LABORATORIO DI SISTEMI SOFTWARE

Introduction

Remember our motto:
there is no code without a project, no project without problem analysis and no problem without requirements.

Requirements

Design and build a ButtonLed software system (bls) that makes human users able to control a led according to different control policies each time a button is pressed:

1. es2-Blink/un-blink the led

The system should be a distributed system, in which the Button and the Led run on a different computational supports, e.g. a Conventional PC, a RaspberryPi or Arduino.

Requirement analysis

```
System sys_bls

Dispatch cmdOn       : cmdOn(ARG)
Dispatch cmdOff : cmdOff(ARG)
Event       button  : button( KIND )

Context ctxbutton ip [host="localhost" port=8010]
Context ctxled ip [host="127.0.0.1" port=8077]

QActor button context ctxbutton {

      State off initial {
            discardMsg On
            println("    Button pressed: Turned Off       ")
            forward led -m cmdOff : cmdOff(1)
      }
      Transition t0 whenEvent button -> on

      State on {
            println("    Button pressed: Turned On       ")
            forward led -m cmdOn : cmdOn(1)
      }
      Transition t0 whenEvent button -> off
}

QActor led context ctxled {

      State off initial {
            discardMsg On
            println("    Turned Off       ")
      }
      Transition t0     whenMsg cmdOn -> on

      State on {
            println("    Turned On       ")
            println("    =====Blinking=====   ")
      }
      Transition t0     whenMsg cmdOff -> off
}

/* For local test */
QActor user context ctxbutton {
      State s0 initial {
            println("user presses button")
            emit button : button( on )
            delay 2500
            println("user presses button")
            emit button : button( off )
            terminate 0
      }
}
```

Problem analysis

```
   System sys_bls

Request    cmdOn     : cmdOn(ARG)
Request    cmdOff     : cmdOff(ARG)
Reply              ack           : ack(ARG)
Event              button  : button( KIND )
Dispatch   blink : blink(ARG)

Context ctxlogicbutton ip [host="localhost" port=8010]
Context ctxlogicled ip [host="127.0.0.1" port=8077]

QActor button context ctxlogicbutton {

      State init initial {
            println( "BUTTON init done. Default initial led and button state: Off" )
      }
      Goto turningOff

      State turningOff {
```

```
		request led -m cmdOff : cmdOff(0)
	}
	Transition t0 whenReply ack -> off

	State off {
		printCurrentMessage
		println("    BUTTON turned OFF        ")
	}
	Transition t0 whenEvent button -> turningOn

	State turningOn {
		request led -m cmdOn : cmdOn(1)
	}
	Transition t0 whenReply ack -> on

	State on {
		printCurrentMessage
		println("    BUTTON turned ON  ")
	}
	Transition t0 whenEvent button -> turningOff

}

/*For local test*/
QActor user context ctxlogicbutton {
	State s0 initial {
		delay 2500
		println("USER presses button")
		emit button : button( on )
		delay 5500
		println("USER presses button")
		emit button : button( off )
		terminate 0
	}

}

QActor led context ctxlogicled {

	State init initial {
	discardMsg On
		println( "   LED init done." )
	}
	Transition t0     whenRequest cmdOn -> on
				whenRequest cmdOff -> off

	State off {
		onMsg(cmdOff : cmdOff(ARG)) {
			[#  val V = payloadArg(0)
				val Answer = "ackTo_${V}"
			#]
			replyTo cmdOff with ack : ack($Answer)
			println( "led has sent ACK with Answer: $Answer")
		}
		println( "led has been turned off")
	}
	Transition t0     whenRequest cmdOn -> on

	State on {
		onMsg(cmdOn : cmdOn(ARG)) {
			[#  val V = payloadArg(0)
				val Answer = "ackTo_${V}"
			#]
			replyTo cmdOn with ack : ack($Answer)
			println( "led has sent ACK with Answer: $Answer")
		}

		println( "led has been turned on" )
	}
	Goto blinkOn

	State blinkOn {
		println( "blink ON" )
		delay 500
		forward led -m blink : blink(off)
	}
	Transition t0     whenRequest cmdOff -> off
				whenMsg blink -> blinkOff

	State blinkOff {
		println( "blink OFF" )
		delay 500
		forward led -m blink : blink(on)
	}
	Transition t0     whenRequest cmdOff -> off
				whenMsg blink -> blinkOn
}
```

By Laura Mazzuca email: laura.mazzuca@studio.unibo.it