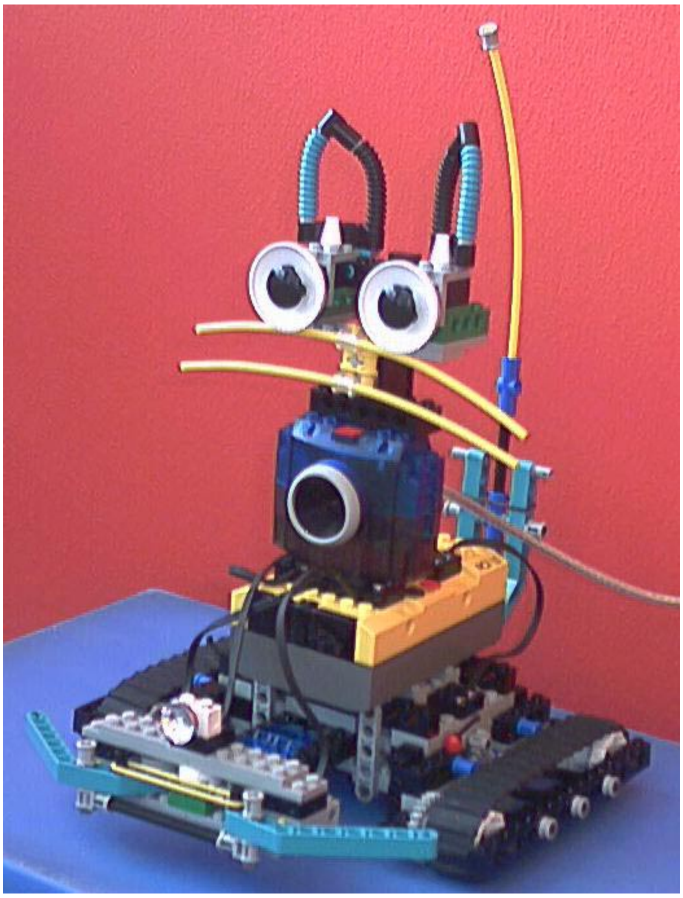
MAELIA

Multimodal Application for Extensible Lego Intelligent Agent



TU DELFT

April 1st to August 31st 2002

User Manual

Guillaume Barraud & Priam Pierret	Department : KBS	August 2002

0. TABLE OF CONTENTS

1. Introduction

This document should allow any person eager to use the application we developped to be able to understand its features and in particular to assimilate the process control of the language of the robot cat (in summary: CCL for Cat Command Language).

By traversing this document you will thus find a description of the various functionalities of MAELIA as well as a precise description of the language of the cat, its syntax, its grammar, its vocabulary...

2. Maelia, the application

2.1 General presentation

MAELIA is the application we developed during our project of end of studies. It makes it possible to order the robot cat we built.

2.2 The catch in hands

It is done using the hands.

2.3 Main features

2.3.1 Direct orders:

The majority of all the orders of the cat are accessible via the language from the cat. It is thus necessary to learn a little on the syntax and the grammar of the language of cat (CCL) to be able to exploit the functionalities of them. Some orders are also accessible by other processes, in addition to their ID in the language. Let us take the example of "takePicture", this order takes a photograph with the camera. It can be started as well by pressing the red button on the camera or by clicking on the button "takePicture" on the window of the camera.

2.3.2 Save of a sequence of orders:

Any order cat can be saved in a file XML and can be thus pointed out thereafter. To save a sequence of orders, subject it to the cat via the text interface or via the icons interface, possibly test it to check that it functions correctly, then click on the Save button.

This order is also accessible via the File menu - > Save This Command as Xml.

2.3.3 Removal of a complex order:

Sometimes, in order to reduce the graphic interface of the icons, it is necessary to remove some useless or unutilised complex orders.

This functionality is available via the File menu - > Delete Complex Command.

2.4 Some windows

See report for more explanations about the MDI interface.

3. The Cat Command Language (CCL)

The language of the cat makes it possible to order the robot cat quite simply. You just have to compose a sentence and to validate it.
This sentence can be written either with the words of the language, or with command icons. These two languages (text and icons) are completely equivalent, that is to say that to each word of the language text matches an icon.

3.1 Basic commands

The basic commands form the primary bricks of the language of the cat. They are the basic orders of the robot which match to its basic functionalities such as going straight, lighting the lamp or playing some music, taking a picture or saying something.

Some of these basic commands are followed by some parameters, not optional for the most part of them.

Here are all basic Commands of the CCL :

BasicCmds

driveForward

This command triggers the movement forward of the robot. It accepts a parameter : a distance in centimeters.

driveBack

This command triggers the movement backward of the robot. It accepts a parameter : a distance in centimeters.

rotateRight

This command triggers the rotation by right of the robot. It accepts a parameter : an angle in degrees.

rotateLeft

This command triggers the rotation by left of the robot. It accepts a parameter : an angle in degrees.

turnRight

This command triggers a rotation with one wheel of 90 degrees on the right.

turnLeft

This command triggers a rotation with one wheel of 90 degrees on the left.

driveForwardDuring

This command triggers the movement forward of the robot during the specified time. Time must in seconds (s).

driveBackDuring

This command triggers the movement backward of the robot during the specified time. Time must in seconds (s).

rotateRightDuring

This command triggers the rotation by right of the robot during the specified time. Time must in seconds (s).

rotateLeftDuring

This command triggers the rotation by left of the robot during the specified time. Time must in seconds (s).

