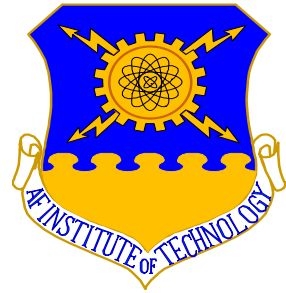


July 2000



agentTool User's manual

GRADUATE SCHOOL OF ENGINEERING AND MANAGEMENT
AIR FORCE INSTITUTE OF TECHNOLOGY
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

Approved for public release; distribution unlimited

Table of contents

BASIC GRAPHIC INSTRUCTIONS	1
1. ADDING	1
1.1 Objects	1
1.2 Connectors	1
1.3 HotKeys	1
2. MOVING & MANIPULATING	1
2.1 Objects	1
2.2 Connectors	1
2.3 Manipulating	3
2.4 Multiple Objects	3
3. DELETING	4
3.1 Objects & Connectors	4
4. PRINTING	4
BUILDING YOUR SYSTEM	5
1. GOALS	5
2. USE CASES	6
3. SEQUENCE DIAGRAMS	7
4. ROLE DIAGRAMS	7
5. SPECIFYING AGENTS	9
5.1 Agent Panel	9
6. CONVERSATIONS	11
6.1 Specifying Conversations	11
6.2 State Diagrams	11
7. DEPLOYMENT PANEL	13
VERIFYING CONVERSATIONS	16
1.1 Verifying Conversations	16
2. SET UP FOR VERIFICATION PURPOSES	19
2.1 Within VisualAge	19
2.2 Outside VisualAge	19

AGENT TOOL V1.8 ADDITIONS (1-12-01)..... 20
AGENT TOOL V1.8 BUG FIXES..... 22

BASIC GRAPHIC INSTRUCTIONS

1. Adding

1.1 Objects

To add an object, click on the button on the left hand of the screen that is labeled as the object that you wish to add. (i.e.- the “Add Agent” button adds an agent).

1.2 Connectors

To add a connector, at least one object must already exist on the panel. If it does not, there is nothing to connect and a message in the message box will tell you that you must first add an object. Once you have at least one object on the panel, you may begin to add a connector. There are three steps to adding a connector. They are:

1. Click the correct button. (i.e.- if you wish to add a conversation, click “Add Conversation”)
2. Click on the first object that you wish to connect. (Click within the box for that object)
3. Click on the second object that you wish to connect.

If you wish to create a loop, just click on the same object twice.

1.3 HotKeys

On many of the buttons you will find you can also use hotkeys alt-XX to “press” the button without pressing it. This allows you to quickly add many components without moving the mouse back and forth to the buttons. Also, many of the menu items also have hotkeys associated with them. (XX will be underlined on the button text. i.e. – “Add Agent” button has alt-a for its hotkey.)

2. Moving & Manipulating

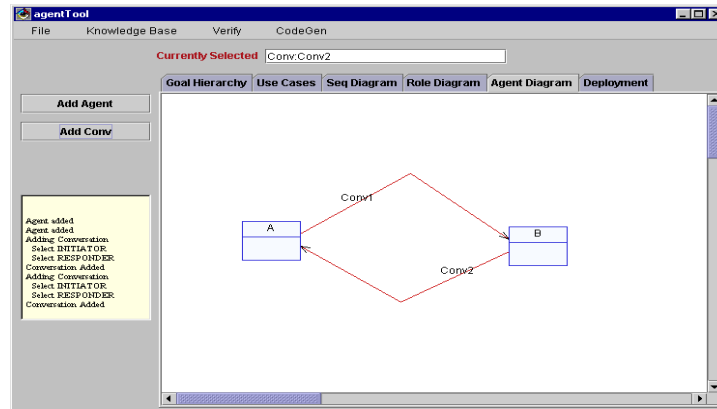
2.1 Objects

Moving- To move an object, click and hold the mouse somewhere inside the object you wish to move and then drag your object to wherever you wish on the displayed panel.

NOTE: It is not wise to move the selected object off the displayed panel. You may not be able to recover the object

2.2 Connectors

Moving- Connectors will move with their objects. Points may be added to the connectors between the two objects (except for loops). To do this, click on the connector between the two objects. Try to stay a good distance away from the components when doing this. This will add another point to your connector that can then be moved around just by clicking on it. When Components are moved, the points between the object do not move, so you may have to manually move them to meet your visual desires.

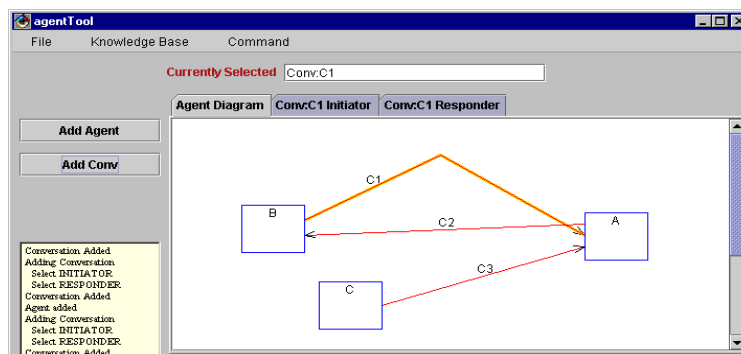


When multiple connectors are connected to the side of an object, the component arranges them in a manner it sees fit. If this order does not mesh with your requests, you can change it. To do so, click on or very near to the point where the connector and object connect. Now, drag the mouse above or below the closest connector. You will see that the two connectors will then swap places.

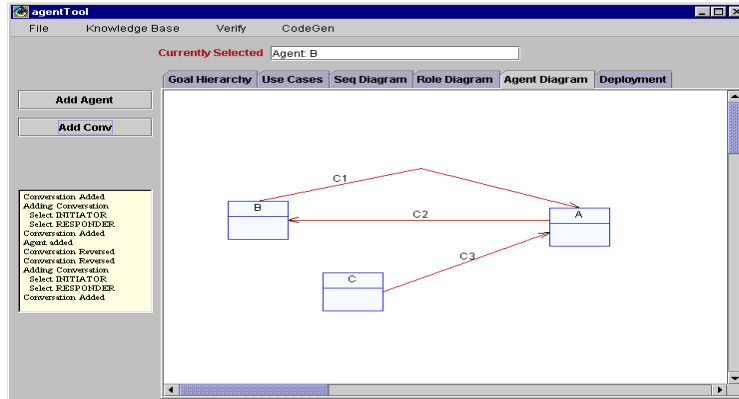
It is easy to clutter your screen with lines that contain many points, and often you find you have created a point that you do not want anymore. To remove a point from a line, simply drag it on top of another point in the line, or either end of the line. The two points will automatically merge when they are within a certain distance of each other.

2.2.1 Example

There are three connectors going into the left side of component “A”. Let us call them C1, C2, and C3, from top to bottom.



If we want C1 to be below C2, click on the end point of C1 (outside of the component “A”) and drag the mouse below the end point of C2. They will automatically switch. Now the order of connectors into the left side of component “A” from top to bottom is C2, C1, and C3.



NOTE: Do this slowly. Just switch two components at a time.

