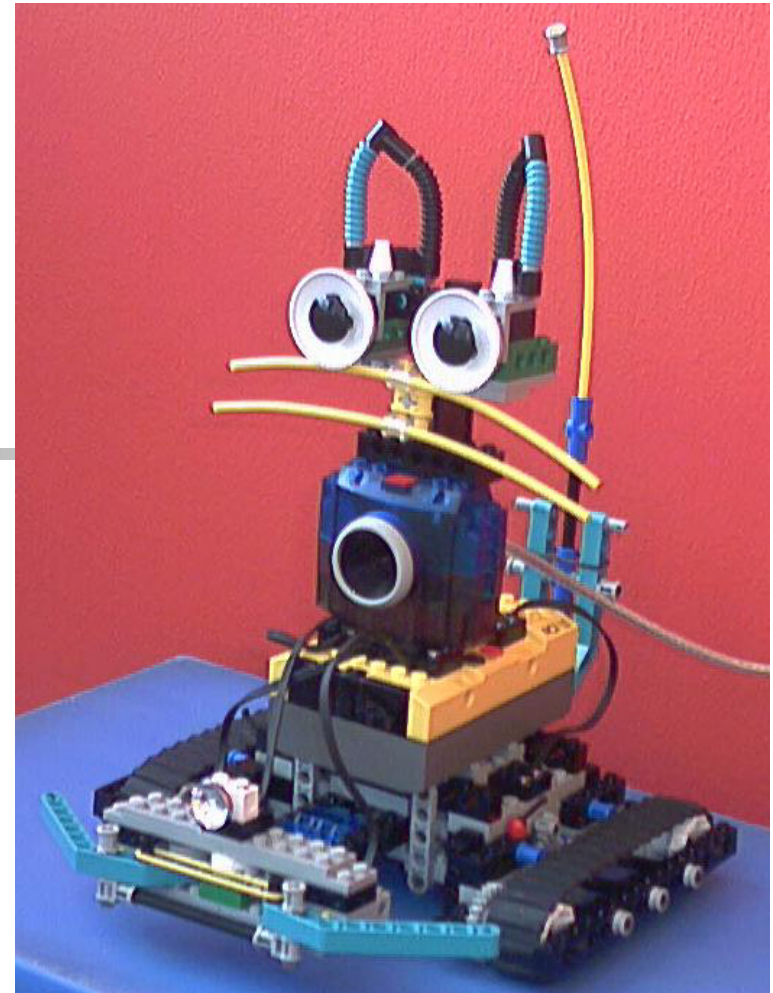


Multimodal  
Application for

Extensible  
Lego  
Intelligent  
Agent





# Problem Definition

---

- Lego MindStorms System
- Communication with user aspect
- Environnement communication aspect
- AI aspect



