag and Drop

A Path relating two Areas can be created in the Geography Model Tree by dragging the Area node representing one terminus of the Path and dropping it

onto the 'paths' node of the Area representing the other terminus of the Path. A node representing the new path will be created in the 'paths' nodes of both Areas.

The newly created Path node will be presented ready for editing so that you may edit the new Path's name and distance properties. When you have entered a valid name and pressed the ENTER key a file for the new Path will be created with the name '<path_name>.b'

The new file will have the same package statement as the main model file and will be located in the file system in the same package directory as the main model file. The new Path's properties can be further edited using the Path Editor as described below.

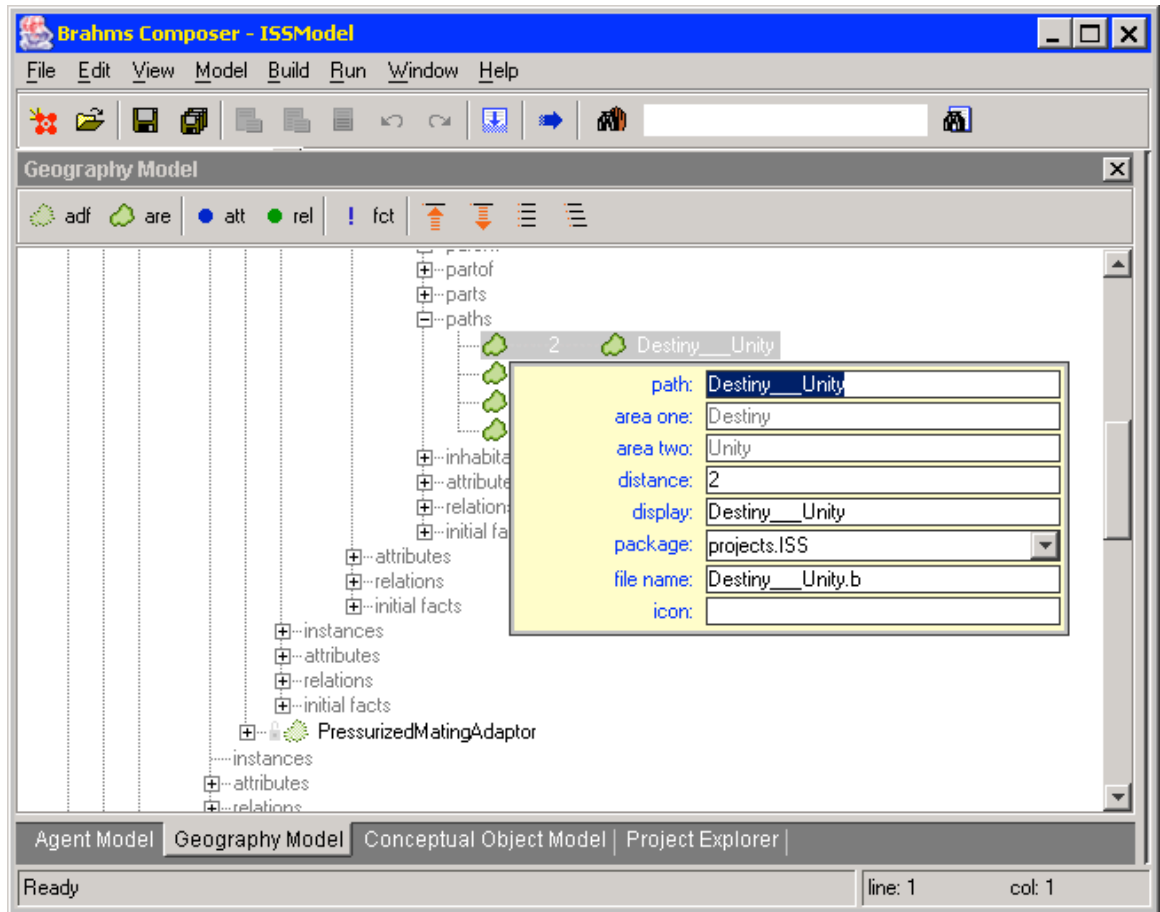
Selecting a Path node and pressing the DELETE key will delete that Path from the model. All nodes representing the Path will be deleted from the Geography Model Tree and the source code defining the Path will be deleted from the file where the Path is defined.

8.11. Editing Paths Using the Path Editor

Once a Path has been created its properties can be edited using the Path Editor from within the Geography Model Tree. Selecting a Path node and pressing CTRL-E will open the path editor panel positioned beside the selected node.

The Path Editor can be used to edit the Path's name, display name, distance, icon property, the source file where it is located and the source file's package.

Figure 7.3 The Path Editor



Pressing the ENTER key will save the edits made in the Path Editor and close the panel. Pressing the ESCAPE key or clicking outside the panel will close the editor panel without saving any edits.

All edits made and saved in the Path Editor are automatically written to the source code file where the Path is defined. If you edit the file name or package for the file then the application will move the file in the file system to the new location. If you edit the file name and package to match a file that already exists in the model then the source code for the Path being edited will be cut from its present file and appended to the end of the newly selected file. The original file will not be deleted automatically. If you wish to delete the file you must use the Project Explorer.

NOTE: You cannot open a Path Editor for a Path whose source file is open on the desktop. You must first close the source file before attempting to edit the Path using the Path Editor.

9. Creating and Editing Concept Elements Using the Visual Designers

All Brahms Model elements can be created and added to concepts using the Visual Designer Model Trees. Model elements are represented as sub-nodes of the concept nodes in the Model Trees.

The Composer application allows for rapid creation of model elements by typing 3-character keyboard shortcuts while navigation in the Visual Designer's Model Trees. Typing a 3-character keyboard shortcut creates a new element with default values and presents it as a tree node in the concept's tree structure ready for editing in a custom tree cell editor. Edits made to the model in this way are added incrementally to the source code file where the concept being edited is defined.

9.1. Adding Attributes and Relations to Concepts Using the 3-Character Keyboard Shortcuts

Attributes and relations can be created and added to concepts in the Composer application by typing a three-character abbreviation when a node is selected within the tree structure for that concept in its model tree.

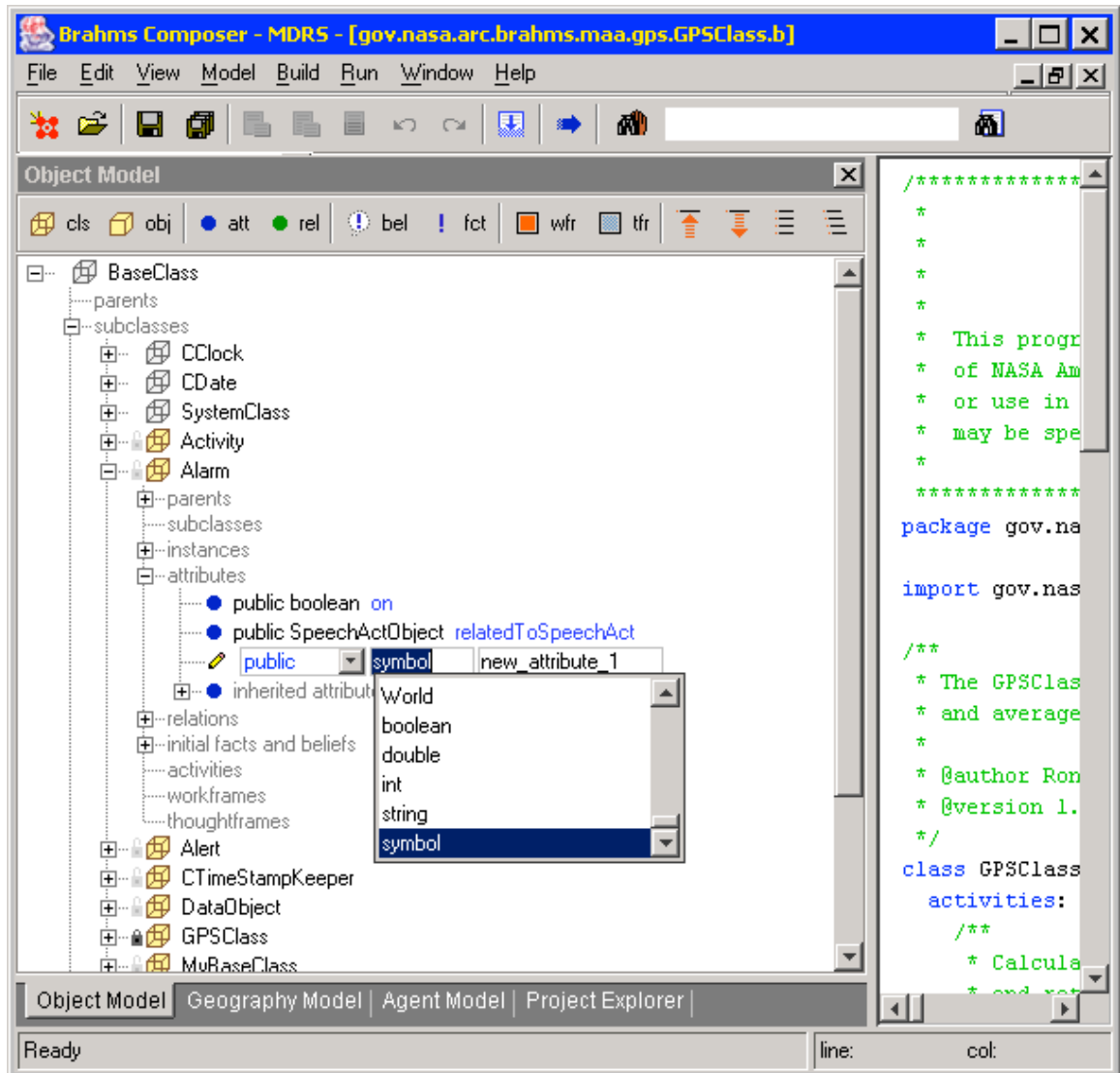
The 3-character abbreviations for attributes and relations are:

- attribute 'att'
- relation 'rel'

For example, if a Group node is selected in the Agent Model Tree and you type the three characters 'att', a new attribute will be created and will appear as a child node of the 'attributes' node of the selected Group. Typing 'rel' will result in a new relation being created as a child of the selected Group's 'relations' node.

The new attribute or relation will be presented in the model tree ready for editing.

Figure 8.1 New attribute ready for editing.



New attributes created in this way will have scope=public, type=symbol and name="new_attribute_n". New relation created in this way will have scope=public, type=Object and name="new_relation_n". All fields are editable using the tree cell editor.

When the editor is activated the 'type' field will have focus and a drop down list of all possible attribute or relation types is presented. This list will include all concepts that are valid attribute or relation types that are currently in the model.

Pressing the ENTER key while editing an attribute or relation will cause the edits to be saved. The source code file where the concept is defined will be auto-edited to include the new attribute or relation declaration.

Clicking the mouse outside the bounds of the attribute/relation tree cell editor or pressing the ESCAPE key will cause the edits to be canceled and the editor to close.

9.2. Creating Attributes and Relations Using the Model Pane Toolbar Buttons

Each of the panes for the Agent Model, the Object Model, the Geography Model and the Conceptual Object Model contains a toolbar at the top of the pane with buttons for creating attributes and relations for the concepts in that model.