stopMoving

This command stops any movement of the robot cat.

startStroll

This command triggers a stroll for the cat. It could be able to detect contact on the walls and to react.

endStroll

This command stops the stroll for the cat. It could not be anymore able to detect contact on the walls, neither to react.

setLightOn

This command sets the light on.

setLightOff

This command sets the light off.

setPowerMotor

This command sets the power of the motors. It must be an integer between 0 and 7.

setPowerDownTime

This command sets the powerdown time of the robot.

playMusic

This command makes the robot playimg some music. It can take parameters : some music tone.

playSound

This command makes the robot playing some specific sound. It take a integer between 0 and 7 as parameter, each integer matches with a specific sound.

stopMusic

This command stops any music of the robot cat.

takePicture

This command takes a picture with the Lego camera.

doNothing

This command does nothing. It is used to set 'no reaction' on a specific event.

watchLeftMiddleRight

This command sets the watchLeftMiddleRight layer as active.

watchSymbol

This command sets the watchSymbo layerl as active.

watchTarget

This command sets the watchTarget layer as active.

stopWatch

This command disables any layer.

autonomous

This command triggers an autonomous behavior.

say

This command orders the cat to say the specified string. (It takes a parameter : a string between quotes).

wait

This command orders the cat to wait for the specified time. (It takes a parameter : a integer in seconds).

stopWait

This command orders the cat to stop waiting.

stopAll

This command orders the cat to stop all actions.

3.2 Test icons

The test icons represent the states of the robot cat. They will be used with the keywords "if", "while" and "doWhile" in order to compose conditional or repetitive behaviors. When these icons are used, they call upon the register base of the cat (its base of knowledge, its brain) to know the value of the test and thus to decide the continuation of the behavior.

The icons of tests are : isHungry, isSleeping, withCam, …
See chapter on the control icons to view an example.

Tests

true

This test is always true.

false

This test is always false.

isHungry

This test tells if the cat is hungry.

isHappy

This test tells if the cat is happy.

isTired

This test tells us if the cat is tired.

isSleeping

This test tells us if the cat is sleepy.

isSleepy

This test tells us if the cat is sleeping.

isSearching

This test tells us if the cat is searching something

withCam

This test tells us if the Lego camera is plugged.

withPC

This test tells us if the robot is communicating with the PC.

3.3 Event icons

Event icons stand for events for which the robot cat is sensitive, that is to say all the events it can detect and which are available for the user. These icons will be used with the keyword "when", thus you will be able to set up specific reaction on some event.

See chapter on the control icons to view a example.

Event

contactLeftPushed

Occurs when the left contact sensor of the robot is pushed.

contactLeftReleased

Occurs when the left contact sensor of the robot is released.

contactRightPushed

Occurs when the right contact sensor of the robot is pushed.

contactRightReleased

Occurs when the right contact sensor of the robot is released.

seeSmthOnLeft

If the LeftMiddleRight layer is active, occurs when an something is seen on the left.

seeSmthOnMiddle

If the LeftMiddleRight layer is active, occurs when an something is seen in the middle.

seeSmthOnRight

If the LeftMiddleRight layer is active, occurs when an something is seen on the right.

distanceTarget

If the Target layer is active, occurs when the distance to the target changes. Use the variable distanceToTarget in the knowledge to know what is the new distance.

symbolRead

If the ReadSymbol layer is active, occurs when a symbol is recognized. The ID of the symbol is updated in the Knowledge.

RCXMouseMessage

Occurs when the Cat receives an infrared RCX message from the Mouse.The message ID is stored in the Knowledge with the name RCXMessageID.

doNow

Occurs just after the user input a new CCL sentence, to execute the correponding new behavior.

3.4 Control icons

The control icons make it possible to structure the cat language from a logical way. It makes it possible to combine basic icons between them, but also to introduce the events and the tests in the language.

3.4.1 « repeat » icon

The "repeat" keyword makes it possible the cat to make several times the specified action in this sentence.
Its syntax :

```
repeat four times driveForward 20 cm, rotateLeft 90 end
```

3.4.2 « if » icon

the « if » keyword makes it possible to have conditionnal orders.
Its syntax :

```
if isHungry then startStroll end
```

3.4.3 « while » and « doWhile » icons

With the « while » keyword, you can order the robot cat to repeat the specified action as long as the test is not true. Its syntax:

```
while isHappy then rotateRight 10 end
```

3.4.4 « when » icon

The key word "when" makes it possible to specify reactions to him. A sentence beginning with the key word when does not start actions. It does nothing but set up reactions.
Its syntax:

```
when contactLeftPushed then driveBack 20 end
```

3.4.5 « doBoth » icon

The "doBoth" keyword makes it possible the cat to make two actions at the same time.
An example:

```
doBoth driveForward 120 and playSound 2 end
```

3.5 Complex commands

Because of the save function commands sequence, any sentence stated with the cat can be recorded and re-used thereafter. We will call it complex order, any sequence of orders saved by our application and then being able to be recalled with its identifier (its name).

- A complex order is thus a combination of basic orders, control commands and possibly others complex orders defined beforehand.

- A complex order is considered in grammar as a basic order, it can for example be used within control commands.
- A complex order is entirely defined by its ID (its name), which was given to him at the time of its save. Because of this name, it could be recalled in the language.

4. Conclusion

This application is not completely intended for all public. Even if the interface lets it think, it is really a appli of killer.