2.3 Manipulating

Some of the connectors can be reversed, made to originate from different objects, and made to connect to different objects. If you wish to do this, select the connector with the left mouse button and then click the right mouse button to get the popup menu to appear. If you choose to reverse the connection, all you have to do is choose “Reverse”. If you wish to change the objects that the connector is attached to, there is a series of three steps to be followed.

1. Decide if it is the initiator or the responder that you wish to change.
2. Make the appropriate selection from the popup menu.
(i.e.- if you wish to change the initiator choose “Change Initiator”)
3. Click on the object that you wish to connect it to.

2.3.1 Example

We have three objects “A”, “B”, and “C”, and there is a connector from “A” to “B”. First we want to reverse the connector, so we select “Reverse” from the popup menu and then the connector will start at “B” and finish at “A”. Now, we want to have the connector start at “C” and finish at “A”, so we select “Change Initiator” in the popup menu and then click on “C”. Now, the connector is from “C” to “A”

2.4 Multiple Objects

To move multiple objects, hold down the shift key while selecting the objects. If you wish to deselect an object, hold down the shift key and click on that object. To move the multiple objects, select one of the selected objects and move it. The rest will follow in formation. (This option does not currently work for all panels. –JAG 1-12-01)

NOTE: It is only necessary to move the objects (not the connectors). Connectors move with their relevant objects. However, the interior points (not the end points) do not move with the objects. Therefore, it is necessary for the user to move these after the objects have been placed appropriately.

3. Deleting

3.1 Objects & Connectors

To delete, click on the right button of the mouse over the object or connector you wish to delete. A popup menu will appear and "Delete" is one of the selections. Choose this and your selected object will be deleted. NOTE: To delete an object, all of its connectors must first be removed.

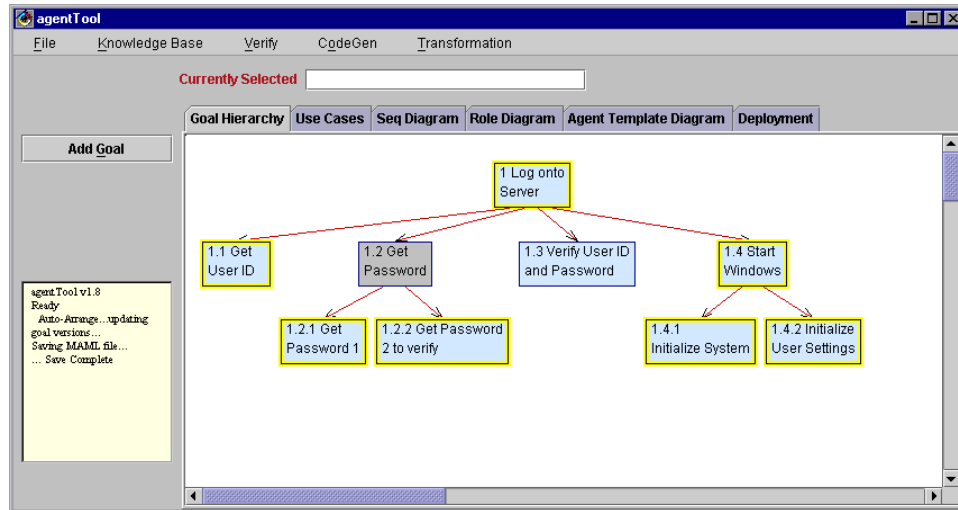
To remove a point from a line, simply drag it on top of another point in the line, or either end of the line. The two points will automatically merge when they are within a certain distance of each other.

4. Printing

By selecting the print options from the File Menu you can print a single panel of your system. Currently there are two options when printing. You can either print the default number of pages determined by panel active width and height, or you can scale that panel down to a single page. Printing is a bit slow at this point in time, so please be patient. Also, be aware that Java printing requires a significant amount of temporary swap space on your C: drive. If your print job does not process within about 1 minute, or you receive full-disk errors, you may need to clear more space. It would be safest to give Java between 75-100 Mb of drive space to operate.

BUILDING YOUR SYSTEM

1. Goals



To add a goal, push the “Add Goal” button. If no goals have been previously added, the new goal will appear on the screen. Otherwise, a goal must be selected to be the parent of the new goal. Click on the chosen goal and the new goal will appear below the selected goal.

To input the description of a goal, use the popup menu to pull up the description window. Input the description and click “Finish”.

Goals must be implemented in a role. Until this has happened, the goal will remain highlighted. In the image above you can see that “1.3 Verify User ID and Password” has been assigned to a role.

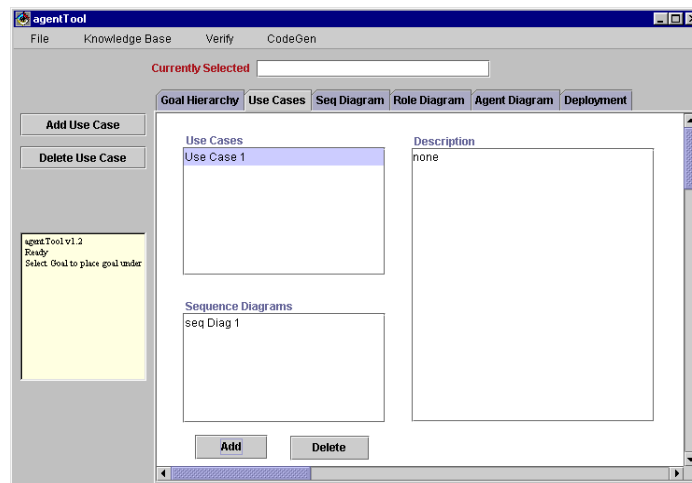
There are goals that do not need to be implemented in a role. These goals are completely implemented through their children goals, so they are called “partitioned”. To set a goal to be partitioned, simply select “Set Partitioned” in the popup menu for that goal. The goal will then be displayed gray and will not appear in any role lists. Goal “1.2 Get Password” is one such goal. If a goal is already assigned to a role, it must be unassigned before you can partition it. The program does not check to make sure a goal has children, so it will be up to you to verify the consistency of your system.

Several new features have been implemented on this panel. First, there is a level locking function that restricts movement of goals to the horizontal. This is intended to visually preserve the tree structure of the goal hierarchy by keeping goals of the same depth on the same level. It can be toggled off and on through the default popup menu. Toggling it back on will lock goals into their current vertical position. The second feature is the Auto-Arrange function in the popup menu. This will process your goal hierarchy and algorithmically arrange the goals into an ordered tree, with parents centered above their children. It is often the case that this will extend the tree beyond the width of the page, in which case the panel will automatically resize. This function does have an undo in case of an error, or if the result is simply undesired. Undo can be run from the popup menu after an Auto-Arrange has been performed and before any new additions have been made.

More new features that have recently been added to deal with reassignment of the goal parents. All of these appear on the popup menu for a goal. To reassign the parent of a goal you would select “Change Parent” from its popup menu and then click on the goal you wish to become the new parent of this goal. You can also assign (and remove) secondary parents. These are handled differently than the primary parent in order to preserve the tree structure of the goal hierarchy. Select “Add Secondary Parent” from your child’s popup menu and then click on the goal you wish to add as a secondary parent. To remove it, select “Remove Secondary Parent” on the child and then click on the goal to remove. If you wish to make a secondary parent the primary, you need only use the “Change Parent” function and the existing primary will be switched to secondary and the selected secondary parent made primary. There are checks in place that will prevent you from assigning a goal’s descendant as its parent, and none of these functions apply to the root goal. However, it is still possible to create cycles in this system, so it is up to you to make sure that you are being consistent with your assignments.