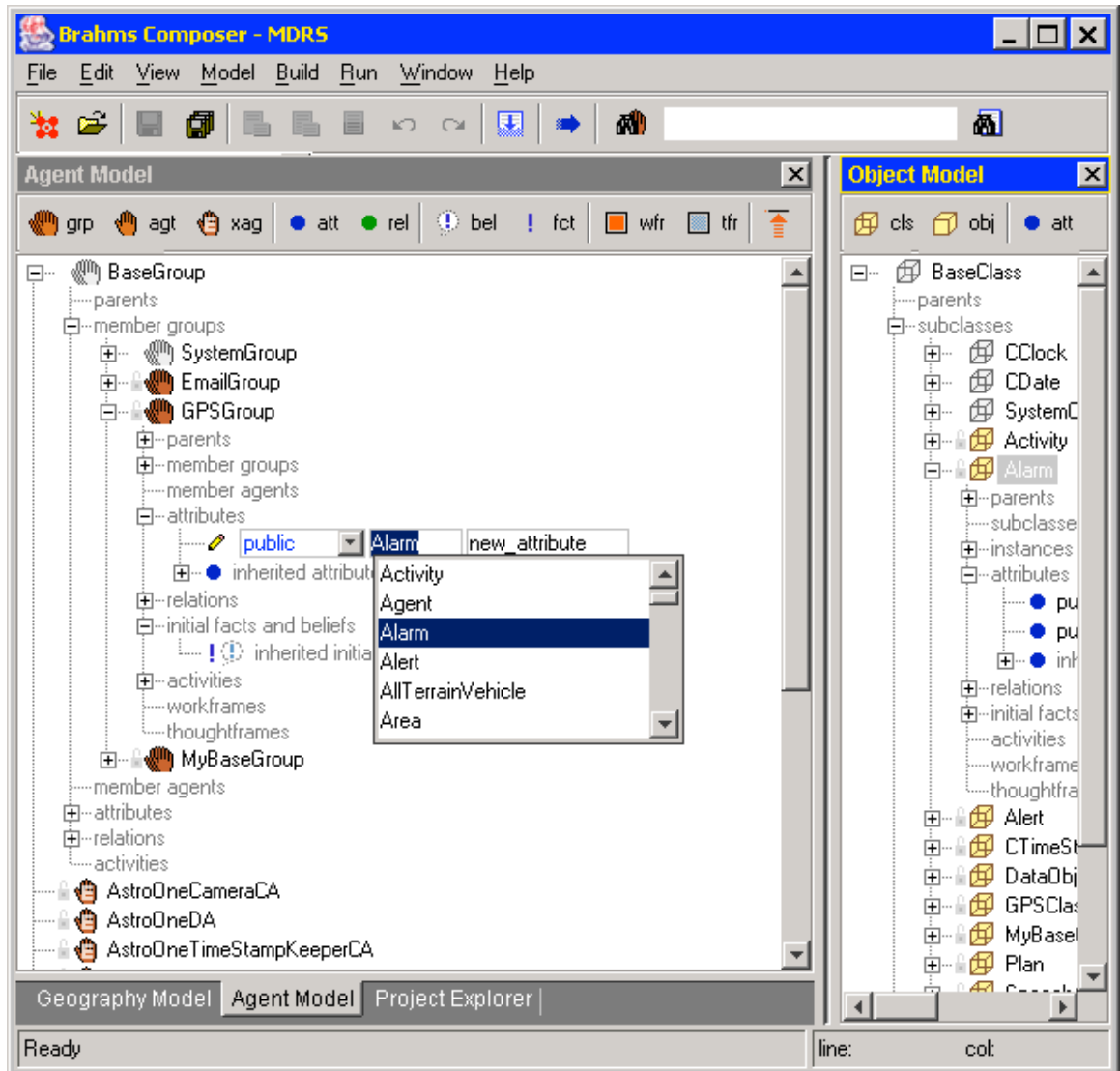
A node must be selected somewhere in the tree hierarchy for the concept for which you wish to add an attribute or relation. Pressing the ‘attribute’ toolbar button will cause a new attribute to be created and added to the specified concept and appear ready for editing in the tree. Pressing the ‘relation’ toolbar button will cause a new relation to be created and added to the specified concept and appear ready for editing in the tree. If no node is selected in the tree then pressing the ‘attribute’ or ‘relation’ button will have no effect.

9.3. Creating Attributes and Relations Using Drag and Drop in the Model Trees

A new attribute or relation can be created for a concept by dragging the node representing a valid type for an attribute and relation and dropping it onto the ‘attributes’ or ‘relations’ node of the concept to which you wish to add the new attribute or relation. A new attribute or relation having the type of the dragged concept will be created and added to the target concept.

For example, figure 8.2 shows the result of dragging the tree node representing the class ‘Alarm’ in the Object Model Tree and dropping it onto the ‘attributes’ node of the Group ‘GPSGroup’ in the Agent Model Tree.

Figure 8.2 Creating a new attribute using drag and drop



The new attribute “public Alarm new_attribute” is added to the group ‘GPSGroup’ and presented in the model tree ready for editing. You can now tab to the name field, enter the desired name for the new attribute and press ENTER.

The source code file where the concept ‘GPSGroup’ is defined will be auto-edited to add the new attribute.

9.4. Moving or Copying Attributes and Relations using Drag and Drop in the Model Trees

Any attribute or relation can be moved from one concept to another by dragging the node representing the attribute or relation and dropping it onto the ‘attributes’ or ‘relations’ node of the concept to which you wish to move the attribute or relation. The dragged attribute will be deleted from the original concept and added to the target concept. The source code files for both concepts will be auto-

edited to reflect the change. This provides a useful way to move attributes or relations up or down the inheritance hierarchy as you are designing your model.

An attribute or relation may be copied to another concept by dragging the attribute or relation with the CTRL key held down. This will result in an attribute of the same scope, type and name being added to the target concept while the original concept is left unchanged.

9.5. Editing Attributes and Relations using the Model Tree Cell Editors

Any attribute or relation can be edited by selecting the tree node representing the attribute or relation in the tree hierarchy and pressing CTRL-E. This will cause the tree cell editor to activate for that attribute or relation and you will be able to edit any of its fields. Pressing the ENTER key will keep the edits and cause the source code file where the attribute or relation is declared to be auto-edited to reflect the change.

NOTE: In a large model an attribute or relation may be referenced by name in the condition statements of many other model elements. Editing the name of an attribute or relation using the visual designers will cause every tree node in the model that references the edited element to be updated to reflect the name change. Also, every source file that contains an element with a reference to the edited attribute or relation will be auto-edited to reflect the change, providing the source code file is not open on the application desktop. When editing attributes or relations using the model trees it is a good idea to close all documents on the application desktop so that edits can be propagated throughout the model.

Attributes and relation can be deleted from a concept by selected the tree node representing the attribute or relation and pressing the DELETE key. The source code file for the specified concept will be auto-edited to have the attribute or relation declaration removed. No elements that reference the deleted attribute or relation will be changed. You must edit those elements individually.

9.6. Adding Initial Facts and Beliefs to Active Concepts Using the 3-Character Keyboard Shortcuts in the Model Trees

Initial facts and initial beliefs can be created and added to concepts in the Composer application by typing a three-character keyboard shortcut when a node is selected within the tree structure for any concept in the Agent Model Tree or Object Model Tree.

Initial facts can be created and added to concepts in the Geography Model Tree by typing a three-character keyboard shortcut when a node is selected within the tree structure for any AreaDef or Area.

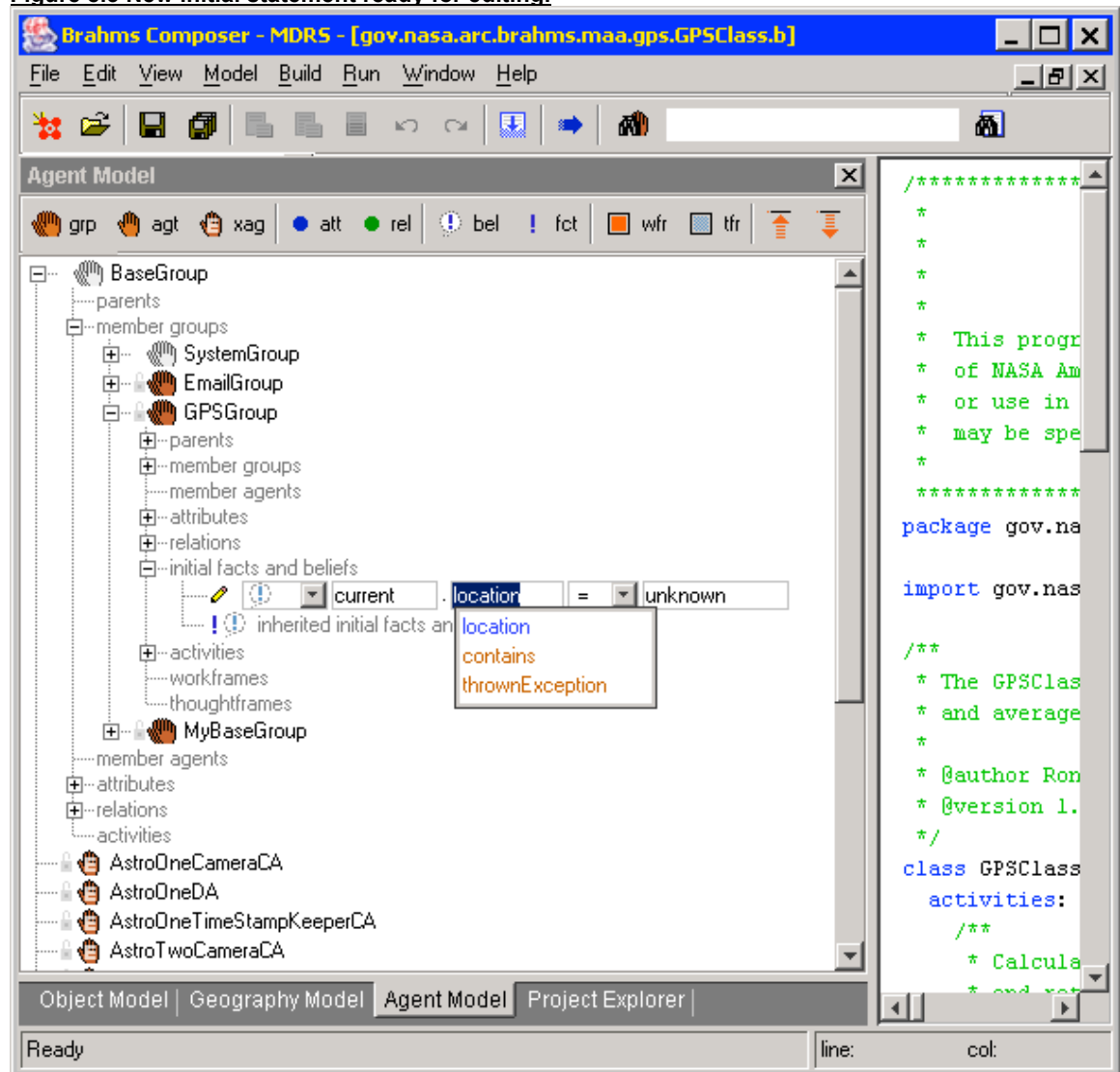
The 3-character shortcuts for initial facts and initial beliefs are:

- initial_fact 'fct'
- initial_belief 'bel'

For example, if a Group node is selected in the Agent Model Tree and you type the three characters 'bel', a new initial belief will be created and will appear as a child node of the 'initial statements' node of the selected Group. Typing 'fct' will result in a new initial fact being created as a child of the selected Group's 'initial statements' node.