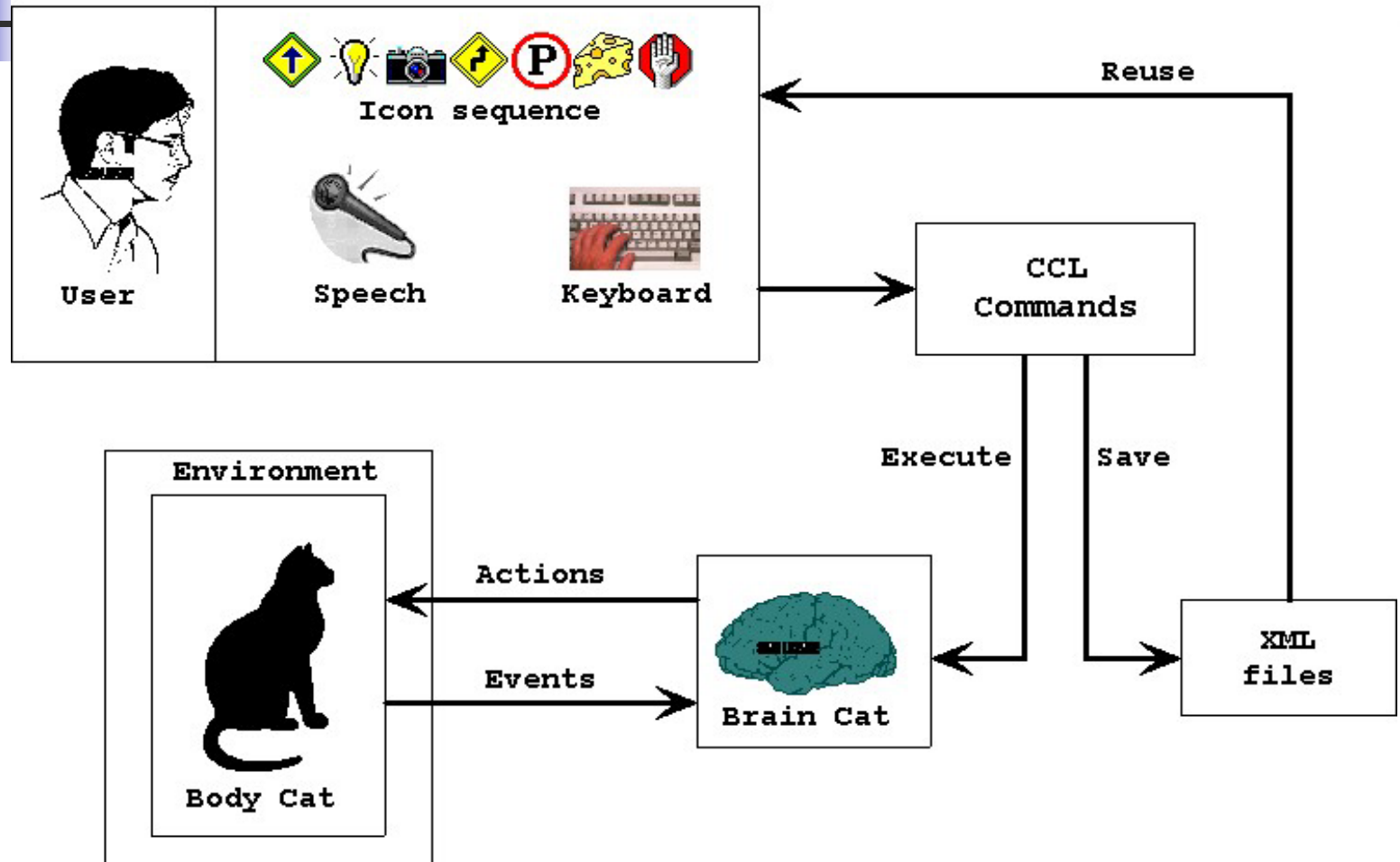
# Global presentation

---

Three aspects :