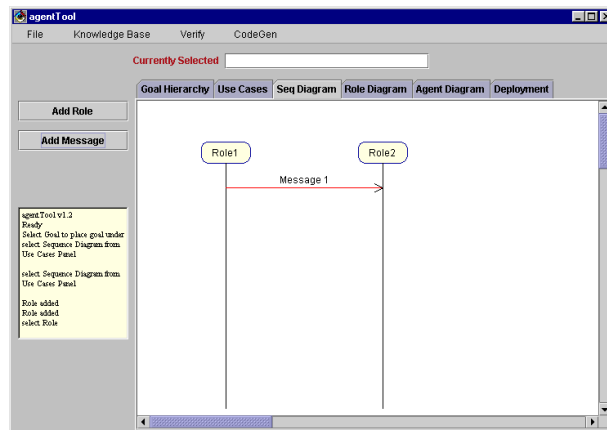
One final feature that has recently been added is the ability to reorder your goals. In reassigning parents it is very common to wind up with goals in an ordering that is not satisfactory. You will find at the bottom of the popup menus two items labeled “←” and “→”. These items respectively will decrement and increment the index of the selected goal within its group of siblings. That is, if a goal is labeled 1.2.3 and you decrement, the goal will be relabeled 1.2.2 and the previous 1.2.2 will become 1.2.3.

2. Use Cases



To add a Use Case, push the “Add Use Case” button. A new Use Case will then appear in the list of use cases. To add a sequence diagram to a use case, select the use case and then push the “Add” button at the bottom of the screen. A new sequence diagram will then appear in the list of Sequence Diagrams.

3. Sequence Diagrams

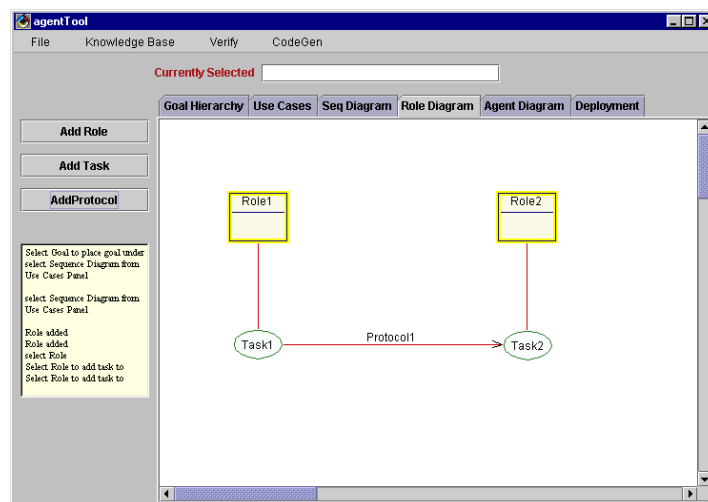


To edit a sequence diagram, first select it in the Use Case Panel. Then select the Sequence Diagram Panel Tab. Now, Roles and Messages may be added to the Sequence Diagram. To add a role, push “Add Role”. A window will appear. If roles already exist in the system, the user may select them for the Sequence Diagram, or create new Roles. When new roles are created in a Sequence Diagram, they are automatically added to the Role Panel. A user can change the role assignment of a particular role instance by selecting “Type” from the popup menu for that role. The Role Selection dialog will reopen, allowing the user to change the role assignment at that point.

Messages may be added by pushing “Add Message”. Then select the two roles that the message will go between. The message will appear between the two line descending from the two selected roles. It may be moved by dragging it up and down.

The lines descending from the roles will adjust their length automatically as more messages are added. If it becomes necessary to increase the length manually, this may be done by opening the popup menu over the vertical line and selecting “Increase Line Length”.

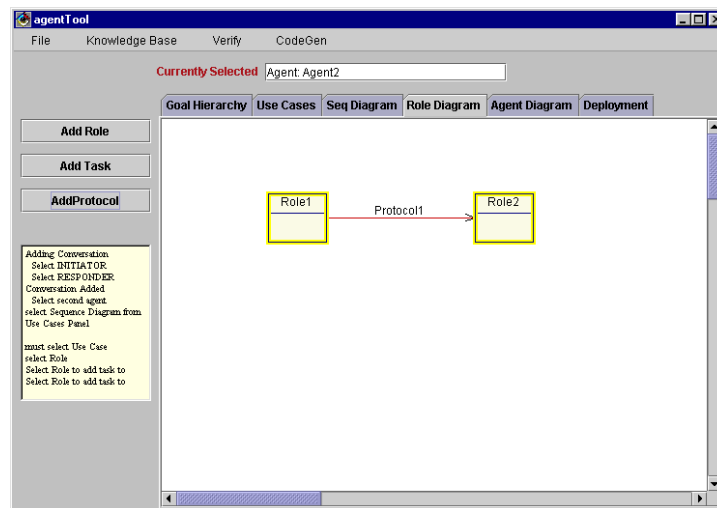
4. Role Diagrams



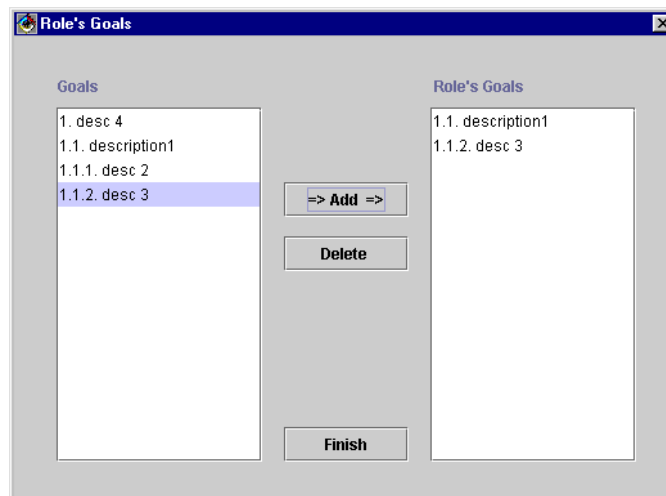
To add a Role, click the “Add Role” button. To add a Task, click the “Add Task” button, and then select the role to which the task will belong. Protocols must be between tasks. To add a Protocol, click “Add Protocol”. Then, select the two tasks that the protocol will go between.

Protocols have an internal/external setting. With the external setting being for tasks between different roles and internal being for two tasks of the same role. They can be changed at any time from the protocol popup menu.

It is possible to hide the tasks and just see the roles and protocols. To do this, click the right mouse button outside of all the objects. The popup menu will have the option “hide Tasks”. By selecting this, the tasks will be hidden and only the roles and protocols will be shown. However, before another protocol is added, the tasks must be made visible again.



To add goals to a role, pull up the popup menu for the selected role. Select “Goals”. A list of the goals will then appear. Add the appropriate goals to the list of goals that this role accomplishes. Then click “Finish”.



Roles must be implemented by at least one agent. Until they are, they will remain highlighted.

