

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

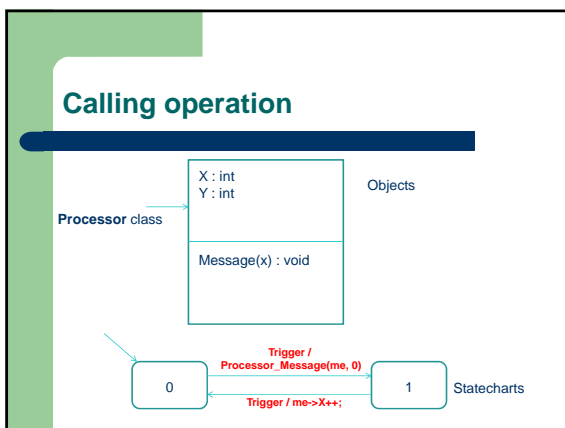
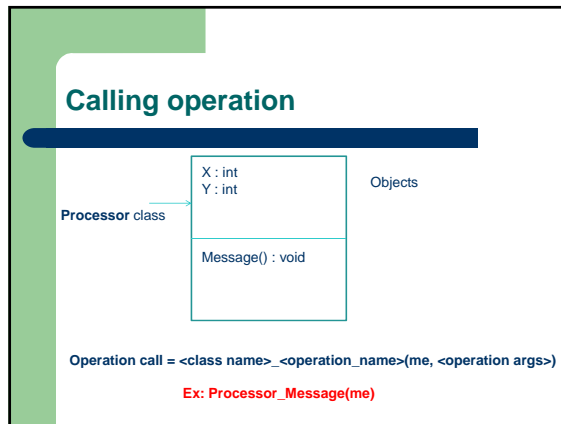
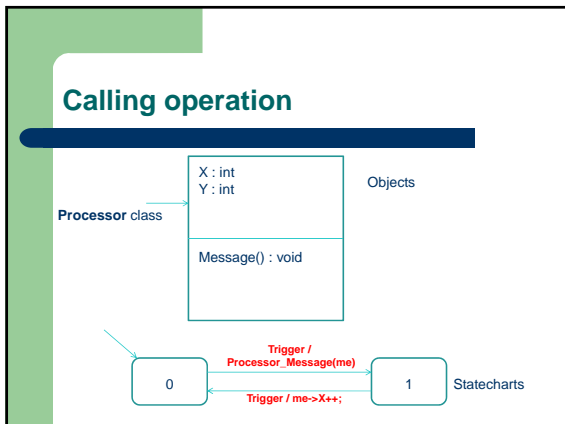
Creating class and objects

Creating attributes and operations inside a class

Creating statecharts

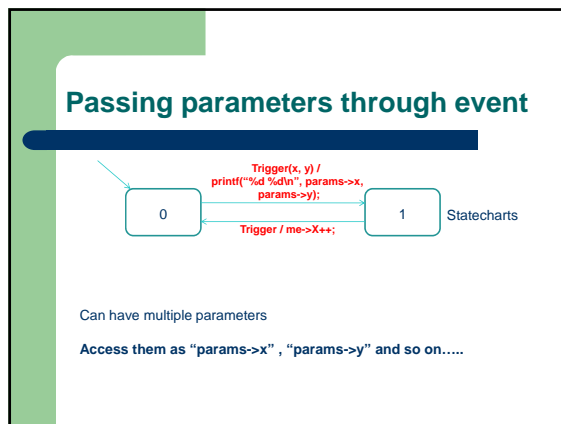
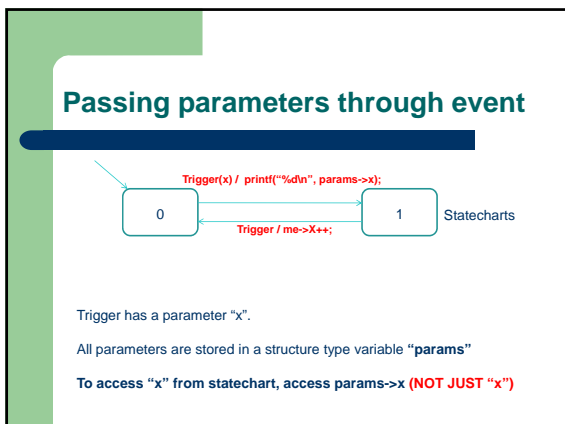
Trigger = some event
 Action = valid C code

Accessing attributes



Implementation of operation

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



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

Conditional states

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

Communication between objects

Say object name : itsProcessor Say object name : itsMemory
 Use `RICGEN(&itsMemory, read())` if you want to send `read()` event to Memory object

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

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

Dealing multiple objects of same class

Change the multiplicity field of "object" and "association"

Dealing multiple objects of same class

Object name : itsProcessor

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

Dealing with multiple objects of same class

Now read() will be received only by "itsMemory[1]" object.

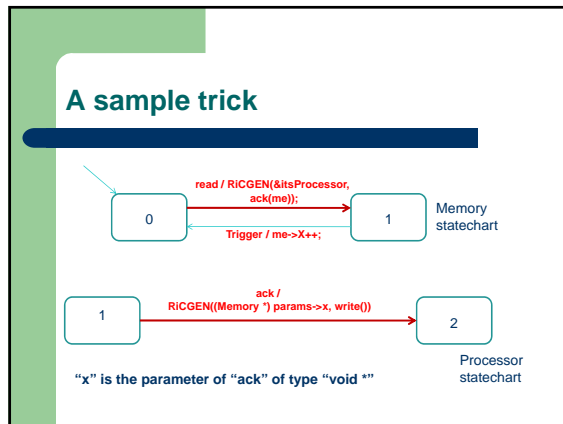
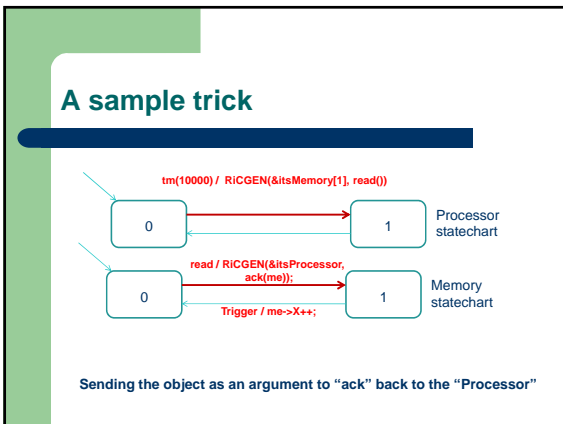
Therefore, the instance statechart of "itsMemory[1]" will change state

Instance statecharts corresponding to "itsMemory[0]" and "itsMemory[2]" will have no effect

Broadcasting

All of itsMemory[0], itsMemory[1] and itsMemory[2] will change state from state 0 to state 1

A sample trick



- ### 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