The new initial fact or belief will be presented in the model tree ready for editing.

Figure 8.3 New initial statement ready for editing.



New initial beliefs and facts are created with the condition statement “current.location = unknown”. All fields are editable using the tree cell editor. When the editor is first presented the ‘attribute/relation’ field will have focus and a drop down list of all possible attribute or relations for ‘current’ is presented.

While editing, an initial statement can be specified as an initial belief or initial fact by selecting the appropriate icon for belief or fact in the tree cell editor.

Pressing the ENTER key while editing an initial statement will cause the edits to be saved. The source code file where the concept is defined will be auto-edited to include the new initial statement declaration.

Clicking the mouse outside the bounds of the initial statement tree cell editor or pressing the ESCAPE key will cause the edits to be canceled.

9.7. Creating Initial Facts and Beliefs Using the Model Pane Toolbar Buttons

The panes for the Agent Model and the Object Model contain a toolbar at the top of the pane with buttons for creating initial facts and initial beliefs for the concepts in that model. The pane for the Geography Model contains a toolbar at the top of the pane with a button for creating initial facts the concepts in that model.

A node must be selected somewhere in the tree hierarchy for the concept for which you wish to add an initial statement. Pressing the ‘bel’ toolbar button will cause a new initial belief to be created and added to the specified concept and appear ready for editing in the tree. Pressing the ‘fct’ toolbar button will cause a new initial fact to be created and added to the specified concept and appear ready for editing in the tree. If no node is selected in the tree then pressing the ‘bel’ or ‘fct’ button will have no effect.

9.8. Creating Initial Facts and Beliefs Using Drag and Drop in the Model Trees.

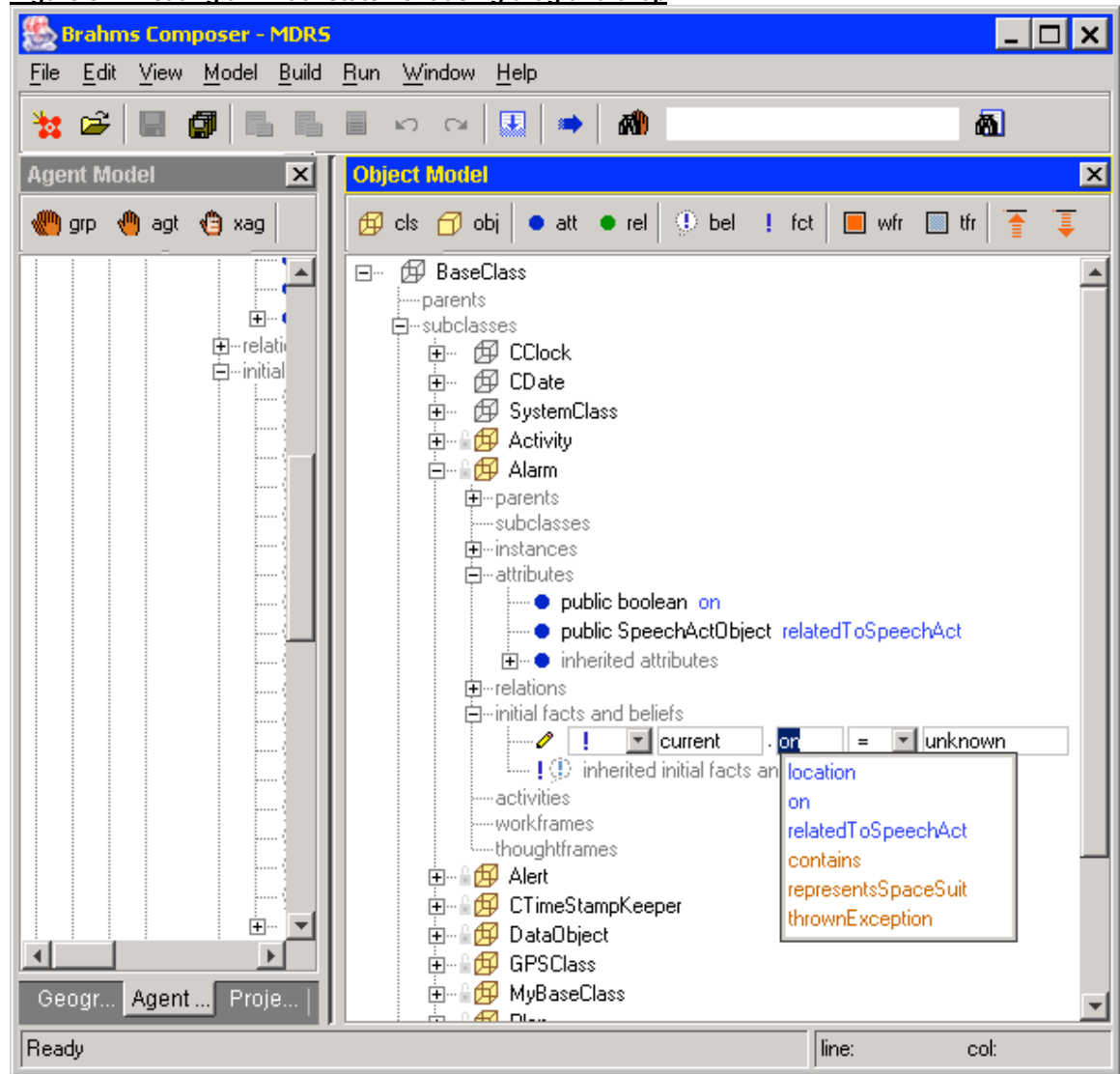
An initial fact or belief regarding a specific attribute or relation of a specific concept can be created and added to a concept by dragging the tree node representing the attribute or relation and dropping it onto the ‘initial statements’ node of a Group, Agent, Class, Object, AreaDef or Area.

If the attribute or relation is dropped onto the ‘initial statements’ node of a Group or Agent then a new initial belief is created and added to the Group or Agent and is presented ready for editing in the model tree. If the attribute or relation is dropped onto the ‘initial statements’ node of a Class, Object, AreaDef or Area then a new initial fact is created and added to the concept and presented ready for

editing in the model tree. The source code file for the edited concept is auto-edited to add the new initial statement declaration.

For example, figure 8.4 shows the result of dragging the tree node representing the attribute “public boolean on” in the concept Alarm and dropping it onto the ‘initial statements’ node of the same concept.

Figure 8.4 Creating an initial statement using drag and drop



A new initial fact is added to the concept alarm with the condition statement: “current.on = unknown”. The new initial fact is presented ready for editing in the model tree.

Tree nodes representing inherited attributes and relations can also be dragged and dropped onto ‘initial statements’ nodes.

9.9. Moving or Copying Initial Statements Using Drag and Drop in the Model Trees

Any initial statement can be moved from one concept to another by dragging the node representing the initial statement and dropping it onto the ‘initial statements’ node of the concept to which you wish to move the initial statement. The dragged initial statement will be deleted for the original concept and added to the target concept. This provides a useful way to move initial statements up or down the inheritance hierarchy as you are designing your model.

An initial statement may be copied to another concept by dragging the initial statement with the CTRL key held down. This will result in an initial statement with the same condition statement being added to the target concept while the original concept is left unchanged.

9.10. Editing Initial Statements Using the Model Trees

Any initial statement can be edited by selecting the tree node representing the initial statement in the tree hierarchy and pressing CTRL-E. This will cause the tree cell editor to activate for that initial statement and you will be able to edit any of its fields. Pressing the ENTER key will keep the edits and cause the source code file where the initial statement is declared to be auto-edited to reflect the changes.

Initial statements can be deleted from a concept by selected the tree node representing the initial statement in the tree hierarchy and pressing the DELETE key. The source code file for the specified concept will be auto-edited to have the initial statement declaration removed.

9.11. Adding Activities to Active Concepts Using the 3-Character Abbreviations in the Model Trees

Activities and activity parameters can be created and added to concepts or composite activities in the Composer application by typing a three-character keyboard shortcut when a node is selected within the tree structure for any concept or composite activity in the Agent Model Tree, Object Model Tree or Activity Model Tree.

The 3-character abbreviations for activities and parameters are:

- primitive_activity ‘pac’
- move ‘mov’
- create_agent ‘caa’
- create_area ‘car’
- create_object ‘coa’
- communicate ‘com’

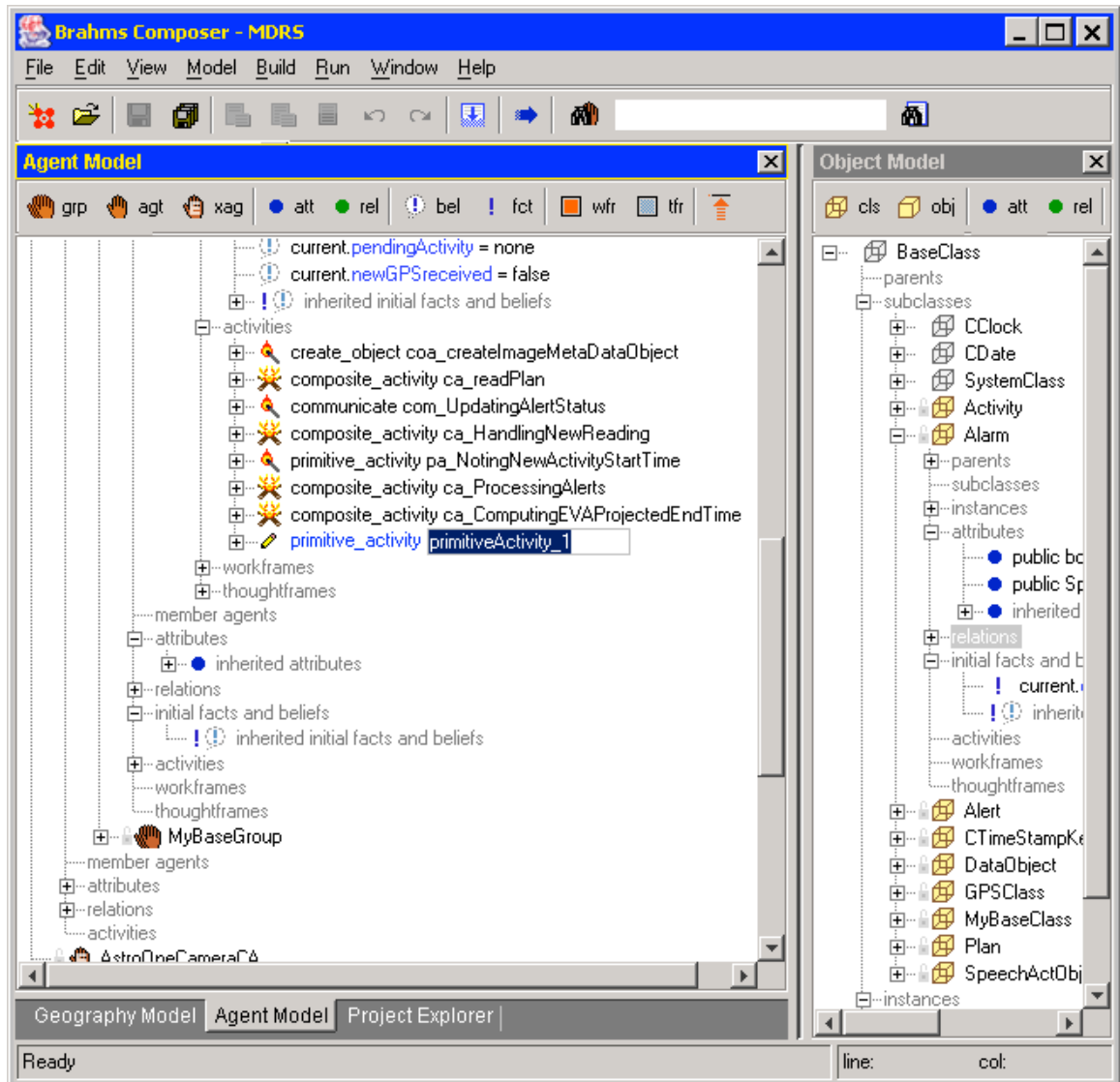
- broadcast 'bct'
- get 'get'
- put 'put'
- gesture 'gac'
- java 'jac'
- composite 'cac'
- activity parameter 'par'

For example, if a Group node is selected in the Agent Model Tree and you type the three characters 'pac', a new primitive activity will be created and will appear as a child node of the 'activities' node of the selected Group. Typing 'cac' would result in a new composite activity being created as a child of the selected Group's 'activities' node.

