

Program the Emulator

Yang Su

Laboratory for Software Technology
ETH Zurich

Programmatic Interface

- Emulator control software mainly written in Java
- Users can write java code to control the emulator (and experiments)
- User classes have full accesses to internal emulator data structures
- Complex behavior possible

Steps to write user code

- Write java program which extends `SimpleUserclass`
- Compile the java program into bytecode program (`javac`)
- Specify the user class in the experiment script
- Run the experiment script

SimpleUserClass

```
public abstract class SimpleUserClass
extends java.lang.Object
implements EmulatorUserClass
```

- Field
 - **dComm**: communication with `nodeDaemon`
 - **emulator**: main class in charge of running the emulator
 - **emuNodes**: physical device in the emulator
 - **numNodes**: # of physical devices
 - **sigEnv**: signal propagation environment
 - **world**: virtual world
- Method
 - **start ()**: start the emulation

Hello world

```
import edu.cmu.emulator.*;

/*
 * HelloWorld.
 */
public class HelloWorld extends SimpleUserClass {
    public void start(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Specify user classes in the script

```
<UserClassDef>  
  <class>  
    <name>HelloWorld</name>  
    <args></args>  
    <url>http://localhost/</url>  
  </class>  
</UserClassDef>
```

Example

- Remote execution
- Manipulate channels

Remote execution

```
public class RemoteExec extends SimpleUserClass {
    public void start(String[] args) {
        try {
            String nodeName;
            if (args.length != 1) {
                System.out.println("args: <nodeName>");
                return;
            }
            nodeName = args[0];
            result = dComm.execProcessAndWait_CentralLogIO(nodeName,
                "hostname",
                "hostname."+ nodeName + ".stdout",
                "hostname."+ nodeName + ".stderr");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```