CASE – MODULES AND DIAGRAMS

package myentitites

import case.lang.System

namespace EntitiesNameSpace {

@module

String->Object->Main

#public class Program

[public Program(String [] args)

[EntityPool Pool = EntityPool.getEntityPool]

assert(Pool) //asserts that Pool exists and has a value

Stream (n) String

Int MyInt = EntityPool.getStreamMemory() //retrieve mem from pool

Int GetInt = EntityPool.get(“MyInt”) //pointer to MyInt using pool get

//get pointer to CurrentLocationInList from the pool

Int ListStatus = n.get(“CurrentLocationInList”)

]

[ printTodaysDate(Date dateObject)

Write dateObject.today()

]

#end class

//module connection

//Print “Hello World”

//Becomes

//<keyword command=”Print”>Hello Word</keyword>

@module MyApplication

<class command=”printTodaysDate”></>

Where the keyword portion is (in this case) indicates we are accessing a class

Command indicates there will be a function that we will be executed should this module be Pinged. The value in between the brackets is the value that will be passed to that function (command).

Another function of modules is something called Diagrams. Diagrams

are embedded XML meant to represent classes, data types, etc.

For example:

[ myFunction()

module {

<variable=”storeResults” type=”String”/>

<class=”Program” command=”printTodaysDate” return=”storeResults”/>

<keyword=”return” value=”storeResults”>

}

Write storeResults

]