If a composite activity is selected and you type the shortcut 'car' then a new create_area activity would be created as a subactivity of the selected composite activity.

The new activity will be presented in the model tree ready for editing its name.

Figure 8.5 New primitive activity ready for editing the name.



After entering a name for the new activity and pressing enter, the source code file where the concept containing the activity is defined will be auto-edited to add a definition for the new activity. All newly created activities in the model trees are given the default properties as defined in the Brahms Language Specification, and, if necessary, any default parameters required for the activity to compile without errors. You may further edit an activity's properties using the Activity Editor as described below.

If an active concept contains at least one activity, then a node representing that concept is added to the 'performers' node in the Activity Model Tree.

Note: You cannot add activities to concepts or composite activities using the keyboard shortcuts if the source file where the concept or composite activity is defined is open on the desktop. You must first close the source code file.

9.12. Adding Activities Using the Activity Model Pane Toolbar Buttons

The pane for the Activity Model contains a toolbar at the top of the pane with buttons for creating each type of activity in a Brahms model. The Activity Model also contains an activity scratch-pad node where activities may be added that are not assigned to any concept. Activities added to the scratch-pad do not result in any edits being made to source code, so you may construct activity hierarchies here without concern for compile errors since they will not be included when you compile your model. Activities created in the scratch-pad can be moved or copied to active concepts in the model using drag and drop as described below.

A new activity can be added to an active concept, to a composite activity, or to the scratch-pad using the create activity buttons in the toolbar. A node representing one of these elements must be selected in the Activity Tree. Pressing one of the Activity Model Pane's toolbar buttons will cause a new activity of the button type pressed to be created and added to the selected element in the model tree. The new activity will be presented ready for editing.

If no node is selected in the tree then pressing a create activity button will have no effect.

New activities added to an active concept or to a composite activity that is part of an active concept will result in the source code file where that concept is defined being auto-edited to include the new activity declaration.

9.13. Moving or Copying Activities Using Drag and Drop in the Model Trees

Any activity can be moved from one activity-containing element (active concept, composite activity or activity scratch-pad) to another by dragging the node representing the activity and dropping it onto the node representing the active concept, or dropping it onto the 'activities' node of an active concept, or dropping it onto the node representing a composite activity, or dropping it onto the 'activities' node of a composite activity, or dropping it onto the node representing the activity scratch-pad. The dragged activity will be deleted from the original element and added to the target element. This provides a useful way to move activities up or down the inheritance hierarchy as you are designing your model.

An activity may be copied to another element in the model by dragging the node representing the activity with the CTRL key held down. This will result in an exact copy of the dragged activity being added to the target element (concept, composite activity or scratch-pad) while the original element is left unchanged.

Moving activities using drag and drop is a good way to shift activities up or down the inheritance hierarchy while designing your model.

The source code files for the concepts that contain activities that are moved or copied in the model trees will be auto-edited to add or remove the activity definitions.

NOTE: You cannot drag an activity when the source file where that activity is defined is open on the desktop and you cannot drop activities onto an element whose source code file is open on the desktop. You must first close the source files before attempting to drag and drop activities.

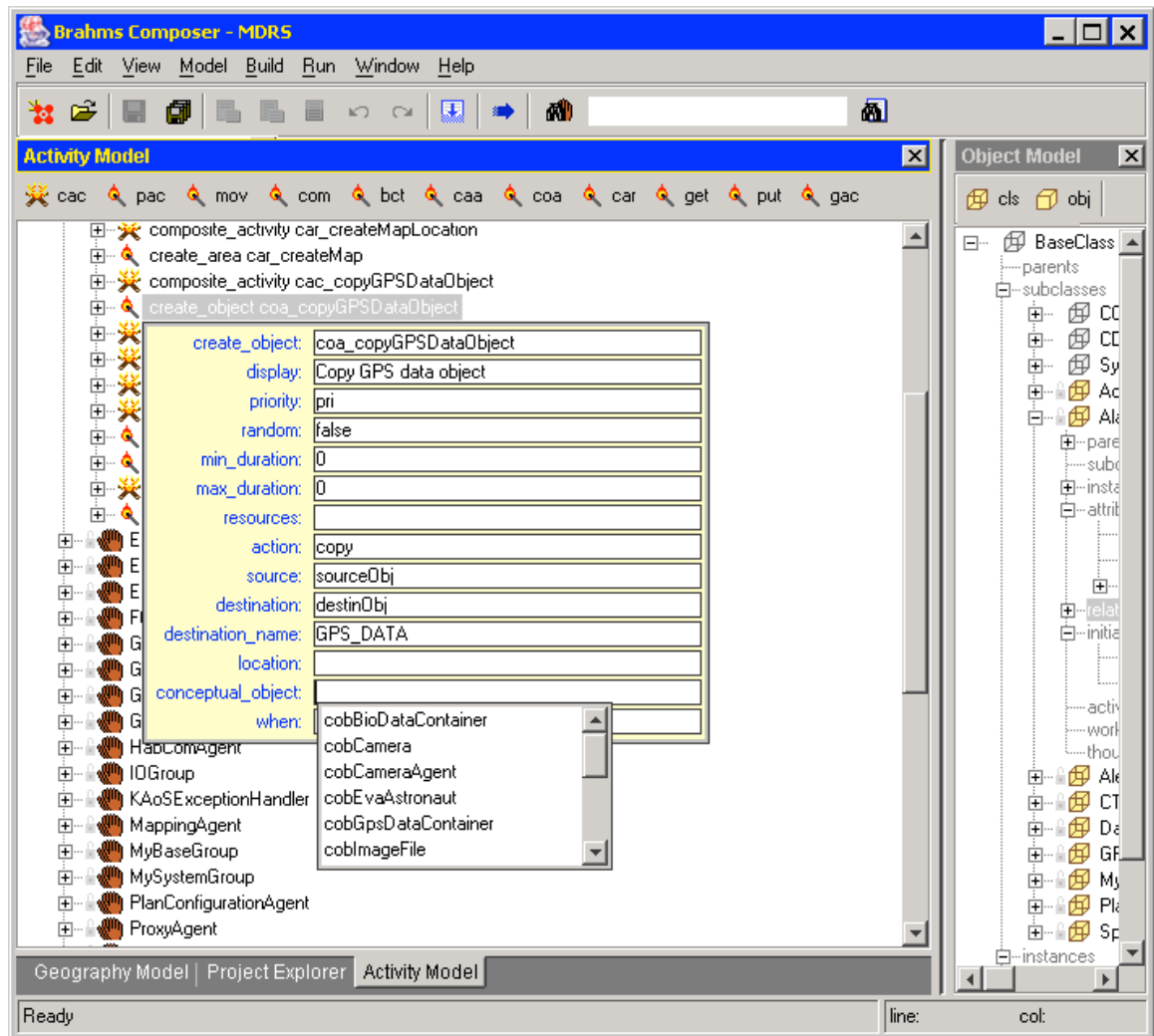
9.14. Editing Activities Using the Activity Editor

Once an activity has been created its properties can be edited using the Activity Editor from within the Model Tree. Selecting an activity node in a Model Tree and pressing CTRL-E will open the Activity Editor panel positioned beside the selected node.

The Activity Editor can be used to edit the activity name and all built-in properties for the particular activity type being edited.

When you edit an activity's name using the Activity Editor, all tree nodes representing model elements that reference that activity name are updated to reflect the change. Also, all source code files that contain references to that activity are auto-edited to reflect the change (provided the source code file is not open on the desktop).

Figure 8.6 The Activity Editor for a create_object activity



Pressing the ENTER key will save the edits made in the Activity Editor and close the panel. Pressing the ESCAPE key or clicking outside the panel will close the editor panel without saving any edits.

All edits made and saved in the Activity Editor are automatically written to the source code file where the concept containing the activity is defined.

As shown in figure 8.6, placing the cursor in some fields in the Activity Editor will pop-up a list of choices that are valid values for that field, including the names of the parameters declared for that activity where it is permitted to use a parameter as the assigned value. In the example in figure 8.6, a list of all Conceptual Objects in the model has been presented for editing the 'conceptual_object' property for the 'create_object' activity being edited. Double-clicking an item in the list will set the text in the field being edited to the selected item. Also, selecting an item in the list using the mouse or arrow keys and then tabbing to the next field in the Activity Editor will set the text in the field to the selected item.

NOTE: You cannot open an Activity Editor for an activity when the source file where that activity is defined is open on the desktop. You must first close the source file before attempting to edit the activity using the Activity Editor.

