Rhapsody in C – an overview

Sudipta Chattopadhyay (TA)
School of Computing
National University of Singapore

Rhapsody so far (basic)
- Creating class and object
- Creating state-charts
- Accessing attributes of an object
- Calling an operation of a class
- Accessing parameters sent by an event
- Creating conditional states
- Accessing Rhapsody in built timer
- Create configuration and run model

Rhapsody (more)
- Create multiple class, association
- Create multiple objects of same class
- Communication among different objects
- Creating animated sequence diagram

Creating class and objects

Attributes

Operations

Creating attributes and operations inside a class

Creating statecharts

Objects

0
1
Trigger / action

Trigger = some event
Action = valid C code
Statecharts

X : int
Y : int
Message() : void

X : int
Y : int
Message() : void

Accessing attributes

0
1
Trigger / printf("%d\n", me->X);
Trigger / me->X++

Statecharts
Calling operation

Processor class

Message() : void

X : int
Y : int

Objects

0 1

Statecharts

Operation call = <class name>_<operation name>(me, <operation args>)

Ex: Processor_Message(me)

Calling operation

Processor class

Message() : void

X : int
Y : int

Objects

0 1

Statecharts

Implementation of operation

- Any valid C code
- Implementation resembles the body/definition of the function in C programming language

Passing parameters through event

Processor class

Message(x) : void

X : int
Y : int

Objects

0 1

Statecharts

Trigger(x) / printf("%d
", params->x);

Trigger has a parameter “x”.

All parameters are stored in a structure type variable “params”

To access “x” from statechart, access params->x (NOT JUST “x”)

Passing parameters through event

Processor class

Message(x, y) : void

X : int
Y : int

Objects

0 1

Statecharts

Trigger(x, y) / printf("%d %d
", params->x, params->y);

Can have multiple parameters

Access them as “params->x”, “params->y” and so on...
Conditional states

Condition provided in the "guard" field of transition

Condition: Any valid condition in C language
e.g. x == 0, x >= 0, x!= 0 and so on

Trigger has a parameter called "x"

Accessing Rhapsody timer

Rhapsody in built timer is called "tm"
It takes argument in milliseconds
In the above example, state 0 to state 1 transition will take place after 10 sec

Communication between objects

After 1 sec, processor sends read() signal to the "itsMemory" object, which is of type Memory.
itsMemory must change its state from 0 to 1 at that point.
Communication between objects

Processor

statechart01

Trigger / me->X++

Memory

statechart01

Note that read() is not generated by the external environment

Dealing multiple objects of same class

Processor

X : int
Y : int

Message() : void

Memory

data : int

Message() : void

Change the multiplicity field of “object” and “association”

Dealing multiple objects of same class

X : int
Y : int

Message() : void

Processor

Object name : itsProcessor

Memory

Object name : itsMemory[0]
itsMemory[1]
itsMemory[2]

Dealing with multiple objects of same class

Processor

statechart01

Trigger / me->X++;

Memory

statechart01

Trigger / me->X++;

read

Now read() will be received only by “itsMemory[1]” object.
Therefore, the instance statechart of “itsMemory[1]” will change state
Instance statecharts corresponding to “itsMemory[0]” and
“itsMemory[2]” will have no effect

Broadcasting

Processor

statechart01

Trigger / me->X++;

Memory

statechart01

Trigger / me->X++;

A sample trick

Want to talk?

Yes

No

Hi
A sample trick

Sending the object as an argument to “ack” back to the “Processor”

“x” is the parameter of “ack” of type “void”

Creating animated sequence diagrams

- Packages -> Default -> add new -> sequence diagram
- Drag the instances not the classes
  - Why?
    - A class may have multiple objects at runtime
- Keep the setting in “Design” mode

THANK YOU