Modality Manager
User Manual

(for internal use only)

Boi Sletterink
boi@linuxfan.com

June 10, 1999

Contents

Chapter 1

Introduction

This document describes how to start the multimodal system within the DISCIPLE desktop.

Figure 1.1 shows an overview of the different hardware and software components, and the connections between them.

The modality interfaces connected to the modality manager make TCP connections to their OS-level drivers. This allows the NT machine to use the gaze tracker and glove or the SUN machine to use the speech recognizer. In principle, the desktop can be run on both since it is written in Java. In practice, the 3D terrain map will not work on the SUN because it requires Java3D. All other applications work fine.

To use the multimodal system in the DISCIPLE system, you have to perform these steps:

1. Start the DISCIPLE system
2. Open a workspace
3. Drop the multimodal manager into the workspace
4. Start any modality drivers and drop their interfaces into the workspace

The manager, modality interfaces, and modality drivers are described below. Use of the DISCIPLE system is covered by the DISCIPLE manual which can be found at:

`http://www.caip.rutgers.edu/disciple/software/`

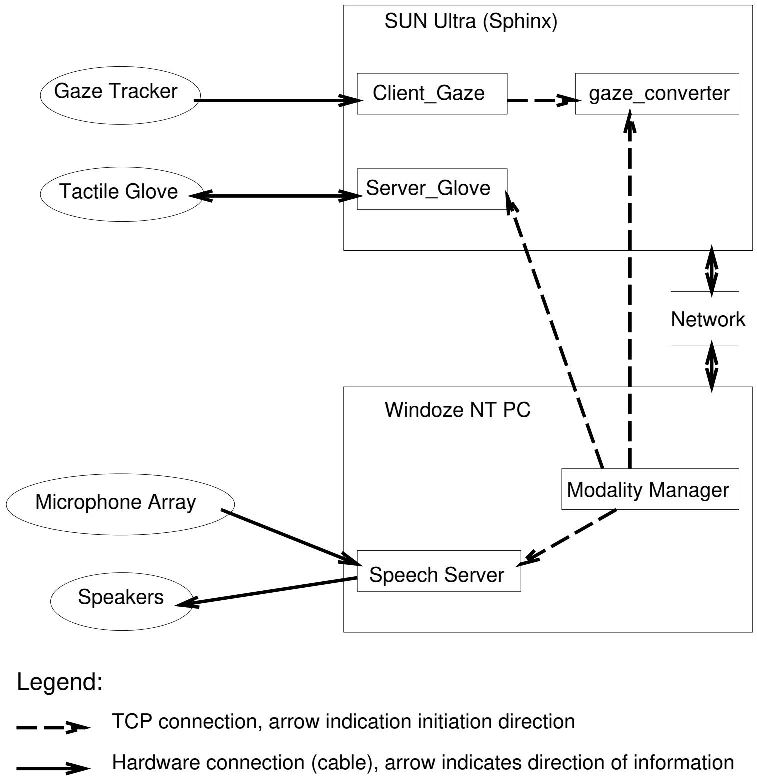

Figure 1.1: Components of the multimodal workstation and relationships

Chapter 2

Multimodal Manager

2.1 Starting the MM

The multimodal manager is started by dropping the *Octopus* bean into the workspace. Typically, the bean browser will show a tree like this:

- file:/C:/stimulate/jars/0whiteboard.jar
- file:/C:/stimulate/jars/areamap.jar
- file:/C:/stimulate/jars/cnav.jar
- file:/C:/stimulate/jars/multimodal.jar
  - EDU.rutgers.stimulate.MITools.gaze.GazeTracker
  - **EDU.rutgers.stimulate.MITools.octopus.Octopus**
  - EDU.rutgers.stimulate.MITools.speech.SpeechInterface
  - EDU.rutgers.stimulate.MITools.keyboard.KeyboardInterface
  - EDU.rutgers.stimulate.MITools.glove.TactileGlove
- file:/C:/stimulate/jars/wbenhance.jar

Select the highlighted bean and drop it into the workspace. It is advisable to move it to a convenient spot since there are going to be more beans in the workspace, and after setting the multimodal system up, you will usually not have to do anything with MM anymore.

2.2 Using the manager

2.2.1 Configuration panels

Click on the picture of the Octopus to get a list of loaded modality interfaces. Clicking on any of these in the list will pop up a configuration dialog specific for that interface.

2.2.2 Speech commands

Once the speech recognition is online, you can use the speech commands listed in table 2.1.

create <class> bean	Drops a new instance of <class> in the workspace
switch *use gaze* switch to <bean>	Switches to another bean
iconize <bean>	iconizes (minimizes) a bean
play deaf disable speech stop listening	Blocks speech recognition input
hello computer	Are-you-there command
listen enable speech	Resumes speech recognition input

Table 2.1: Voice commands for the Octopus

Chapter 3

Speech Subsystem

The speech subsystem uses Mircosoft Whisper. Whisper comes with the SpeechSDK, which we used to build a "speech server".