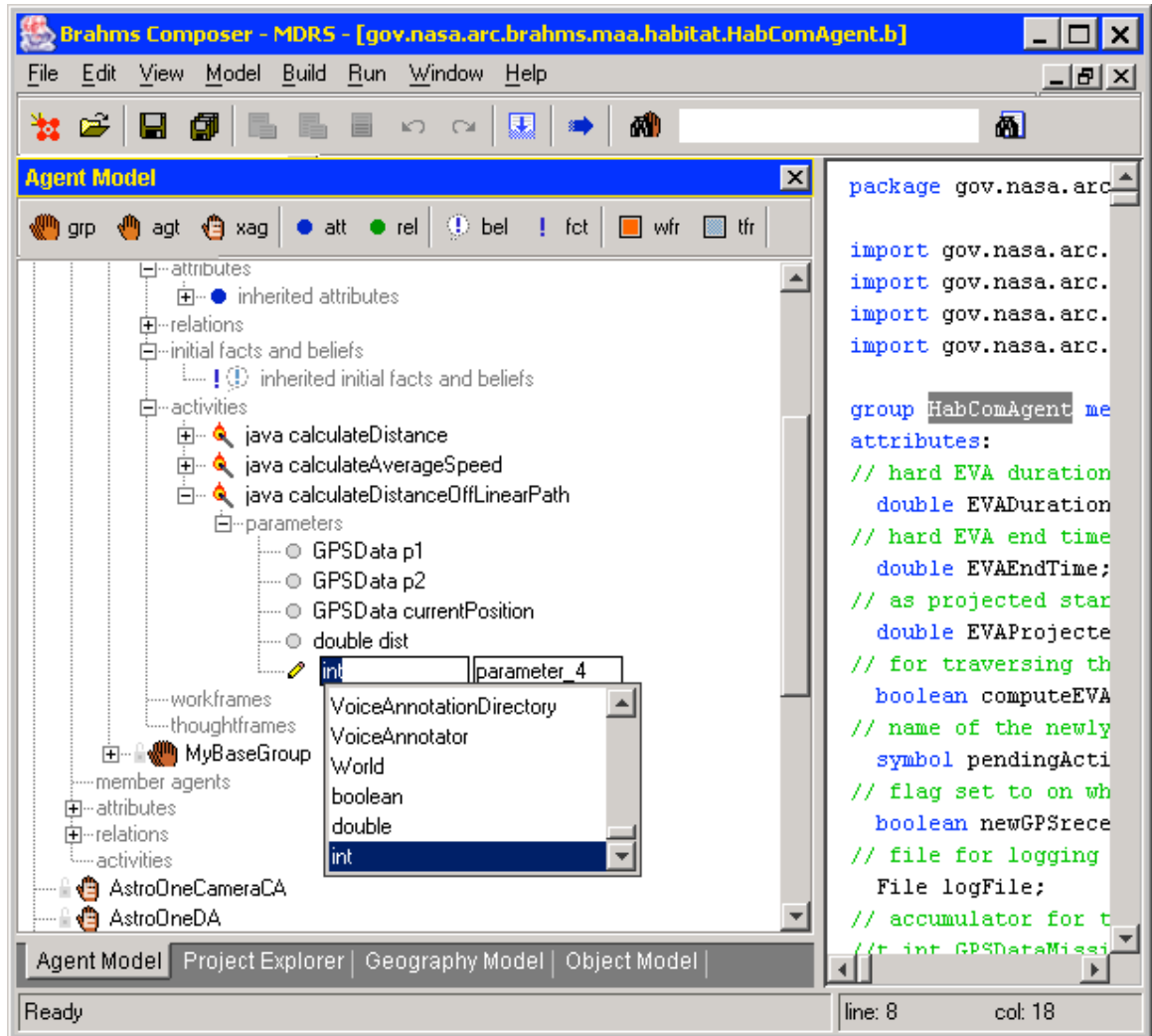
Activities may be deleted from the model by selecting any activity node in a model tree and pressing the DELETE key. The source code file where the activity was defined will be auto-edited to remove the activity definition. Deleting a composite activity will also delete all sub-activities of the composite activity.

When the last activity is deleted from a concept, the node representing that concept under the 'performers' node of the Activity Model Tree is also deleted.

9.15. Creating and Editing Activity Parameters in the Model Trees

Parameters may be defined for an activity by selecting the tree node representing the activity or one of its sub-nodes and typing the characters 'par'. A new parameter will be created and added to the activity. The parameter will be presented as a sub-node of the activity node's 'parameters' node. The new parameter will appear ready for editing using the parameter tree cell editor.

Figure 8.7 New parameter presented for editing



New parameters are created with type="int" and name="parameter_n". When the cursor is in the 'type' field of the parameter tree cell editor a list is popped-up that contains all possible parameter types that are currently available in the model. Typing in the field will auto-select the first item in the list that matches the characters being typed. Tabbing out of the field will set the text in the field to the selected item in the list. You may also double click any item in the list to set the text in the 'type' field to that item.

A parameter may be edited at any time by selecting the tree node representing the parameter and pressing CTRL-E. This will cause the parameter tree cell editor to activate.

A parameter may be deleted from an activity by selecting the tree node representing the parameter and pressing the DELETE key.

Adding, editing or deleting parameters using the Model Trees will result in the source code file where the activity is defined being auto-edited to reflect the changes.

9.16. Adding Workframes and Thoughtframes to Active Concepts and Composite Activities Using the 3-Character Keyboard Shortcuts in the Model Trees

Workframes and thoughtframes can be created and added to active concepts or to composite activities in the Composer application by typing a three-character keyboard shortcut when a node is selected within the tree structure for any concept or composite activity in the Agent Model Tree or Object Model Tree.

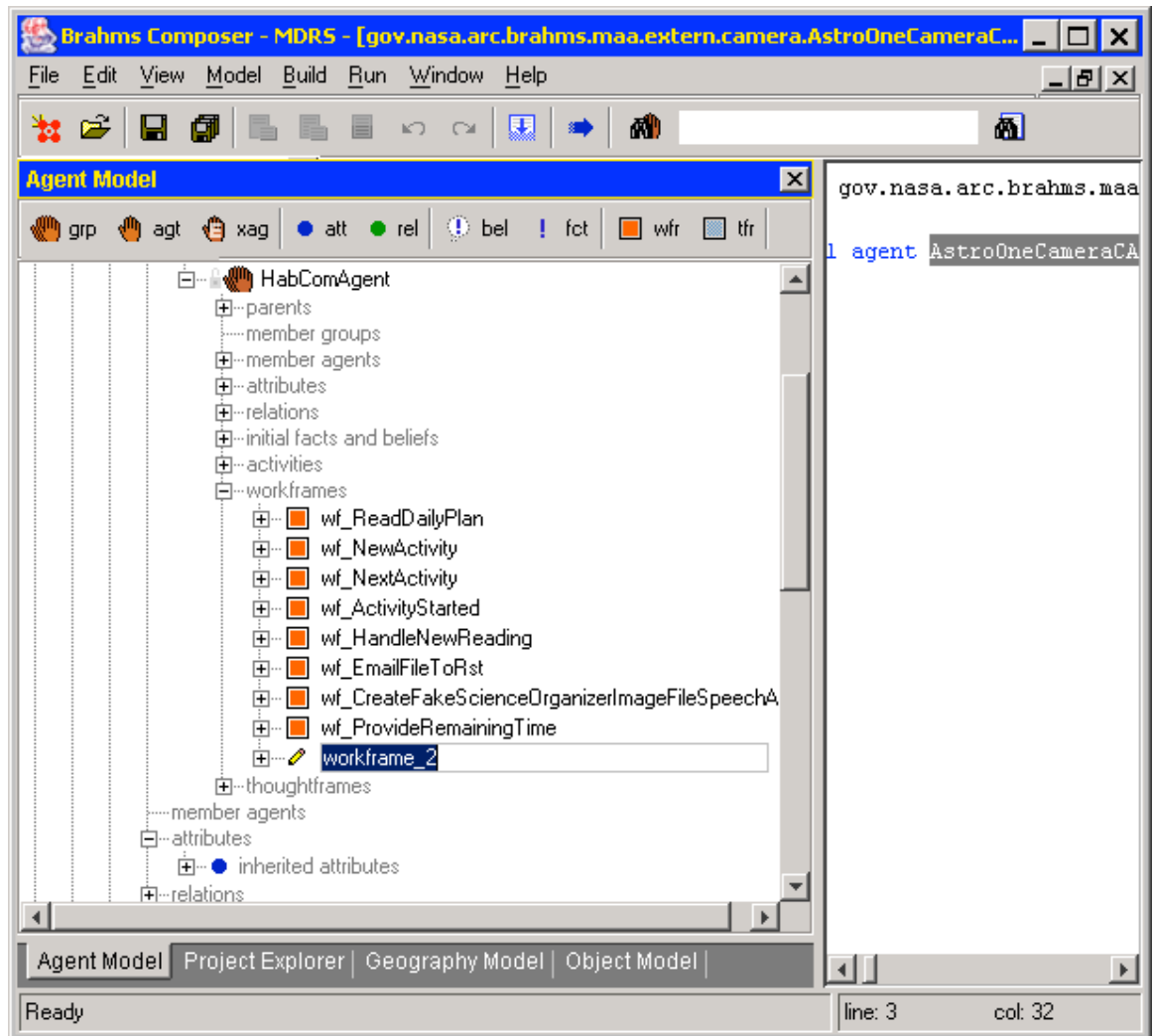
The 3-character abbreviations for workframes and thoughtframes are:

- workframe 'wfr'
- thoughtframe 'tfr'

For example, if a Group node is selected in the Agent Model Tree and you type the three characters 'wfr', a new workframe will be created and will appear as a child node of the 'workframes' node of the selected Group. Typing 'tfr' will result in a new thoughtframes being created as a child of the selected Group's 'thoughtframes' node.

The new workframe or thoughtframe will be presented in the model tree ready for editing.

Figure 8.8 New workframe presented for editing in the Agent Model Tree.



After giving the new workframe or thoughtframe a name and pressing the ENTER key an empty workframe or thoughtframe will be added to the model. The source code file where the active concept is defined will be auto-edited to have the new, empty workframe or thoughtframe definition added.

A workframe or thoughtframe can have its name edited at any time by selecting the tree node representing the frame and pressing CTRL-E. This will cause the tree cell editor for the frame to appear in the model tree. Pressing ENTER after editing the name will result in the source code file where the frame is defined to be auto-edited to reflect the change.

A workframe or thoughtframe can be deleted from the model by selecting the node representing the frame and pressing the DELETE key. The frame will be deleted from source code as well.

9.17. Creating Workframes and Thoughtframes Using the Model Pane Toolbar Buttons

The panes for the Agent Model and the Object Model contain a toolbar with buttons for creating workframes and thoughtframes for the concepts in that model.

A node must be selected somewhere in the tree hierarchy for the concept or composite activity to which you wish to add a new workframe or thoughtframe. Pressing the 'wfr' toolbar button will cause a new workframe to be created and added to the selected concept or composite activity and appear ready for editing in the tree. Pressing the 'tfr' toolbar button will cause a new thoughtframe to be created and added to the selected concept or composite activity and appear ready for editing in the tree. Pressing the toolbar buttons will have no effect if a node is not selected in the tree.

9.18. Moving or Copying Workframes and Thoughtframes Using Drag and Drop in the Model Trees

Any workframe or thoughtframe can be moved between frame-containing model elements by dragging the node representing the frame and dropping it onto a node representing the frame-containing element or the 'workframes' or 'thoughtframes' node of the frame-containing element to which you wish to move the frame. The dragged frame will be deleted from the original element and added to the target element. This provides a useful way to move frames up or down the inheritance hierarchy or activity hierarchy as you are designing your model.

A workframe or thoughtframe may be copied to another frame-containing element by dragging the node representing the frame with the CTRL key held down. This will result in an exact duplicate of the dragged frame being added to the target element while the original element is left unchanged.

Moving or copying frames using drag and drop in the model trees will result in the source code files where the concepts containing the frames are defined being auto-edited to reflect the changes in the model.

9.19. Adding Frame Elements to Workframes and Thoughtframes Using the 3-Character Keyboard Shortcuts

Workframes and thoughtframes are presented in the model trees with sub-nodes for containing the nodes representing the sub-elements of the frames.

Workframe nodes have the sub-nodes:

- variables

- detectables
- preconditions
- body

Thoughtframe nodes have the sub-nodes:

- variables
- preconditions
- body

The elements of workframes and thoughtframes can be added to the model using the 3-character keyboard shortcuts for each element. When created, nodes representing the elements will appear in the appropriate sub-node of the frame. Consequences and activity-refs are added to the 'body' node.

The 3-character keyboard shortcuts for frame elements are:

- variable 'var'
- detectable 'det'
- precondition 'pre'
- consequence 'con'
- activity-ref 'arf'

Typing one of the 3-character shortcuts while a node within the tree structure for the frame is selected will result in a new element of that type being added to the frame. The new element will appear as a sub-node of the appropriate node in the frame's tree structure and will be presented ready for editing using a tree cell editor.

The source code file where the workframe or thoughtframe is defined will be auto-edited to add the new element to the frame definition.