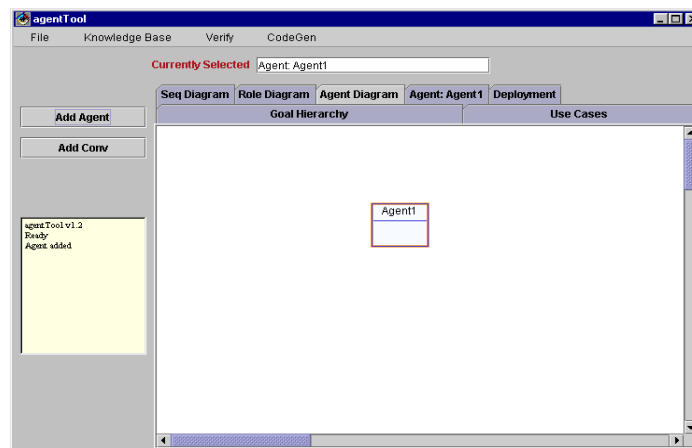
Selecting a task will add a state diagram panel called “Task Panel” which is utilized like all other state diagram panels.

5. Specifying Agents

To specify the roles that an agent implements, pull up the popup menu for the agent and select “Roles”. A window will then popup with the names of the roles that exist in the system. Add the selected roles to the agent's list of roles and click “Finish”. When a role has been implemented by an agent, the role is no longer drawn as highlighted in the Role Diagram.

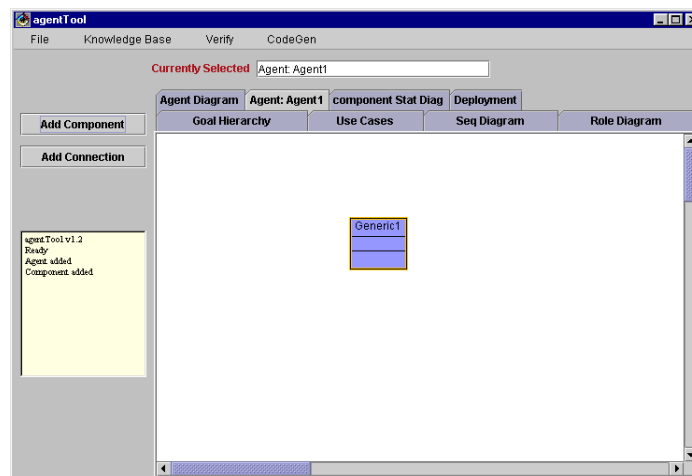
When an agent is selected, a new tab will appear that is named after the agent. This new panel describes the components of the agent. To get to this new panel, select the tab named for the presently selected agent.

5.1 Agent Panel



5.1.1 Components

Each component has a state diagram. To get to this diagram, select the component and then select the tab that says, “component State Diagram” on it.



This panel is manipulated just as the state diagrams are for conversations and will be described later. Now, to describe a component, right click on the component to show the popup menu.

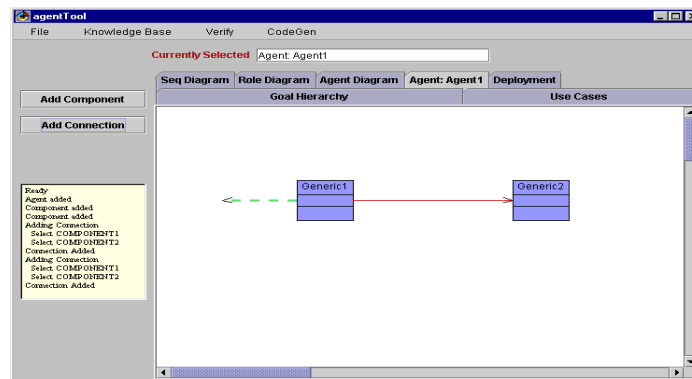
The first option is the preferences. Choosing this will display another window. In this window, the component's name can be changed, its constraints can be set, and whether or not it has a substructure can be set. If it has a substructure, when the component is selected, another tab will appear with the component's name on it and the word "Architecture". This panel describes the substructure of that component. The substructure can be edited just as the Agent Panel was edited.

The next few options in the popup menu for a component deal with the attributes and methods of the component. To add an attribute or method, selected the correct option and then fill in the specifics for the attribute or method that is being added. To edit an attribute or method, select that attribute or method and change it accordingly.

The final option in the popup menu for a component is "Delete". To delete a component, first all the connectors that are connected to that component must be deleted. Then the component may be deleted. The sub-architecture and state diagram will be lost for this deleted component.

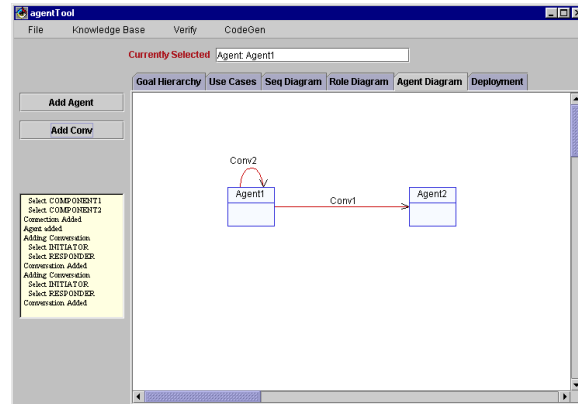
5.1.2 Connectors

There are two types of connectors. The first connects one component to another component. The other connects one component to the framework. To add a connection from a component to the framework, only one component must exist. First, click on the "Add Connection" button. Then, double click on the selected component. The connection will then appear. To move the end point that points to the framework, just click and drag on the end point.



There are three different configurations of the arrows on the connections. It can point to the first component, the second component, or both components. To specify the arrows, right click on the connection. A popup menu will appear. The first three menu items specify where the arrows will point. The first two are the names of the components that the connection connects. Selecting a components name means the connection will point to that component and not to the other. The third is the option of having arrows at both ends of the connection.

6. Conversations

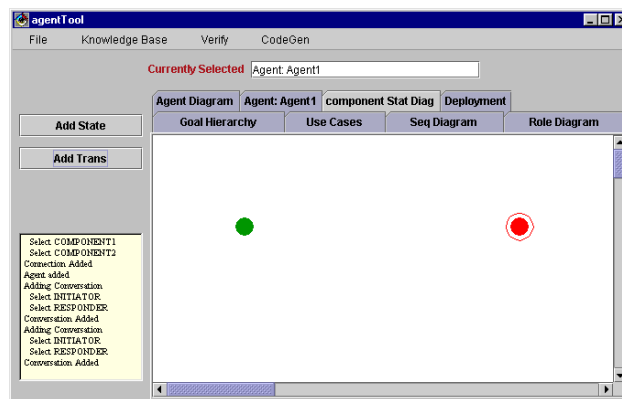


6.1 Specifying Conversations

A conversation can be from one agent to another or from an agent to itself. The conversations name and multiplicity may be set in the preferences window that is reached through the picking the preferences option in the popup menu of the conversation. To change a conversation's orientation, the related options are also in the popup menu.

Every conversation has two diagrams that describe it: the Initiator State Diagram and the Responder State Diagram. Tabs for these will appear when the conversation is selected.

