Monitoring your Lego Mindstorms™ with Giotto

Sébastien Saudrais - Olivier Barais – Noël Plouzeau - Jean-Marc Jézéquel
Motivations

• to merge a formal model for timing specification with ‘classical’ components
• to build a process based on model driven engineering
• to cover time issues in classical applications (e.g. cooperative edition, shared spaces, i.e. applications where late results are not catastrophic but useless)
Mindstorms and LeJOS

Technical specifications
32-bit ARM7 microcontroller
256 Kbytes FLASH, 64 Kbytes RAM
8-bit AVR microcontroller
4 Kbytes FLASH, 512 Byte RAM
Bluetooth wireless communication
(Bluetooth Class II V2.0 compliant)

LeJOS is a Java based replacement firmware for the Lego Mindstorms
Overview of the process

Automaton of Component → Timed Automaton of Component

Time introduction [CBSE07]

Composition of components

Timed Automaton of the System

Implementation of the System

Implementation

Monitor in Giotto

Monitors

Model Transformation

Traditional Development

Specification

Implementation
Example

Controller
- IDmotor
- IDdistance
- void slow_down()

Distance
- void object_near()
- void object_far()

Motor
- Imotorspeed

Code:
- void object_near()
- void object_far()
- void slow_down()
Behaviour

- s1
  - ?object_near
  - ?object_far

- s2
  - !slow_down

- s3
  - ?object_near
Applying duration pattern

Generic duration pattern

Duration pattern
(?object_near, ?object_far, <, 5)
Overview of the process

Timed Automaton of the System Monitor
In Giotto
Composition of components
Implementation of the System

Timed Automaton of Component
Automaton of Component

Traditional Development

Time introduction
[CBSE07]

Model Transformation

Monitors
QoS behaviour

- Projection on timed transitions
- QoS violation transitions added
start s1 {
  mode s1() period 1000 {
    exitfreq 1 s3(drivers1s3near);
    taskfreq 1 do Inc();
  }
  mode s3() period 1000 {
    exitfreq 1 do s1(drivers3s1far);
    exitfreq 1 do s1(drivers3s1error);
    taskfreq 1 do Inc();
  }
}

?object_near
x1:=0
s1

?object_far
x1<5
s3

!violate
x1>=5
public class Conds1s3near extends BaseCondition implements ConditionInterface, Serializable {
    boolean object_near=false;
    public boolean run(Parameter parameter) {
        return object_near && X1.getValue()<5;
    }
    public void objectisnear(){
        object_near=true;
    }
}
Interaction components-monitor

- User
  - Informs the user
- Monitor in Giotto
  - Checks of messages arrival
- Intermediate files
  - Notifies of messages arrival
- Implementation in Lejos
  - Use of drivers
  - Call of functions like objectisnear()
State of implementation

• Process implemented using model transformation (Kermeta)
• Giotto adaptations for Lego Mindstorms (file support, hashtable, switch, class loader)
• Run on Tiny JVM: Lejos