New variables are created with the default declaration:

`"foreach symbol new_variable;"`

New detectables are created with the default detectable statement declaration:

`"when(whenever) detect ((current.location = unknown), dc:100) then continue;"`

New preconditions are created with the default declaration:

`"knownval(current.location = unknown)"`

New consequences are created with the default declaration:

`"conclude((current.location = unknown), fc:100, bc:100);"`

New activity-refs are created with the default value

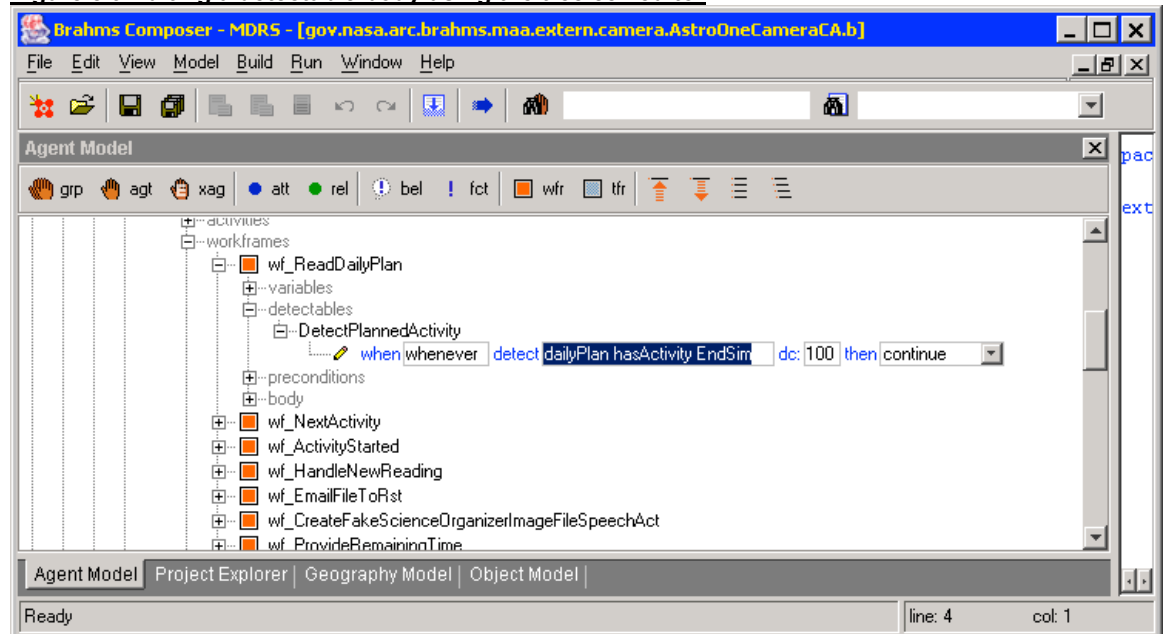
`"new_activity_ref();"`

The elements of workframes and thoughtframes can be edited at any time by selecting the tree node representing the element you wish to edit and pressing CTRL-E. This will result in the tree cell editor for that element being activated in the model tree.

The elements of workframes and thoughtframes can be deleted from a frame by selecting the tree node representing the element you wish to delete and pressing the DELETE key.

The tree nodes representing detectables appear with the detectable name. The detectable tree node contains one sub-node for the detectable body. The detectable name can be edited by selecting the node representing the detectable and pressing CTRL-E. The body of the detectable can be edited by selecting the sub-node and pressing CTRL-E.

Figure 8.9 Editing a detectable body using the tree cell editor



Detectables can also be added to composite activities using the 'det' keyboard shortcut.

9.20. Adding Activity-refs to Workframes Using Drag and Drop

You can add an activity-ref for a specific activity to a workframe by dragging the tree node representing the activity and dropping it onto the 'body' node of the workframe to which you wish to add the activity-ref. This will result in a new activity-ref with the name of the dragged activity being added to the body of the workframe. The new activity-ref will be presented in the model tree ready for editing so you may add parameter names if necessary.

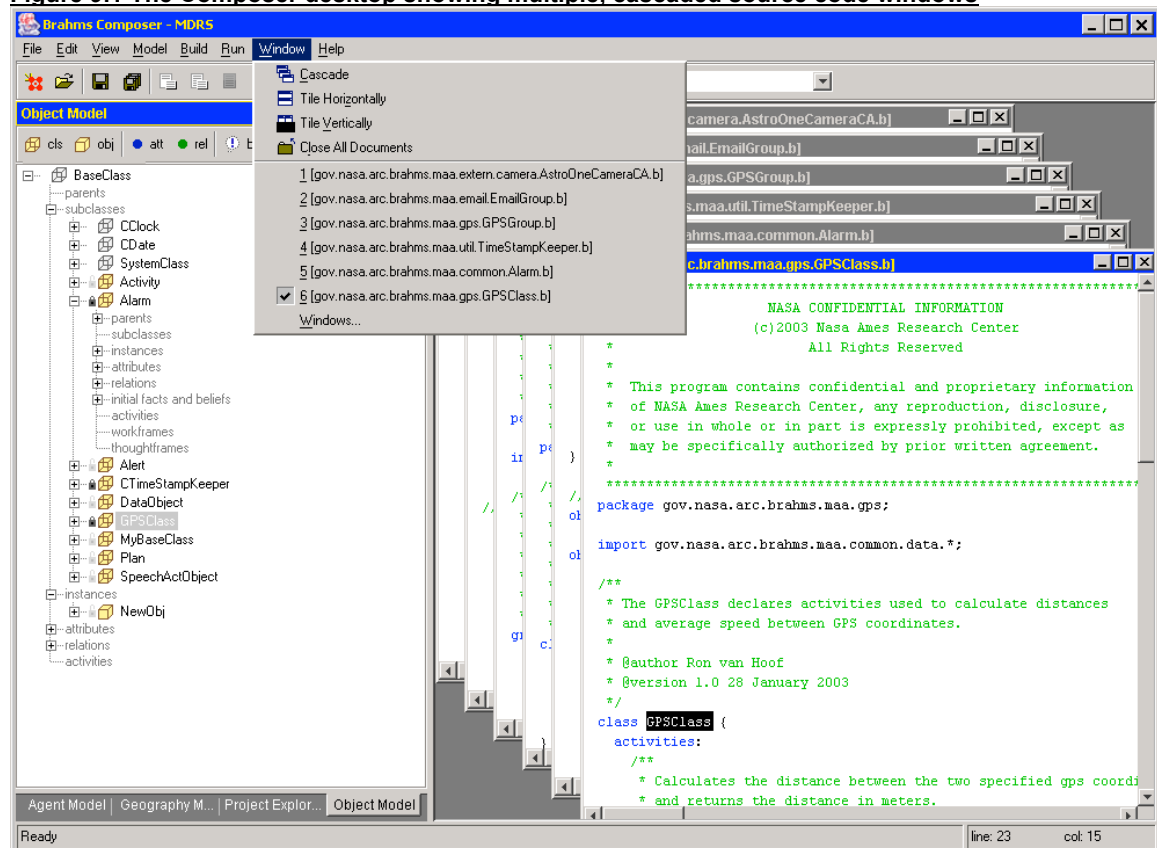
10. The Composer Application Desktop and Source Code Editor

The Composer application features a true multiple-document desktop and a keyword color-coded source code editor that allows you to build complete Brahms models by writing source code.

The multiple-document desktop allows multiple editor windows open on the desktop with presentation style choices of maximized, minimized, free-form, cascaded, tiled horizontally or tiled vertically.

You can choose among the presentation styles or select one of the currently open windows using the 'Windows' menu in the Composer menu-bar.

Figure 9.1 The Composer desktop showing multiple, cascaded source code windows



When the editor windows are maximized on the desktop the title of the active window is presented in the title bar of the application.

10.1. Opening Source Files in the Composer Application

There are several ways to open the source files for model elements in the composer application.

- Double-click on the tree node representing any model element in any of the Visual Designer Model Trees. This will result in the source file where that element is defined being opened in an editor window on the desktop with the selected element scrolled into view and highlighted in the editor.
- Select the tree node representing any model element in any of the Visual Designer Model Trees and press the right mouse button. This will pop-up a menu with the menu choice 'Open Source File'. Selecting this menu item will result in the source file where the selected element is defined being opened in an editor window on the desktop with the selected element scrolled into view and highlighted in the editor.
- Drag a tree node representing any concept from one of the Visual Designer Model Trees and drop it onto the desktop. This will open the file where the dragged concept is defined.
- Double-click on the tree node representing a source file in the Project Explorer pane.
- Select one or more tree nodes representing source files in the Project Explorer pane and press the ENTER key. This will result in all selected files being opened on the desktop.
- Drag the tree node representing a source file from the Project Explorer pane and drop it onto the desktop.
- Use the 'Open File' menu item in the 'File' menu.

When a source code file is open in an editor window, all tree nodes in the Visual Designer Model Trees that represent the concept(s) defined in that file will appear with a 'locked' icon next to the concept icon. This signifies that the concept and its elements are not editable using the Visual Designer editing capabilities while the source file for the concept is open on the desktop.

10.2. Editing Source Files in the Composer Application

Source File Editor Color Coding

The text in a source code editor window has the color coding:

- Brahms keywords blue
- String literals gray
- Comments green

Color coding takes place as you type in the source file editor.

Undo and Redo

For the length of time a file is open on the desktop the Undo and Redo feature is available to allow you to back up through multiple edits or to move forward to redo edits that have been undone.

The Undo and Redo actions can be invoked by selecting the 'Undo' or 'Redo' menu items in the 'Edit' menu or they can be invoked by pressing the 'Undo' or 'Redo' buttons in the application toolbar.

Synchronization between source code editing and the Model Trees

Each time you save a file while editing in a Composer editor window the changes made in the source file are reflected in the Visual Designer Model Trees. If model elements have been added to a file then new tree nodes representing those elements will appear in the Model Trees. If elements are deleted from a file then all tree nodes representing those elements are removed from the Model Trees.

Each time you save your entire model all open source code files with edits are saved and the Visual Designer views are synchronized with the changes made to all the edited files.

In order to keep the views synchronized the Composer application parses the source file each time it is saved to resolve the model elements. If the parser encounters any syntax errors when parsing the file they are presented in the Task List pane. Parser errors are listed in the Task List by file and they will appear before any compiler errors that are currently in the Task List.

See section 11 below for a discussion of the Task List and how to use it to navigate to errors in source files.