6.2 State Diagrams

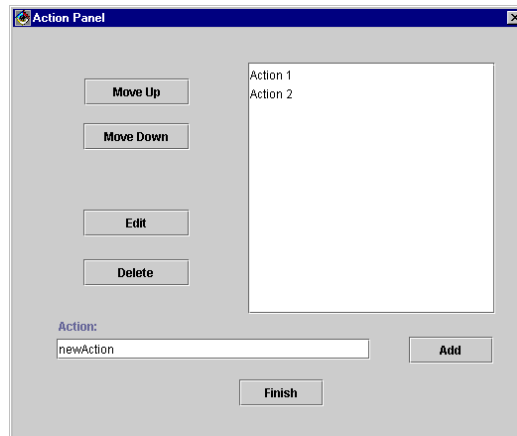


6.2.1 States

Two states always exist in the state diagram. The green circle is the start state and the red circle is the end state. To change the name of a state, select "Properties" from the popup menu of the state. Change the name accordingly and then click the "Apply" button.

To define the actions of a state, select "Define Actions" from the popup menu. In the "Action" text box, type the action. Then click the "Add Action" button. The action will then appear in the large white box that holds the list of actions. Once there are multiple actions in the list, the order may be arranged to the taste of the user. To move an action up or down in the list, select the agent and then click the "Move Up" or "Move Down" button accordingly. To edit an action, select the action. Then click the "Edit" button. The action will then appear in the Action text box. Change the action and then click the

“Add” button. The new version of the action will then replace the old in the list. To finish, click the finish button. The actions will now appear in the state.



The two remaining options on the states popup menu are “Delete Error” and “Delete”. The second is used for deleting the state. Before the state can be deleted, the transitions that are connected to it must first be deleted. The “Delete Error” is used when verifying a conversation. If there is an error in that state, then the state will be highlighted. Once the error has been corrected, remove the error with the “Delete Error” option and then test the conversation again.

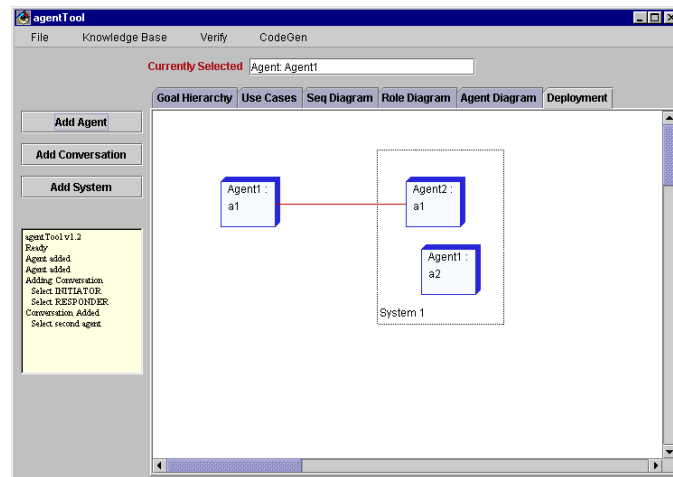
6.2.2 Transitions

To specify the properties of the transition, select the properties option from the popup menu. A new window will appear. Fill in the appropriate text fields. When finished, click “Apply”.

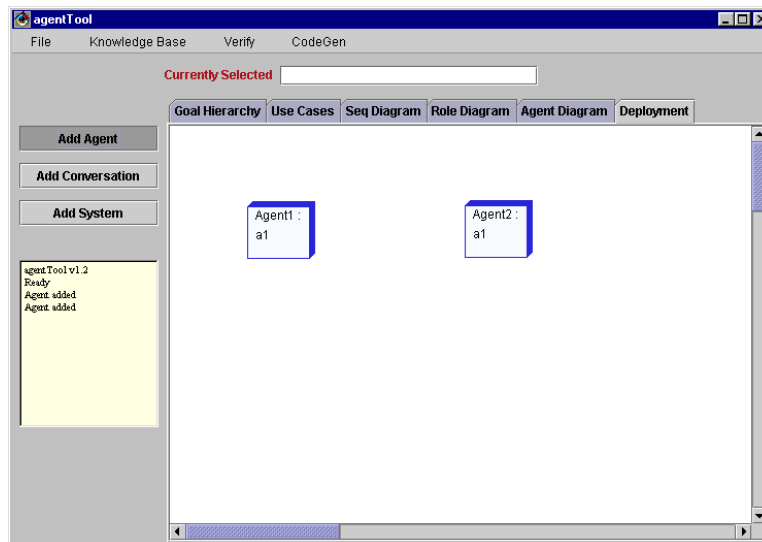
To define the actions of the transition, select the “Define Actions” option from the popup menu. Then define the actions in the same manner as for a state.

A Transition has markers that label the start and end parts of the transition. These options are selectable in the popup menu.

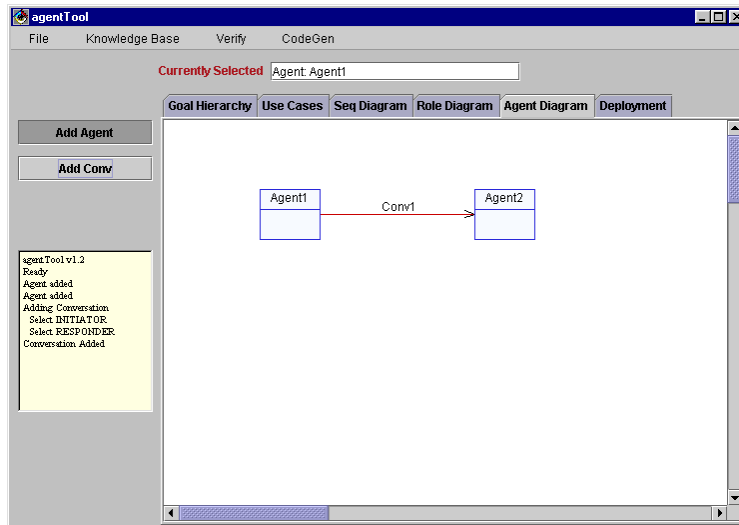
7. Deployment Panel



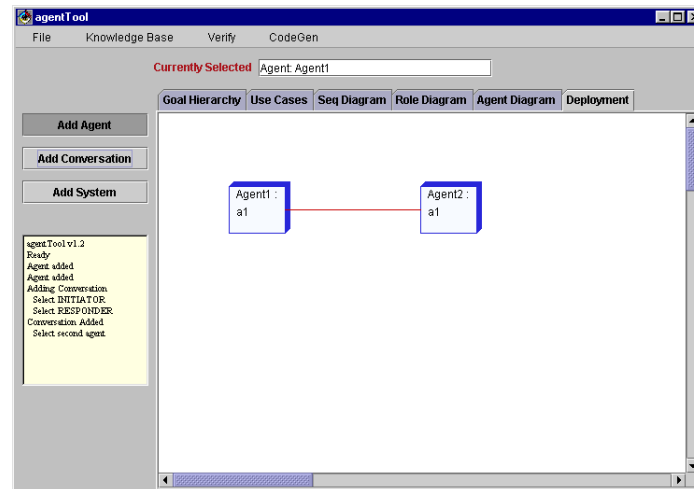
To add an agent to the Deployment Diagram, at least one agent must already exist in the Agent Diagram. If this is true, push the “Add Agent” button and select the correct agent. To add a conversation in the Deployment Diagram, a corresponding conversation in the Agent Diagram must exist. For instance, let Agent1:a1 and Agent2:a1 be agents in the Deployment Diagram.



Now, to add a conversation between these two agents, a conversation must exist between Agent1 and Agent2 in the Agent Diagram.