- AI Aspect
  - Entertaining Aspect
  - Components Aspects
- 
- Lego Environment

# MAELIA : Activity Diagram





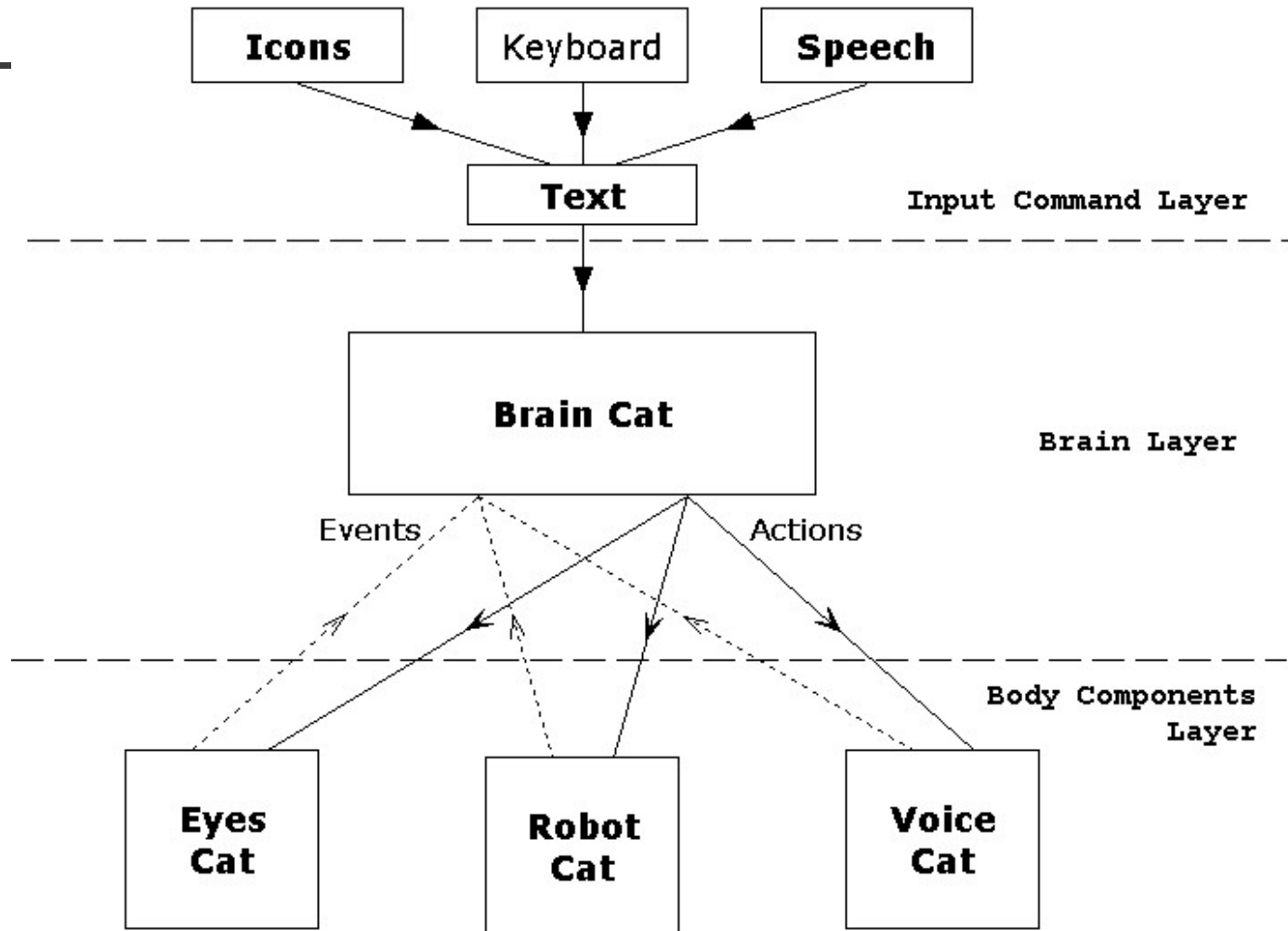
# Application architecture

---

## Diagram

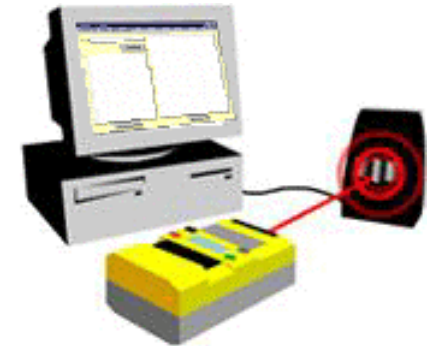
- Body components layer
- Brain layer
- Commands layer

# Architecture of MAELIA



# Body components

- RobotCat -> Lego RCX brick
- EyesCat -> LegoCam camera
- VoiceCat -> Speech generation





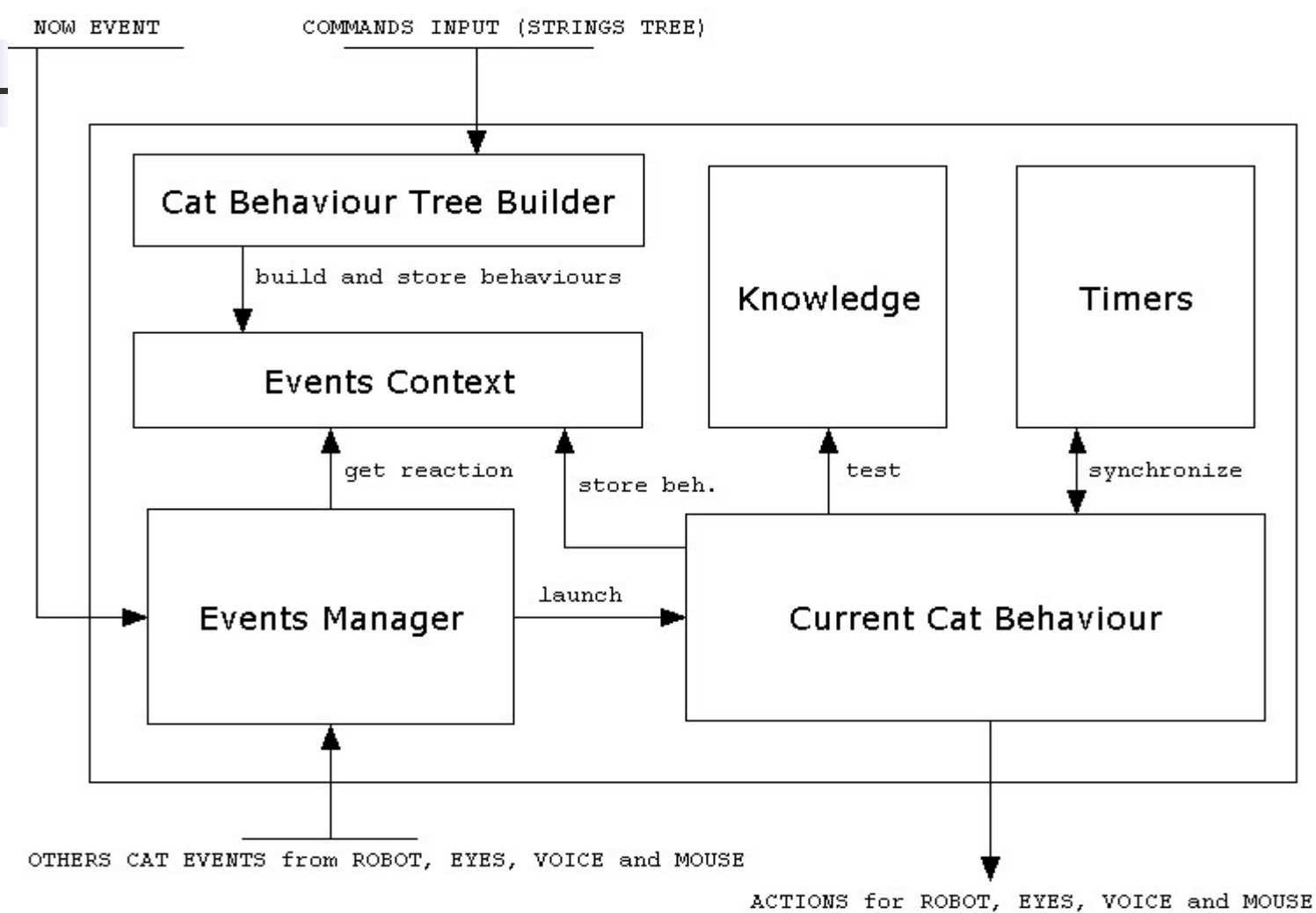
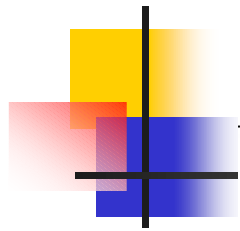
# Brain Cat