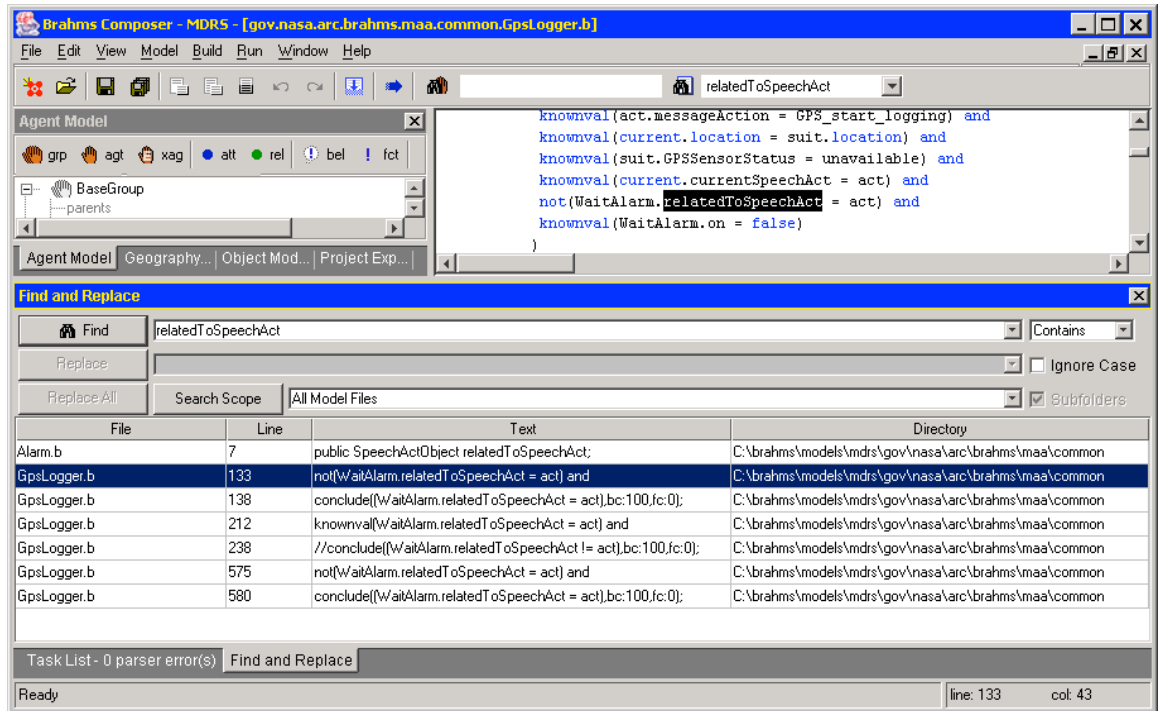
10.3. Using the Find and Replace Dialog

The Find and Replace dialog offers several ways to search the source files in your model to locate specified strings. To display the Find and Replace dialog do one of the following:

- select the 'Find and Replace...' menu item from the 'Edit' menu
- press the 'Find and Replace' button in the application toolbar
- use the keyboard shortcut CTRL-F

The Find and Replace dialog will appear in the application window docked in the last place it was placed by the user.

Figure 9.2 The Find and Replace Dialog



The Find and Replace dialog has several fields that allow you to structure your search.

- The 'Find' field – Enter the text here that you wish to search for.
- The search type list – This is a drop down list that allows you to specify how the string should be located in the file text. The choices are: “Contains”, “Whole Word”, “Starts With” and “Ends With”.
- The 'Replace' field – Enter the text here that you would like used to replace the search text. Note: this field is only editable for searches with a scope of “Current Document” or “Current Selection”.
- The 'Ignore Case' checkbox – Select this if uppercase or lowercase text does not matter for your search.
- The 'Search Scope' field – This is a drop down list that allows you to specify the scope for your search. The default choices are: “Current Document”, “Current Selection”, “All Open Documents” and “All Model Files”. Pressing the 'Search Scope' button will open a dialog that allows you to browse the file system to select a directory for your search scope.
- The 'Subfolders' checkbox – This field is editable when a directory has been chosen for the search scope. Selecting it means the search will include all subfolders of the specified directory as well.

When the search scope is set to “Current Document” or “Current Selection” then pressing the 'Find' button will scroll to and highlight the next occurrence of the search string within the document or within the selected text in the document starting from the present cursor position. Pressing the 'Replace' button will replace the currently highlighted occurrence of the search string and find the next

occurrence if it exists. When there are no more occurrences then a message dialog with this information will pop-up. Pressing the 'Replace All' button will replace all occurrences of the search string with the replacement text and present a message dialog informing you how many occurrences were replaced.

When the search scope is set to "All Open Documents", "All Model Files" or a specified directory in the file system then pressing the 'Find' button will list all occurrences of the search string in the table at the bottom of the Find and Replace dialog. The table contains the following fields:

- File – The name of the file where the occurrence was found
- Line – The line number where the search string occurs in the file
- Text – The complete text of the line containing the search string
- Directory – The directory path where the file is located

Double-clicking a row in the table will open the file on the desktop, scroll the file to the appropriate line and highlight the occurrence of the search string.

Once a file is open you may change the search scope to 'Current Document' and the 'Replace' field will become enabled allowing you to enter replacement text. The 'Replace' and 'Replace All' buttons will then become enabled allowing you to replace the search string with the specified replacement text.

Toolbar 'Find in Current Document' Shortcut

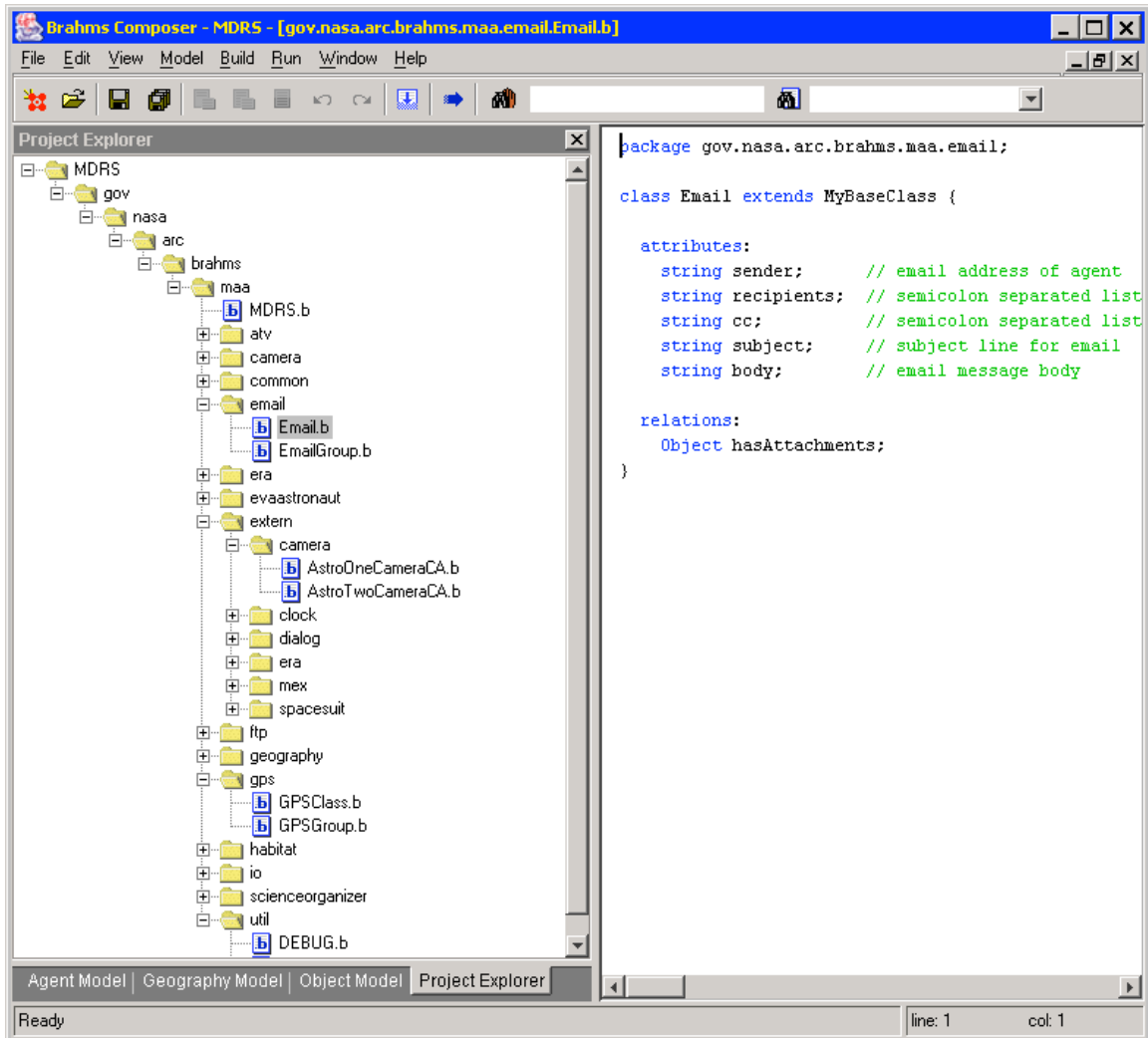
The Application toolbar contains an editable drop-down list next to the 'Find and Replace' button that contains a list of all search strings that have been entered during an application session. Selecting an item from this list or typing a new search string into this list and pressing the ENTER key will search for the string in the currently active document. Pressing the ENTER key again will find the next occurrence of the string.

11. Using the Project Explorer

The Project Explorer gives a package structure and source file view of your model. It is presented as a tree structure that reflects the directory structure of the model's package directories and files in the file system.

Figure 10.1 below shows the Project Explorer with a node representing a source file selected in the Project Explorer tree and the selected source file open on the desktop. Note that the package statement in the source file is reflected in the tree structure of the Project Explorer.

Figure 10.1 The Project Explorer

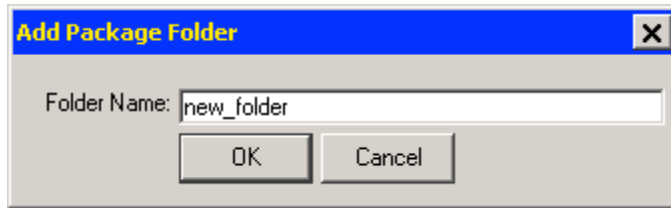


The Project Explorer also contains functionality for editing the package structure, creating new packages, editing file names and creating new files.

11.1. Creating a New Package Folder in the Project Explorer

To create a new package folder in the Project Explorer right-click on the folder you wish to be the parent of the new folder or on any file that is currently a child of the desired parent folder. A pop-up menu will appear with the menu item 'New Package Folder...'. Select this menu item and a dialog box will appear allowing you to enter a name for the new package folder. Enter a valid name and click the 'OK' button or press the ENTER key. The new package folder will appear in the Project Explorer tree and a new directory will be created at the appropriate place in the file system.

Figure 10.2 The 'Add Package Folder' Dialog



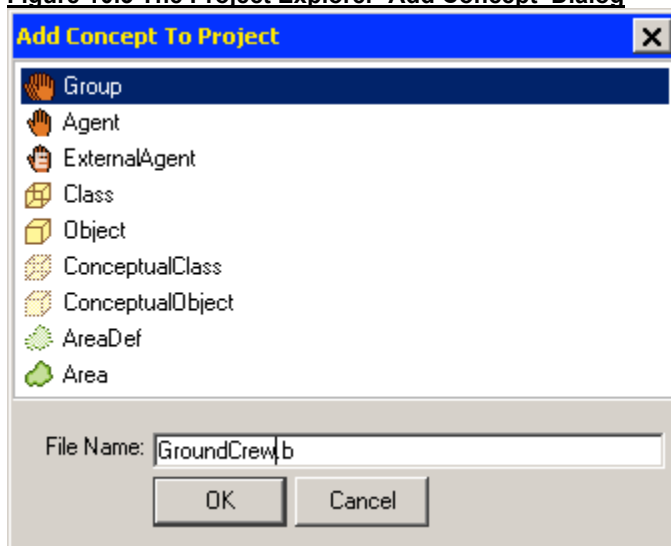
A package folder name can be edited at any time by clicking once on the tree node representing the package folder in the Project Explorer, pausing and clicking a second time. The tree node will become editable and you may enter a new name.