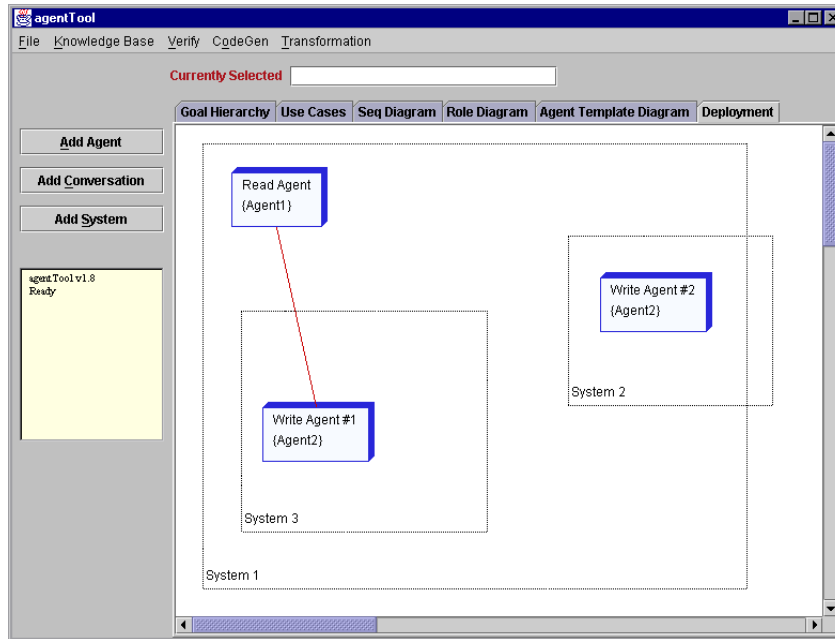


Since Conv1 exists between Agent1 and Agent 2, a conversation may be added between Agent1:a1 and Agent2:a1.



To add a System to the Deployment Panel, click the “Add System” button. Then click and drag the mouse in the diagram to place the system appropriately. The systems can be resized just as a window on your OS. Just “grab” one of the edges or a corner with your mouse and drag it to the correct size.

The systems in the Deployment Panel act as solid units. If you move a system, any agents contained within it will be moved likewise. If you want to move an agent outside the system, just grab it by itself and drag it. You can select multiple objects to move by holding down the shift button as you select items. If you have compound systems (ie – a system completely contained within another system) it will be handled in the same way an agent would be within the larger system. You can picture it like a pyramid. The largest system is the base. By grabbing the base you move everything on top of it. Smaller systems sit on top and can be moved around individually. Agents sit on top of the systems. Conversations float between agents. Any objects that are not completely contained within a system will not be selected unless you select them individually.



In the example image above, you can see compound systems. System 3 is contained within System 1. System 2, however, is not contained by System 1. All three agents are contained within System 1. If we grabbed System 1, all visible objects would be moved except for System 2. If we grabbed System 3, then we would be able to move System 3 and Write Agent #1.

VERIFYING CONVERSATIONS

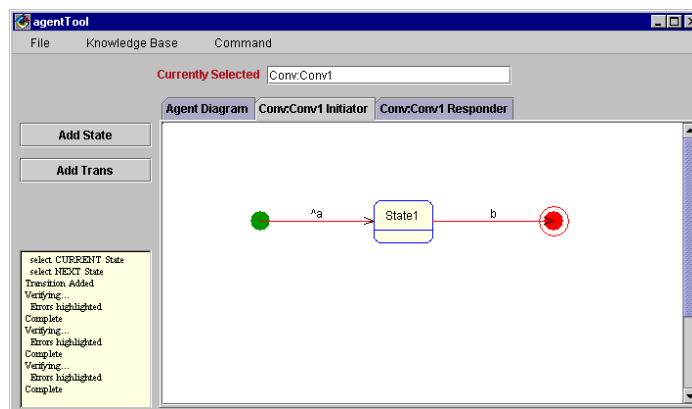
1.1 Verifying Conversations

To verify a conversation, the conversation must first be created. Input the transitions' properties in both the Initiator State Table and the Responder State Table. Once this has been done, select "Verify Conversations" from agentTool's Commands Menu.

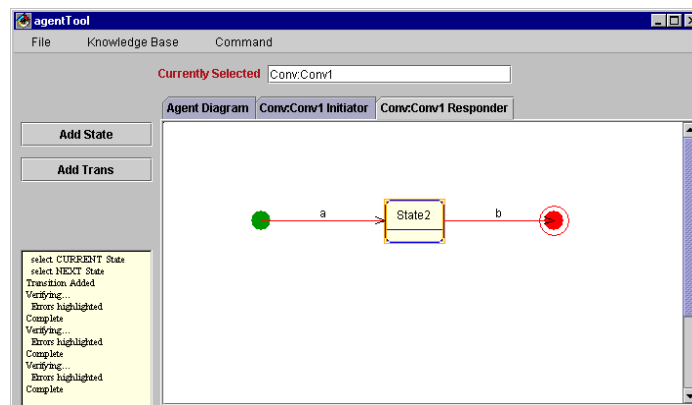
A window will appear called Verification Messages. When the verification is complete, the line

```
***** Testing Completed *****
```

will appear. At this time, you can scroll back up through the Verification Messages window and see if the conversation(s) were verified or if they have errors. If errors exist, the transitions or states with errors will be shown on the screen and will be highlighted in agentTool. A simple example is below.



The initiator of the conversation sends the responder message *a* and receives message *b*.



The responder of the conversation receives message *a* and receives message *b*.
Now, verify the conversation. The following window will pop up.

```

Verification Messages
-----
final state:
-----
#processes: 3
4:   proc 2 (Conv1Responder) line 28 "Goverify" (state 8)
4:   proc 1 (Conv1Initiator) line 12 "Goverify" (state 8)
4:   proc 0 (:init:) line 44 "Goverify" (state 4) <valid endstate>
3 processes created

***** DETAILED DEADLOCK INFORMATION *****

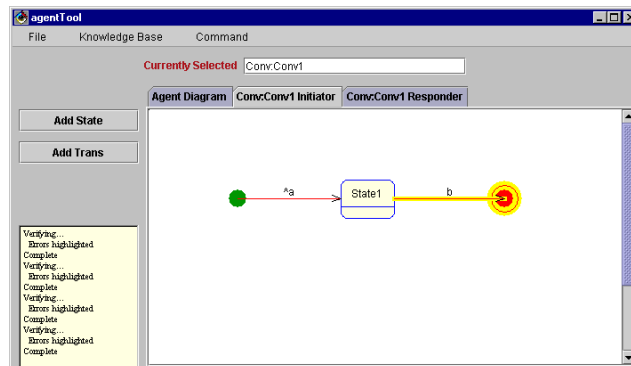
DEADLOCK CONDITION EXISTS IN THE FOLLOWING CONVERSATION:
Conversation Name = Conv1
Participant Name = Responder
Current State = State2
State Transition = b

DEADLOCK CONDITION EXISTS IN THE FOLLOWING CONVERSATION:
Conversation Name = Conv1
Participant Name = Initiator
Current State = State1
State Transition = b

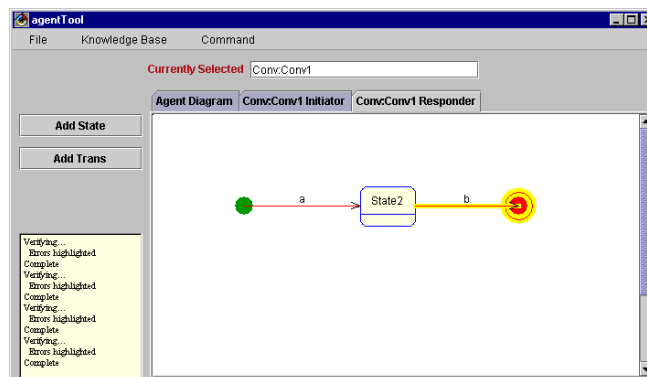
***** TESTING COMPLETED *****

```

The verification window shows that the conversation is deadlocked. Now, look at the agentTool panels after verification.