3.1 Speech Server

The speech server binary can usually be found in the base directory of the DISCIPLE desktop or in the directory `SpeechServer2` below it. If you cannot find the speech server program, search the disk for "SpeechServer.exe". There should also be a copy on:

```
\\CAIPNT\users2\Stimulate
```

Before dropping in the speech interface into the workspace, you want to start the speech server. When you start the speech server, it will show a dialog to choose the recognition engine. Usually, the first one is correct and you choose OK. Then, the main panel pops up. *DO NOT* press OK here unless you want to *exit* the speech server. If you have trouble with the recognition, try the `User Training` button to lower the rejection rate. If you have a slow machine, you might try to lower the accurracy.

3.2 Speech Interface

To use the speech subsystem, drop the SpeechInterface bean in the workspace. Typically, your bean browser will show something like this:

- file:/C:/stimulate/jars/0whiteboard.jar

- file:/C:/stimulate/jars/areamap.jar
- file:/C:/stimulate/jars/cnav.jar
- file:/C:/stimulate/jars/multimodal.jar
  - EDU.rutgers.stimulate.MITools.gaze.GazeTracker
  - EDU.rutgers.stimulate.MITools.octopus.Octopus
  - **EDU.rutgers.stimulate.MITools.speech.SpeechInterface**
  - EDU.rutgers.stimulate.MITools.keyboard.KeyboardInterface
  - EDU.rutgers.stimulate.MITools.glove.TactileGlove
- file:/C:/stimulate/jars/wbenhance.jar

Select the highlighted bean and drop into the workspace. When the speech server is already running on the same machine, it will automatically connect. If not, connect to a running speech server (start one if necessary) by:

1. click on the picture of the octopus in the Octopus bean.
2. click on "SpeechInterface" in the popup box.
3. fill in the appropriate hostname; the default, `localhost` refers to the machine where the desktop is running on.
4. fill in the appropriate port number; unless you changed and recompiled the speech server, the default is correct.
5. click "Connect".
6. if successful, click "Close".

A successful connection will be rewarded by the text-to-speech system saying "speech feedback online" followed by "grammars loaded".

3.3 Trouble shooting

For yet unknown reasons, the Microsoft Whisper speech system is somewhat unstable. Also, the Octopus may not always work as some people would expect. If the computer appears to be deaf, try one of the following:

- say “computer listen”.
- check the SpeechServer main dialog; it must show an active grammar (#0 means only the manager is active, any other number refers to an application), the “Active” checkbox must be checked. Sometimes, unchecking and re-checking this box may help.
- if it responds to “computer listen” but not to an application-specific command, try re-selecting your application in the workspace.
- if some commands are hard to perform, try lowering rejection rate on the speech server.
- check the Windoze volume recording controls. If you can check the volume with a “VU meter” in the dialog box, make sure that it responds to the microphone (or microphone array) but it should *not* get into the “red” area.
- close any Windoze applications that use the soundcard.
- restart the speech server. If you cannot reconnect, you will have to restart the desktop and multimodal system too.
- try recording sound using the sound recorder. If it sounds clear and undistorted, but the speech recognizer does not respond at all or only to the first few tries, ask your friendly local admin to re-install the SpeechSDK. For some systems, this is needed as often as every two weeks.

Chapter 4

Tactile Glove

The glove interface in the desktop connects to a glove server program on a UNIX machine. To use the tactile glove in the DISCIPLE desktop, you must first start the glove server and then connect the glove interface to it.

4.1 Glove Server

To start the glove server, log onto the UNIX machine (at the moment, the machine `Sphinx`, as user `mickeys`) that is connected to the glove, open a console window, and type the following:

1. `cd bin` *(or whereever the binaries are)*
2. `./Server_Glove`

If it complains about a socket being in use, there is either already a copy running or the last time the program was run, it did not shutdown properly. In the last case, wait for a few minutes and retry.

If the program complains about not being able to read a port or something not responding, turn off the Polhemus and RM-II boxes, wait for a ten seconds, turn them on again, wait for a minute and try to run the program again. Note that the Polhemus takes at least 11 seconds to boot and the RM-II box approximately a minute.

4.2 Glove Interface

Drop in the TactileGlove bean. The bean browser will look like this (the TactileGlove is highlighted):

- file:/C:/stimulate/jars/0whiteboard.jar
- file:/C:/stimulate/jars/areamap.jar
- file:/C:/stimulate/jars/cnav.jar
- file:/C:/stimulate/jars/multimodal.jar
  - EDU.rutgers.stimulate.MITools.gaze.GazeTracker
  - EDU.rutgers.stimulate.MITools.octopus.Octopus
  - EDU.rutgers.stimulate.MITools.speech.SpeechInterface
  - EDU.rutgers.stimulate.MITools.keyboard.KeyboardInterface
  - **EDU.rutgers.stimulate.MITools.glove.TactileGlove**
- file:/C:/stimulate/jars/wbenhance.jar

Alternatively, if the speech interface is already active, you can say "create tactileglove bean". Then, connect to the glove server by doing the following:

1. click on the picture of the octopus in the Octopus bean.
2. click on "TactileGlove" in the popup box.
3. fill in the appropriate hostname; the default, `Sphinx` refers to the SUN machine connected to the Polhemus and RM-II boxes in our multimodal workstation setup.
4. fill in the appropriate port number; unless you changed and recompiled the glove server, the default is correct.
5. click "Connect".
6. if successful, click "Close".