---

- Behaviors builder
- Knowledge
- Timers (notion of time)
- Events context (notion of reaction)
- Events manager (notion of event)



## Brain Cat : Collaboration Diagram





# Cat Command Language

---

- Idea from the 'event' pattern
- Structure & Grammar
- Vocabulary from one unique XML file

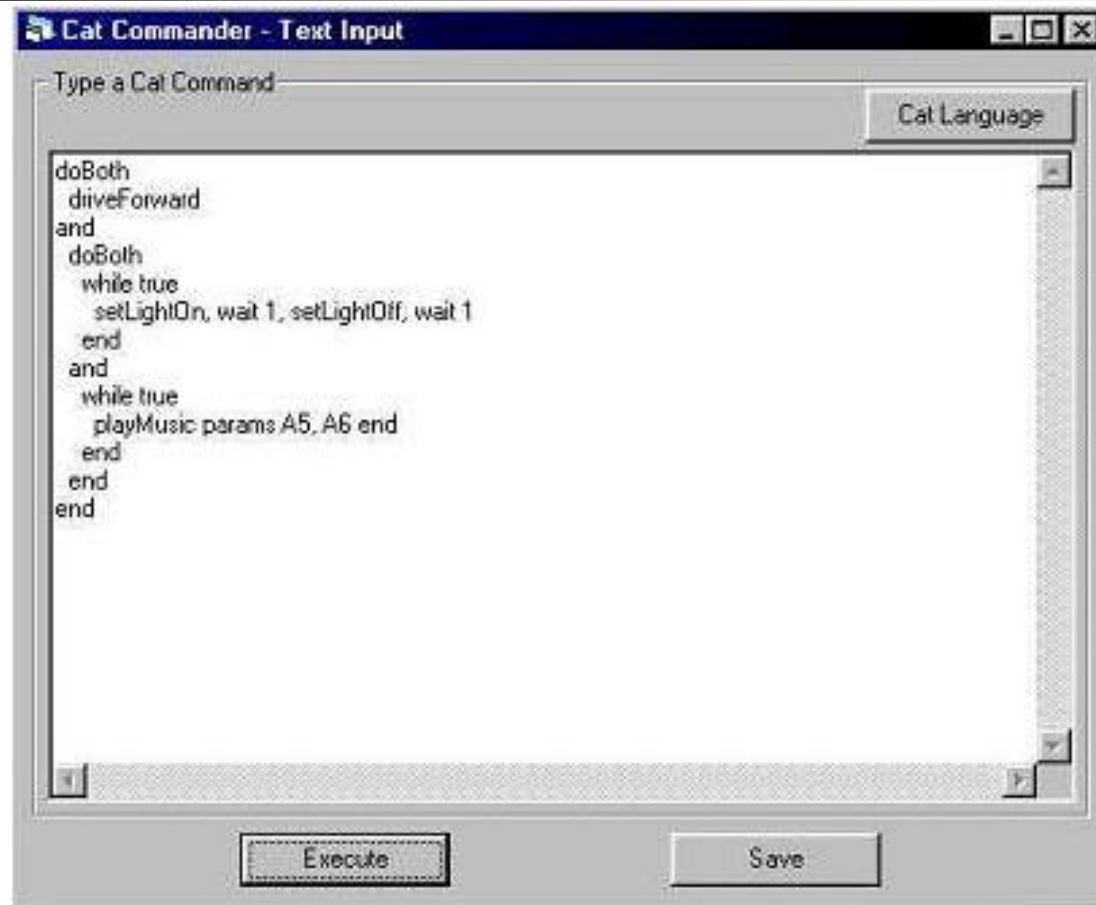


# User Interface

---

- Text Input (Cat Command Language)
- Icons Input
- Voice Input (Speech recognition)