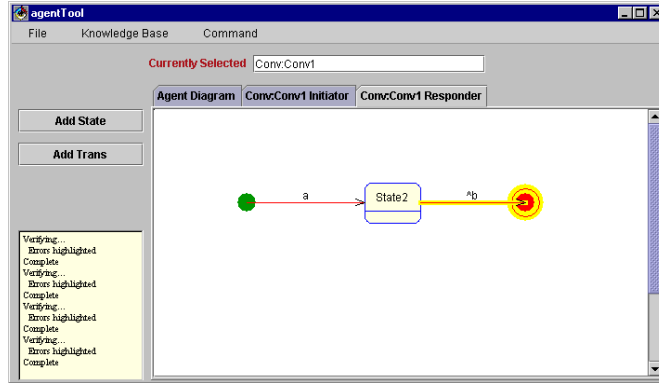


The Initiator Panel has the transition where b is being received highlighted as well as the end state. The transition b is highlighted because b is never received. The end state is highlighted because since b is never received, the end state is never entered.



The Responder Panel also has a transition and the end state highlighted. Once again, b is never received and the end state is never entered.

So, let us change the Responder State Table so that b is sent in the second transition.

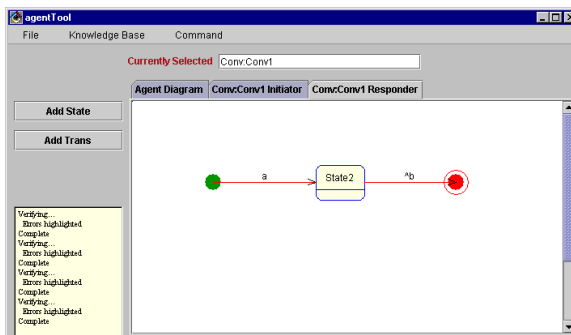
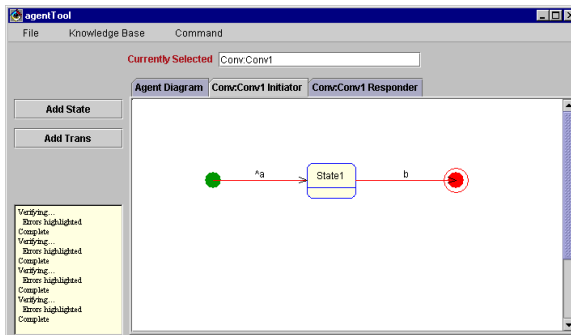


Now, by re-verifying, the changes can be checked to see if the error was corrected.

The screenshot shows the 'Verification Messages' window. The text inside reads:


```
PLEASE STAND BY... TESTING CONVERSATIONS...
    !!!!!!!!!!!!! OUTPUT OF SPIN ANALYSIS !!!!!!!!!!!!!
    Analysis Completed... Evaluating Analysis...
    ***** OUTPUT FROM DEADLOCK CHECK *****
    CONVERSATION IS NOT DEADLOCKED!!!
    ***** OUTPUT FROM PROGRESS CHECK *****
    CONVERSATION DOES NOT HAVE UNUSED STATES!!!
    ***** OUTPUT FROM SIMULATED RUN *****
    No trace available
    ***** TESTING COMPLETED *****
```

The verification window shows that the conversation is not deadlocked, they do not have any unused states, and so they are correct. The state diagrams have also been changed:



The errors have been erased.

2. Set up for Verification Purposes

2.1 Within VisualAge

Begin by ensuring the resource files are loaded into VisualAge. When importing the resource files from a jar file, the files are placed in a `project_resources` directory. For example, if VisualAge is installed in the `c:\IBMJava2` directory, and your project name in VisualAge is `agentTool`, then the resource files would be placed in the `c:\IBMJava2\ide\project_resources\agentTool` directory. All of these files and directories must be copied into the `c:\IBMJava2\ide\program` directory for the program to work correctly.

When executing the verification routines from within VisualAge, the user directory has to be specified by setting the appropriate system properties. This is done by right clicking on the `ATEditor` class and selecting the properties menu option. In the system properties box, enter the path to the program directory where the resource files were copied. For example, using the directory above, the command would read `user.dir=c:\IBMJava2\ide\program`.

2.2 Outside VisualAge

The verification routines can be tested without invoking `agentTool` by running the `VCRun` class in the `Verify` package. Before `VCRun` can be invoked, the `user.dir` command must be setup in the system properties for this class as previously mentioned.

AGENT TOOL V1.8 ADDITIONS (2-9-01)

- Deployment Panel System enhancements – Reworked the deployment panel's selection and movement functionality. Implemented new standard layout involving a principle of system layers and containment. Agents that exist within systems are recognized as part of a system and dragged with the system box automatically. Nested systems are recognized as subsets of a larger system and are moved with the larger system box automatically. We retain the functionality of moving singular items separately regardless of position, with agents being primary select items, nested systems next, then base systems. We also retain the ability for multiple item selection. (2-12-01)
- Printing – The panel printing functionality has been implemented now. The default function to print all pages has been implemented, as well as a function to scale the panel to one page. Further enhancement to the page layout is in development. (2-9-01)
- Unique Keys now assigned to all objects. This allows all objects to be loaded and saved as unique objects, regardless of name. This eliminates problems with identically named objects being mislinked in our Vectors. It also eliminates the need for unique naming conventions that we had implemented prior to now. Not all these conventions have been abandoned, but they are no longer critical to the operation. (2-5-01)
- Goal Parenting – Goals now can have multiple parents (with one primary parent). All the functions to add, remove, and otherwise manipulate them have been added. One can also reassign the primary parent at will, allowing quick restructuring of the goal hierarchy. Finally, we have implemented functions to increment and decrement the goal numberings to compensate for the jumbling involved with our reassignment processes. (1-29-01)
- Role reassignment in Sequence Diagram – Now our sequence diagram roles can have their role type changed by the user on the fly. (1-19-01)
- Message Line Extension – The message lines in the Sequence Diagram can be extended downward at will, and will also extend themselves when appropriate numbers of messages have been added to necessitate the extension. (1-12-01)
- Goal level lock – by default goals can now be moved only horizontally. This is intended to keep vertical cluttering down and maintain the visual appearance of the tree structure when adding new goals. This can be toggled off in the popup menu if necessary, since older MAML's were created without level locking and are generally not organized into a tree. Goals can be resituated manually by turning off the level lock, or one can use the new auto-arrange function.
- Goal Depth Labeling – goals of version 1.8 or greater are now automatically labeled vertically instead of horizontally. Prior to this, a goal description had at most two lines of text which spanned horizontally as far as necessary. Now these labels are set to a limited width with an expandable height. This adaptation was done to eliminate extreme horizontal spanning in the tree structure. Goals prior to v1.8 are labeled as before for consistency with their existing relative layout.
- Auto Arranging for the Goal Hierarchy – added popup menu and functions to handle auto-arranging goals. It rebuilds the goal placement into an orderly and visually uncluttered structure using the internal tree layout of the goals. It recursively arranges a goal's children and then places the goal in the middle above its children. This function also will expand the display as necessary in order to compensate for width and height settings. It will also update all the goal versions to 1.8 to utilize the new depth labeling feature.