If the connection was successful, the glove cursor will stop circling on the screen and the glove will take over cursor control.

Note: force-feedback will only work if there is a compressor connected to the RM-II box.

Chapter 5

Gaze Tracker

The gaze tracker is the most tricky input to setup of the three. It goes in four easy steps:

1. Start the socket converter (explained in section 5.1).
2. Start the gaze reader (also explained in section 5.1).
3. Drop in the gaze interface and connect to the server.
4. Start the gaze tracker processor (the extra PC) and calibrate.

Using the gaze tracker processor and calibration is considered to be out of scope for this document; consult your friendly local lab assistant. This document only describes the software specific to the modality management system.

5.1 Gaze Server

The gaze server is split into two parts: a gaze tracker reader, called `Client_Gaze`, and a program that converts the client into a server, the `socker converter`. You can see the socket converter as a the equivalent of a female-female connector adapter to turn a male connector into a female one.

Both programs run on Sphinx. To start the programs, log in as described in section 4.1, and open two console windows. In one, type:

1. `cd bin` *(or whereever your binaries are)*

2. `./gaze_converter`

Note that the gaze_converter program is a script which runs a Java program (consisting of the classes `GazeConverter` and `SocketConverter` in the package `EDU.rutgers.stimulate.tools`).

Then, in the other window, type:

1. `cd bin` *(or whereever your binaries are)*
2. `./Client_Glove`

The window running the socket converter should now mention that the service connection is established.

5.2 Gaze Interface

Drop in GazeTracker bean. The bean browser will look like this (the GazeTracker bean is highlighted):

- file:/C:/stimulate/jars/0whiteboard.jar
- file:/C:/stimulate/jars/areamap.jar
- file:/C:/stimulate/jars/cnav.jar
- file:/C:/stimulate/jars/multimodal.jar
  - **EDU.rutgers.stimulate.MITools.gaze.GazeTracker**
  - EDU.rutgers.stimulate.MITools.octopus.Octopus
  - EDU.rutgers.stimulate.MITools.speech.SpeechInterface
  - EDU.rutgers.stimulate.MITools.keyboard.KeyboardInterface
  - EDU.rutgers.stimulate.MITools.glove.TactileGlove
- file:/C:/stimulate/jars/wbenhance.jar

If the speech interface is active, you can also drop in the GazeTracker bean by saying "create gazetracker bean".

The next step is to connect the interface to the server:

1. click on the picture of the octopus in the Octopus bean.

2. click on “GazeTracker” in the popup box.

3. fill in the appropriate hostname; the default, `Sphinx` refers to the SUN machine connected to the gaze tracker processor in our multimodal workstation setup.

4. fill in the appropriate port number; unless you changed and recompiled the socket converter, the default is correct.

5. click “Connect”.

6. if successful, click “Close”.

If the connect was successful, the gaze cursor should disappear or stay stable in the middle of the workspace window, until the gaze tracker is activated on the processor (the “gaze tracker computer”).

Chapter 6

Multimodal Applications

This chapter briefly describes the multimodal applications. It is mainly limited to listing the voice commands.

6.1 Whiteboard and Areamap

To make the whiteboard and areamap (mission planning extension of the whiteboard) multimodal, the `WhiteBoard Enhancer` must be dropped into the workspace first. This is the bean called `WBEnhancer` and can be found in the WBEnhance jarfile. Any whiteboard or areamap dropped in after the enhancer will be enhanced with a multimodal extension.

The voice commands available for both are:

- new document
- create line/rectangle/other tools

The areamap has a few more commands:

- load <scenario> scenario
- show weather data
- show satellite image

Currently, these commands only work for the Baguio area. You can only load the Baguio scenario and weather data will always be about this region.

There is work in progress to have more scenarios and live weather for those areas.

Also, the areamap can use the gaze tracker to drop in resource icons instantly (at the spot where the user is looking when saying "create <unit>") and to identify objects by gaze when saying "identify".

6.2 ThreeDeeMap

The 3D map has this bizarre classname because it makes it easier to use voice commands to drop it into the desktop. It uses the tactile glove for 3D pointer input and users can grab objects with it. Voice commands are used to navigate:

1. move forward/backward
2. turn left/right
3. look up/down
4. show view of <object>

6.3 Medical

The medical application is an image guided diagnosis system to assist doctors in diagnosing possible leukemia. To diagnose a sample image, do the following:

1. load an image (through the menu or by saying "open image $< 0..9 >$").
2. select a brightly colored cell by drawing a box around it.
3. segment the image (the eye-button or say "segment").
4. right-click on the cell kernel in the segmented image.
5. press the "Do Search!" button or say "search the database".
6. impress viewers by showing that the first image matches the correct leukemia class. This is true in most cases.