# Text Input Interface





# MDI

The screenshot displays the 'Cat Robot Command System' application window. It features a menu bar (File, Edit, View, Windows, Layouts, Help) and a toolbar with various icons. The main interface is divided into several panes:

- Cat Commander - Text Input:** A text editor window containing a Cat Command script:

```
doBoth
driveForward
and
doBoth
  while true
    setLightOn, wait 1, setLightOff, wait 1
  end
and
  while true
    playMusic params A5, A6 end
  end
end
end
```

Buttons for 'Execute' and 'Save' are located below the text area. A 'Cat Language' button is also present.
- RCX Robot Cat:** A log window showing the execution progress of the command. It includes a 'Download' button and a 'Check RCX Robot' button. The log text is:

```
Processing InitComm and Checking hardware...
... Tower is connected to PC...
... Tower is Alive...
... PB is Alive...
... InfraRed Range set to Long...OK
Starting system task...
...OK
Power Down Time set to 5 min
Event 1 : Task 1 Finished
Bob is diving forward
Bob put light ON
Bob is playing tone A5
Bob is playing tone A6
Bob put light OFF
Bob is playing tone A5
Bob is playing tone A6
Bob put light ON
Bob is playing tone A5
Bob is playing tone A6
Bob put light OFF
Bob is playing tone A5
```
- Cat Actions State:** A panel showing the status of various actions:

RCX Move	Cat is Moving	Stop
RCX Music	Cat is Playing Music	Stop
PC Speech	Inactive	Stop
PC Wait #1	Cat is Waiting for Something #1	Stop
PC Wait #2	Inactive	Stop
PC Wait #3	Inactive	Stop
PC Think	Inactive	Stop
Stop All Actions		

At the bottom of the application window, a status bar displays: 'PBrick is Ready', 'Battery Level 8453 mV', and 'Power Down Time 5 min'. The Windows taskbar at the very bottom shows the Start button, several open applications (CatCommandGrammar, MDIDesign - Microso..., replace.jpg - HyperS..., Cat Robot Comm..., PoliceCar.txt - Notep...), and the system clock showing 8/16/02 at 3:10 PM.



# MDI

The screenshot displays the 'Cat Robot Command System' software interface. The main window is titled 'Cat Robot Command System' and contains several panes:

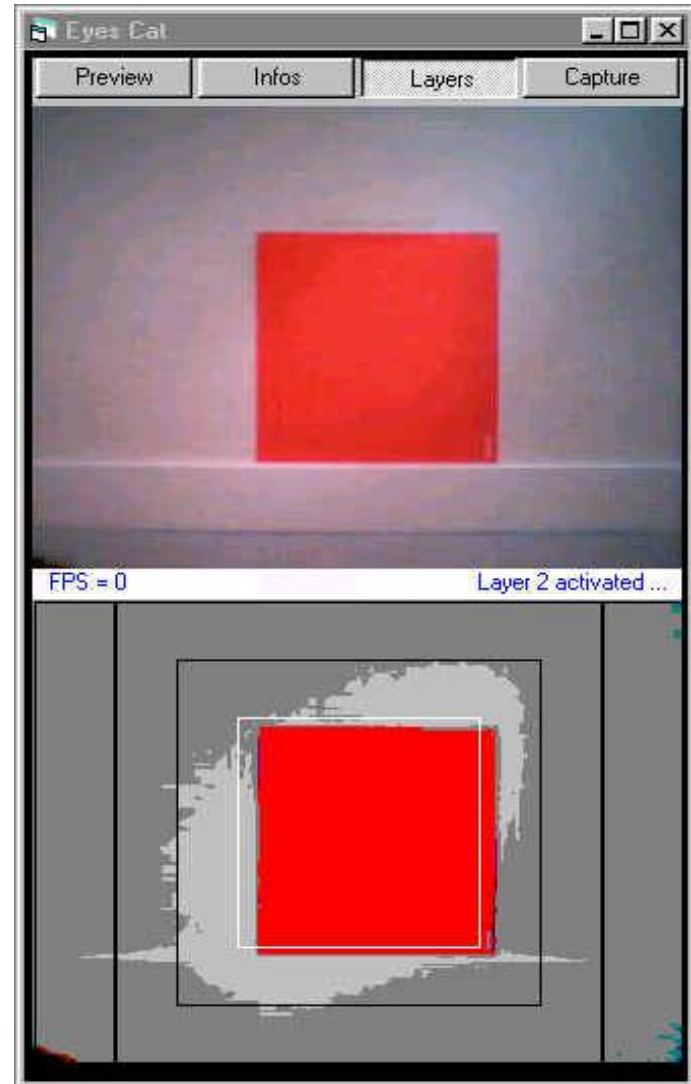
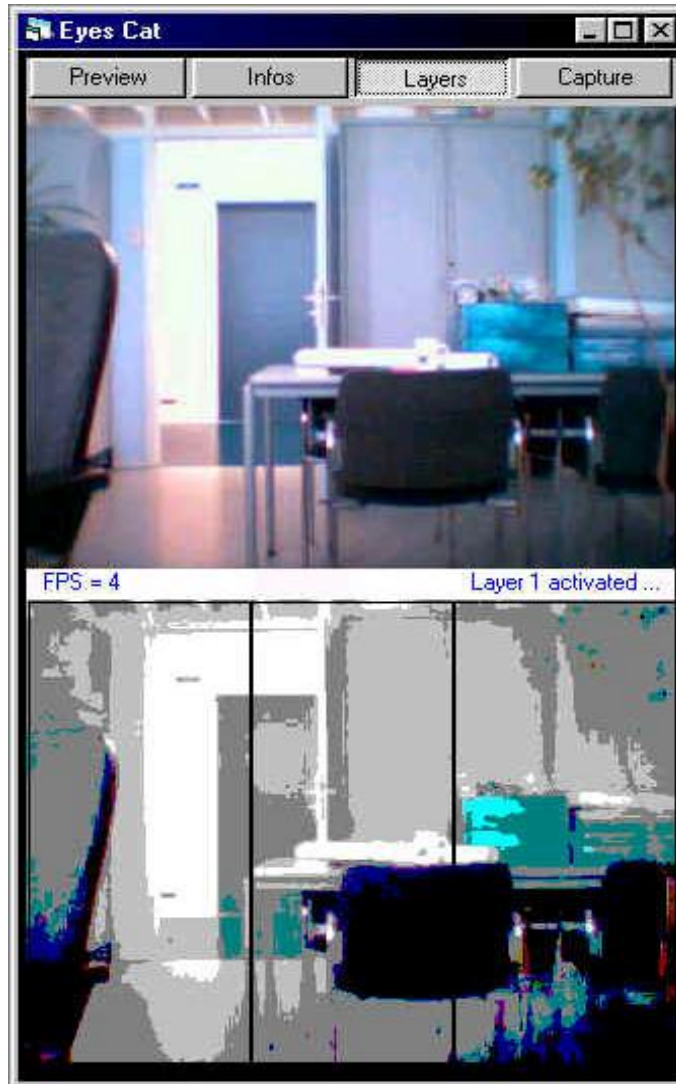
- Brain Cat:** A tree view showing 'Cat Events' with a sub-tree for 'doBoth' containing 'drive forward (infinite)' and 'wait for 20 sec'.
- Icon Commands Input:** A sequence of icons: 'doBoth', 'drive forward', 'and', 'wait', '20', and 'end'. Below this are 'Clear', 'Execute', and 'Save' buttons.
- Icon Library:** A grid of various icons including directional arrows, a stop sign, a lightbulb, a bomb, a play button, a bell, a stop sign with a musical note, a camera, a 'no entry' sign, an eye, a hand, a target, a cat, a pumpkin, a 'P' sign, a hand with a red stop sign, and various other symbols.
- Cat Knowledge:** A table listing various states and their values.
- Cat Actions State:** A list of actions with their current status and a 'Stop' button for each.

The status bar at the bottom right shows the date '8/16/02' and the time '3:06 PM'.

ID	Value
isHappy	False
isHungry	False
isSleeping	False
isSleepy	False
isTired	False
withCam	True
withPC	False
isSearching	False
readSymbol	0
RCXMouseMessageID	0
pictureFolder	..\CamPictures\
pictureName	CatPict