- Undo – added an undo function for the auto-arrange operation. This is currently the only automatic main operation that cannot be undone by simply deleting an item or renaming or resetting a property. It provides the basis for a MAML-based function that can be extended to undo nearly any operation.
- Panel Size setting on startup – As a side effect of the Auto Arrange function, it became common for Goals to extend beyond the default window range. Now there is a function in place that will set the size of the display after reading the MAML. The function tests all visible elements and finds the minimum width and height the screen must be to accommodate the display.
- Graphical enhancements to segmented lines - added blue circles at each vertex when selected to aid in grabbing and shaping lines.
- Transform Menu – added a transform menu with menu items for Add Agent Components, Annotate Component State Diagrams, and Create Conversations. These functions will be scripted later to suit the needs of 1Lt. Clint Sparkman in his research.
- Internal vs. External Protocols – added menu popup, internal data members, graphical changes, and MAML I/O to keep track of the locality of Protocols.
- Goal Partitioning – added menu popup, internal data members, graphical changes, and MAML I/O to keep track of the partitioning of goals. We also updated the assignment of goals to roles so that a partitioned goal is not assigned to a role, and an assigned goal cannot be partitioned until it is unassigned.
- Start State / End State Transition – added menu popup, internal data members, graphical changes, and MAML I/O to keep track of the new start / end status markers in Transitions.

AGENT TOOL V1.8 BUG FIXES

- Move operations changed from delta functions to actual position assignments. That is, instead of moving the object however far the mouse moved, we put the object where the mouse currently is, or wherever is closest within the object's boundaries. This eliminates the problems encountered with getting the mouse shifted off the object. (1-01)
- Memory consumption has been significantly reduced by removing the panel images and rerouting all paint and update functions properly. The update functions now are unused. When repaint is called, each panel is updated properly now using global graphics objects and images. (1-12-01)
- The popup menu malfunction has been patched up. This occurred mainly on the Sequence Diagram Panel. The error would be triggered when no objects were selected because it would attempt to show an empty popup menu. This would cause the menu to be disabled and would cease to activate. This was fixed by checking to make sure that an object was selected before trying to show() the popup. (1-5-01)
- The algorithms for attaching lines to squares and circles have been fixed to more properly arrange them with respect to relative screen position. (12-20-00)
- The algorithm that handles tab addition/deletion has been reworked. (12-4-00)
- Fixed the Menu Clipping error. When a menu was opened in front of the panel display of the Agent Tool Editor, it would be clipped off by the white box in the display. This was fixed by moving the call to superclass update function to the very end of the main update. (11-29-00).
- Dragging the mouse while adding objects in the Role Panel caused many errors in insertion and deletion, including reassigning the last task to a different role. This has been fixed, and the other panels updated to prevent possible bugs in them. (11-29-00).
- Components now have required unique names so that they are tracked properly. (11-29-00).
- Adding lines to a circle (esp. in state diagram) pointed line to center of circle instead of rim. This is now fixed. (11-28-00)
- Components are now counted properly, accounting for sub-architectures as well. Components added are initially numbered according to the current count, including all sub-components as well. (11-28-00)
- File operation cancel – you can now successfully cancel a save or load file operation, and also fixed the subdirectory issues I had with JFileChooser (11-14-00)
- Triple Quote MAML errors – Many MAMLs were being corrupted when saving a null String. The way the read MAML function worked was to read a token in a given location and set the String value to it. However, an empty String would cause the given location to hold a quote instead of a null value, and so the String would be set to a quote. On resaving, the string would save a triple quote. Once that had been done the file could no longer be parsed correctly. To fix this we set a check in the reload that will load a quote mark as a space instead. Second, we changed the initial string from an empty string to a space mark. (11-14-00)
- Goal Happy – in adding goals to a role, only the goals not currently assigned are displayed in the selection list. Upon adding a goal, that goal is removed from the selection list. Upon

deleting a selected goal, it is added back into the selection list. (11-7-00) Also, for agents adding roles, roles are not marked like goals, so all roles are loaded, but only those not already selected are displayed. This makes it possible to assign a single role to multiple agents, which is desired. (11-28-00)

- Component Attribute Editor – now fully tracks the data entered, stores it back into the component attributes, and loads it back into the editor on reload. (11-3-00)
- Eliminated several never-ending operations. Once started, these operations would never end until completed. This often interfered with the logical flow of the interface. We set conditions which would cancel each of these operations : (11-2-00)
 - State Diagrams -> Add Message
 - State Diagram -> Add Transition
 - Role Diagram -> Add Task
 - Goal Hierarchy -> Add Goal
 - Agent : AgentX -> Add Component Connection
- Segmented lines – fixed some errors with selection boundaries, vertex creation, segment processing. (10-31-00)
- Role numbering – roles loaded from a MAML are now counted properly and additions begin from the correct index. (10-31-00)
- Element spacing – added a basic spacing algorithm to Roles, Agents, Components, etc. so that they are no longer generated on top of each other. They may still overlap other elements by circumstance if you have moved things around, but at least if you create multiple elements at one time, they are separated and clearly visible. (10-31-00)
- Protocol Description popup – it has now been scripted to actually do something. (10-31-00)
- Protocol vs Role-Task Connectors – these two items now have different colors in order to distinguish between them visually. (10-31-00)
- Removed redundant commands in Component Connector popup (10-31-00)
- Open Space Selection – in the Goal Hierarchy panel, selecting open space now unselects all goals. (10-31-00)
- “Delete use case” button – now checks for selected use cases (10-27-00)
- Start State / End State class cast exception – you can no longer define Actions for a start state or end state in the state diagrams. They were never supposed to allow that as an option. (10-25-00)
- Asynchronous input with synchronous update – Changes to elements were not being graphically updated because the input dialog boxes were in asynchronous mode. We changed them to synchronous so now the update occurs after the dialog is closed. (10-25-00)
- Arrow Heads – The arrowheads previously did not initialize properly, causing them to draw all over the screen. They work now.
- Component Diagram Arrow Issues – Arrow information was being lost in the AgentX Component Diagram causing all but one arrow to be drawn and none ever deleted. All arrows behave properly now.

- Component Diagram External Arrows - The external conversation arrow(green dotted arrow) used to draw from the left side of a component at a fixed length, this sometimes caused its head to go beyond the window border causing it to be forever lost, until deleted. They now initialize to the top left corner guaranteeing that that will not happen anymore
- Deployment Systems – When reloading MAML files with systems in the deployment panel the systems would tend to drift due to a mispointing index in the loadMAML function. They initialize properly now.
- Extra Conversations – There were issues on various panels where the internal counters would count extra conversations coming off of a box, thus causing you to be unable to delete said boxes even when all visible conversations were gone. These 'extra' conversations were tracked down and dealt with harshly.