To delete package folders in the Project Explorer select the node representing the folder you wish to delete and press the DELETE key or use the right-click 'Delete' menu item. Package folders can not be deleted from the Project Explorer if they are not empty.

11.2. Creating a New Source File Using the Project Explorer

A new source file is created using the Project Explorer by adding a new concept to your model. The new concept will have the same name as the name you give to the file that contains it. To create a new concept and source file using the Project Explorer right-click on the package folder you wish to be the file's package or on any file nodes within that package. A pop-up menu will appear with the menu item 'Add Concept...'. Select this menu item and a dialog box will appear allowing you select the type of concept and to enter a name for the new concept's file.

Figure 10.3 The Project Explorer 'Add Concept' Dialog



Select the concept type and enter a valid file name and click the ‘OK’ button or press the ENTER key. A new file will be created at the appropriate place in the file system and a tree node representing the file will be added to the Project Explorer tree. The new file will be opened on the desktop. The new file will contain a package statement reflected the location where you added the file and an empty concept definition of the selected concept type. The concept name will be the same as the file name (minus the ‘.b’ extension).

Figure 10.4 Text of New Concept Created in Project Explorer
`package gov.nasa.arc.brahms.maa.evaastronaut;`

```
group GroundCrew {  
}
```

A source file name can be edited at any time by clicking once on the tree node representing the source file in the Project Explorer, pausing and clicking a second time. The tree node will become editable and you may enter a new name.

To delete source files in the Project Explorer select the node(s) representing the file(s) you wish to delete and press the DELETE key or use the right-click ‘Delete’ menu item. You will be asked to confirm each file deletion.

11.3. Changing the Package of a Source File Using Drag and Drop

A source file’s package may be change by dragging the tree node representing the source file in the Project Explorer from its current folder and dropping it on another folder in the Project Explorer. The node representing the source file will be placed in the new folder and the file will be moved in the file system. The package statement in the source file will be auto-edited to the new package providing the source file is not open on the desktop.

You can also drag and drop entire package folders. All source files in the dragged folder and its sub-folders will have their package statements auto-edited.

NOTE: It is a good idea to close all documents before editing the package structure of your model using drag and drop in the Project Explore. This will allow the Composer application to auto-edit the source file package statements. If the files are open when you change their package using drag and drop then you will have to edit the package statements manually.

12. Compiling Your Model and Using the Task List

A Brahms model open in the Composer application can be compiled at any time by selecting the ‘Build Model’ menu item in the ‘Build’ menu or by pressing the ‘Build Model’ button in the application toolbar.

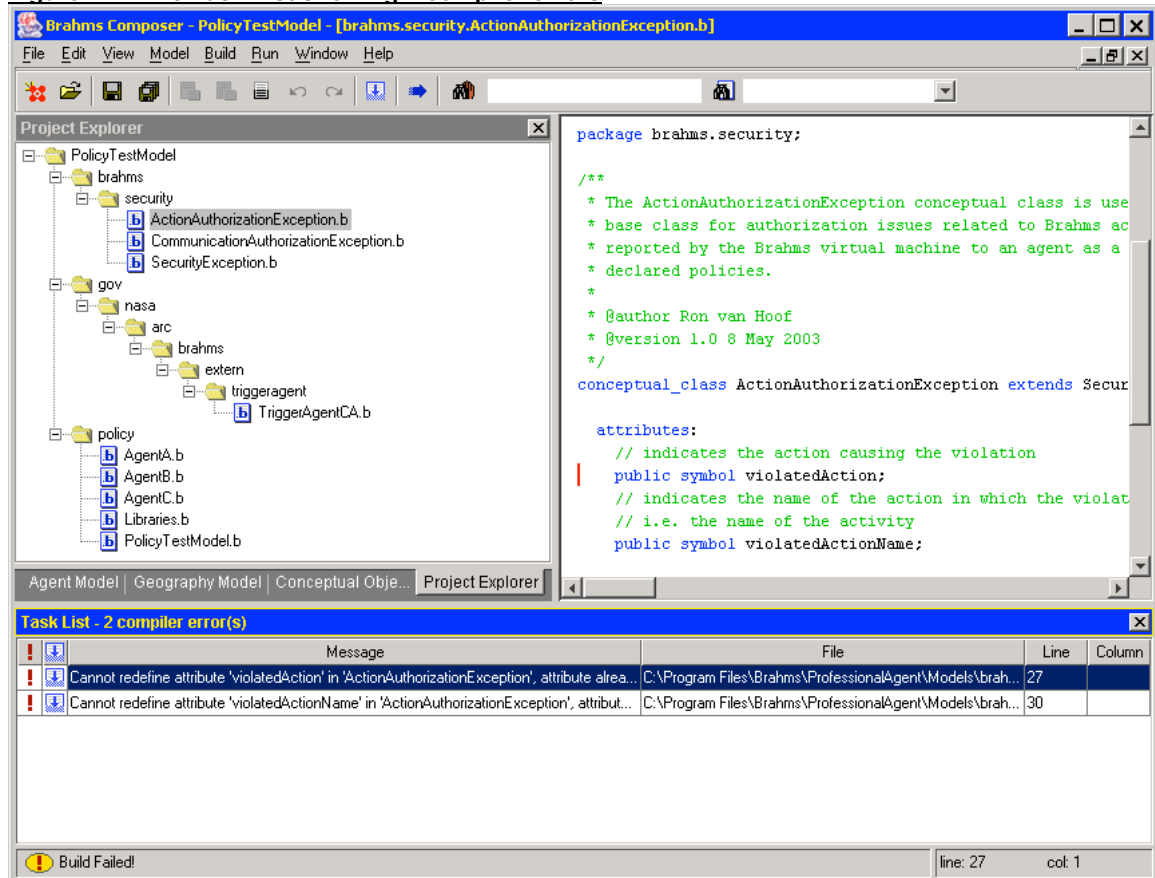
If the model compiles without errors you will see the message “Successful Build!” in the application status bar. If there are compiler errors you will see the message “Build Failed!” in the application status bar and the errors will be presented in the Task List pane. If the Task List pane is not currently open it will be opened automatically and docked where it was last positioned. Compiler errors will replace any parser errors that are currently listed in the Task Pane. The title bar of the Task List pane will contain information on how many compiler and/or parser errors are currently in the list.

Compiler and parser errors are presented in rows in the Task List with the following fields:

- Message – The error message reported by the compiler
- File – The full path name of the file containing the error
- Line – The line number where the error occurs in the file
- Column – The column number where the error occurs in the file

Double-clicking on an error in the Task List will open the source code file where the error occurs, scroll the file to the appropriate line and insert a red blinking cursor at the appropriate column.

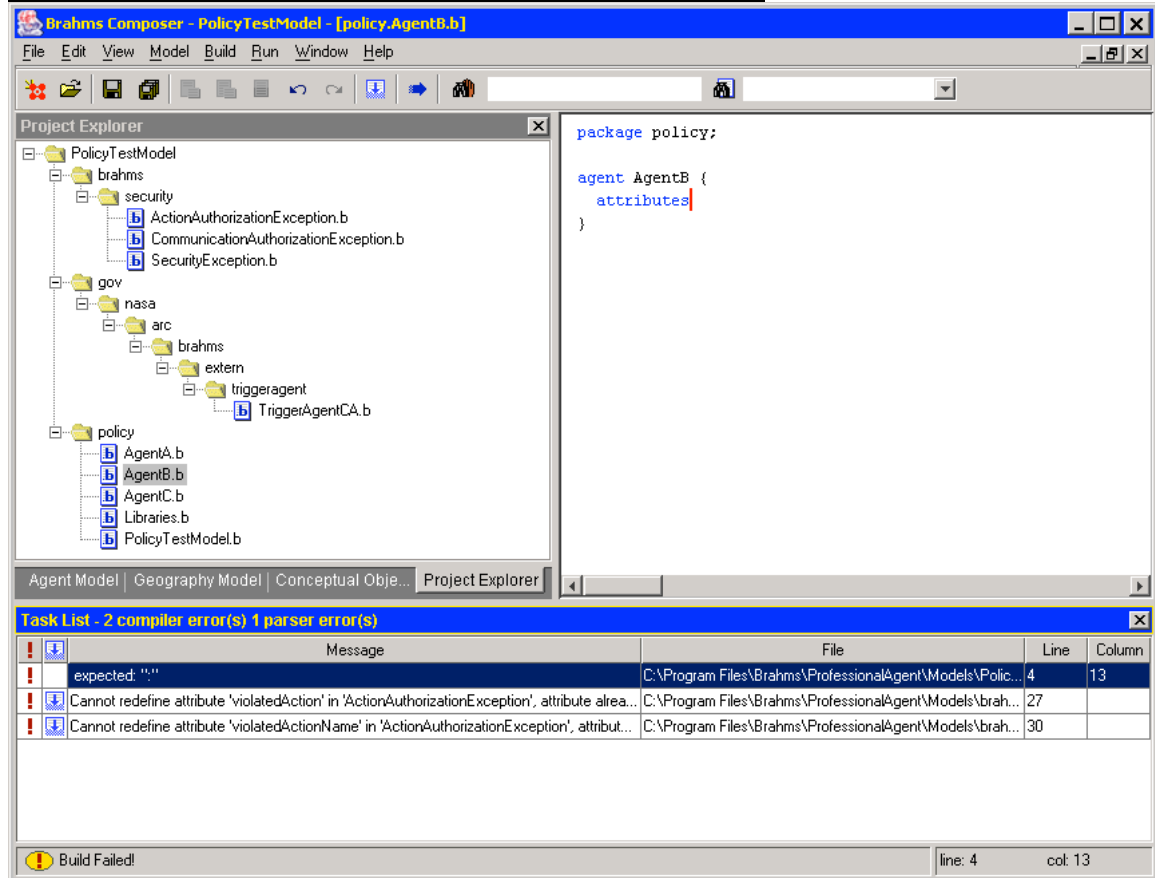
Figure 11.1 The Task List showing 2 compiler errors



Compiler errors will remain in the Task List until the errors are corrected and the model is compiled again.

If parser errors occur while editing and saving source code files they will be inserted into the Task List before compiler errors that are currently listed. Parser errors will not replace compiler errors in the Task List.

Figure 11.2 Task List showing 1 parser error and 2 compiler errors



Compiler errors are distinguished from parser errors in the Task List by the 'build' icon in the column preceding the error message.

Parser errors for a file are removed from the Task List when the error is corrected and the file is saved.

13. Running Your Model in the Composer Application

Once your model has been successfully compiled it may be run from within the Composer application. To run your model do one of the following:

- Select the 'Run Model' menu item from the 'Run' menu.
- Press the 'Run Model' button in the application toolbar.

When your model begins running a panel will open in the application window that will receive log messages from the virtual machine in real time. You can follow the progress of your model run by reading the messages that appear in the log.

Also, once the virtual machine has finished loading the model, the 'Run Model' menu item and toolbar button will change to 'Stop VM' menu item and toolbar button. Selecting the 'Stop VM' menu item or pressing the 'Stop VM' toolbar button will halt the virtual machine and stop your model run.

13.1. Specifying a VM Config File for your Model

A config file containing properties to be used by the Virtual Machine when running your model may be specified using the Composer application. In order to specify a config file select the 'Set VM Configuration File...' menu item from the 'Run' menu. A file chooser dialog will be displayed. Navigate to the .cfg file you wish to use and select it. The virtual machine will now use properties from this file when you next run your model from within the Composer application.