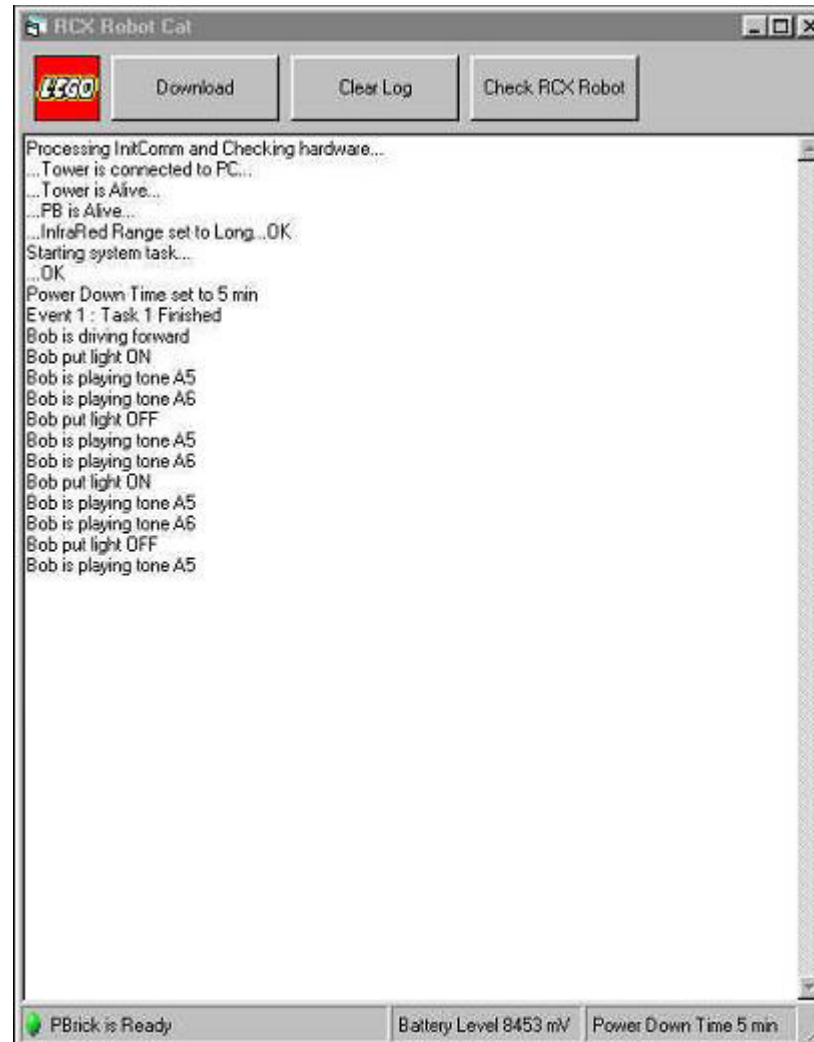
Action	Status	Stop
RCX Move	Cat is Moving	Stop
RCX Music	Inactive	Stop
PC Speech	Inactive	Stop
PC Wait #1	Cat is Waiting for Something #1	Stop
PC Wait #2	Inactive	Stop
PC Wait #3	Inactive	Stop
PC Think	Inactive	Stop
Stop All Actions		

# LegoCam layers

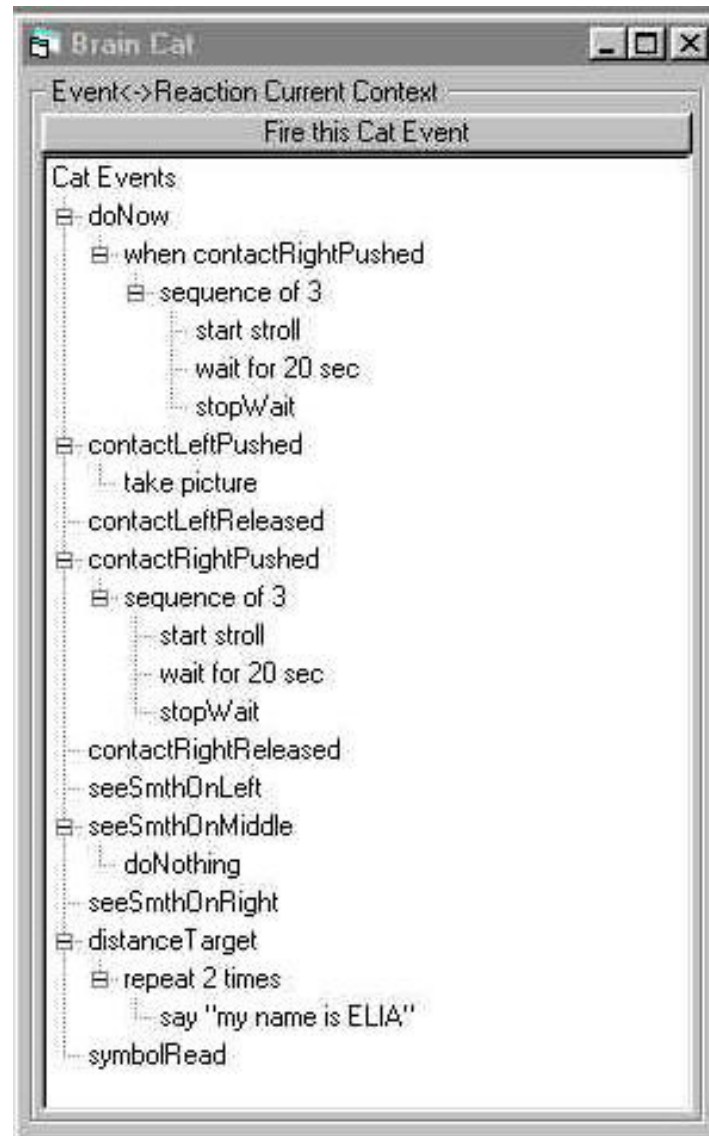




# RCX Robot Cat Interface



# Event context Interface



# frmTimers





# Recommendations

---

- Adapted Environment
  - - Light
  - - Space
  
- Use of the CCL
  - - sample with the pathFinder



# Perspectives & Overviews

---

- Re-use for new behaviors

It is made for that.

- New actions (and new components)
- New events (and new sensors)
- New tests



# Acknowledgements

---

- Dr. Drs. L.J.M Rothkrantz
- People from all the floor.
- Luca Porzio



# Sample : myNameIs

---

- say "Hello, my name is ELIA, I am about 2 months-old.",
- say "I am a robot cat able to communicate with you.",
- say "I am able to move.",
- setPowerMotor 7,
- driveForward 20, rotateLeft 360, say "youhouhou",
  
- say "I accept different parameters for to complete my driveForward command. It can be a distance in centimeters, in meters.",
- say "For example, I can draw a square.",
- repeat 4 times driveForward 20, rotateLeft 90 end,
  
- say "I can light my way",
- setLightOn, wait 3, setLightOff,
- say "I can move slowly",
- setPowerMotor 2, driveForward 20,
- say "or I can move very fast",
- setPowerMotor 7, driveForward 30,



# Sample : myNameIs (2)

---

- say "I am also able to play some music",
- playMusic params G5,G5,G5,A6,B6,A6,G5,B6,A6,A6,G5 end,
- say "I am also reactive, try to push on my contact sensors",
- when contactLeftPushed then say "houhouhou", driveBack 5 end,
- when contactRightPushed then say "hohoho", driveBack 5 end,
- wait 60,
- say "Finished !",
- when contactLeftPushed then doNothing end,
- when contactRightPushed then doNothing end,
- stopMoving,
- setPowerDownTime 6





# Sample : policeCar

---

- doBoth
- while true driveForward 10, turnRight end
- and
- doBoth
- while true
- setLightOn, wait 1, setLightOff, wait 1
- end
- and
- while true
- playMusic params A5, A6 end
- end
- end
- end



# Sample : use of events

---

- say "We are going to learn how to use events in this application",
- say "So you will be able to set some reaction on all events.",
- say "A first example, reaction on the contact sensors. Try to push on the contact sensors.",
- when contactLeftPushed then driveBack 10 end,
- when contactRightPushed then driveBack 10 end,
- wait 30,
- when contactLeftPushed then doNothing end,
- when contactRightPushed then doNothing end,
  
- say "A another good sample can be a small stroll around my environment",
- startStroll,
- wait 30,
- endStroll,



# Sample : use of events (2)

---

- say "We can see the difference between the stroll running on the RCX, the last one and the following stroll",
- when contactLeftPushed then driveBack 10, rotateLeft 20, driveForward end,
- when contactRightPushed then driveBack 10, rotateLeft 20, driveForward end,
- driveForward,
- wait 60,
- stopMoving,
- when contactLeftPushed then doNothing end,
- when contactRightPushed then doNothing end



# Sample : distanceTarget

---

- watchTarget,
- when distanceTarget then
  - when distanceTarget then
    - stopMoving,
    - say "Now I am close from the red target",
    - say "I stop",
    - stopWatch
  - end,
  - stopMoving, say "I am far from the red target",
  - wait 2,
  - driveForward
- end,
- driveForward



# Sample : findThePath

---

- setPowerMotor 2, rotateRightDuring 1, setPowerMotor 7,
- driveFollowingTheLine,
- when arrivedToACross then turnLeft, driveFollowingTheLine,
- when arrivedToACross then driveForwardDuring 1,  
driveFollowingTheLine,
- when arrivedToACross then driveForwardDuring 1,  
driveFollowingTheLine,
- when arrivedToACross then turnRight, driveFollowingTheLine,
- when arrivedToACross then driveForwardDuring 1,  
driveFollowingTheLine,
- when arrivedToACross then stopMoving, playSound 2, say "Yeah I got  
it", stopFollowingTheLine
- end end end end end end