Myx Demo

Hazel Asuncion
hasuncio@ics.uci.edu

For Inf 221

February 26, 2009

Demo from Scott Hendrickson's Hello World 2 Tutorial

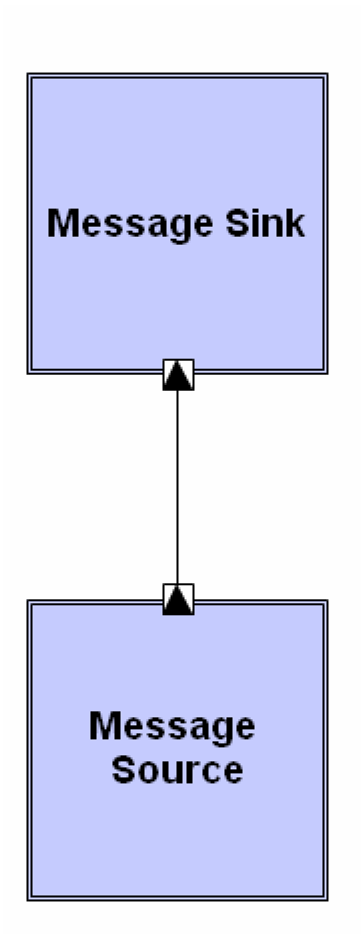
http://tps.ics.uci.edu/svn/projects/archstudio4/documents/HelloWorld_TwoComp.pdf

Developing a Simple System

- Design, Myx Style
- Implement Myx Components
- Assign Myx Components to Implementations

Design a Simple System

- Message Source sends a message to Message Sink
- Message Sink outputs the message to the console



Designing…in a Nutshell

(Reordered, to use shortcuts in Type Wrangler)

1. Create a new component
2. Create new component types (usually one for each new component added)
3. Add signatures to component types
4. Create interface types
5. Type Wrangler
    1. Assign components to component types
    2. Assign interfaces and signatures to interface types (Type Wranger)
6. Add links

Implementation

- Add myx.fw to the build path
- Service object: MessageType

```
package helloworld2;

public interface MessageType {

public void processMessage(String message);
}
```

- Implement Myx components
  - MessageSinkComponentType
  - MessageSourceComponentType

Implementation

```
package helloworld2;

import edu.uci.isr.myx.fw.AbstractMyxSimpleBrick;
import edu.uci.isr.myx.fw.IMyxName;
import edu.uci.isr.myx.fw.MyxUtils;

public class MessageSinkComponentType extends
   AbstractMyxSimpleBrick implements MessageType {
    public static final IMyxName MESSAGES_NAME =
       MyxUtils.createName("messages");

    public Object getServiceObject(IMyxName interfaceName) {
        if (interfaceName.equals(MESSAGES_NAME)) {
            return this;
        }

        return null;
        }
    public void processMessage(String message) {
    System.out.println("Received: " + message);
    }
}
```

Implementation

```
package helloworld2;

import edu.uci.isr.myx.fw.AbstractMyxSimpleBrick;
import edu.uci.isr.myx.fw.IMyxName;
import edu.uci.isr.myx.fw.MyxUtils;

public class MessageSourceComponentType extends
   AbstractMyxSimpleBrick {

    public static final IMyxName MESSAGES_NAME =
       MyxUtils.createName("messages");

    public void begin() {
         super.begin();

         MessageType messages =
            (MessageType)MyxUtils.getFirstRequiredServiceObject(this, MESSAGES_NAME);

         messages.processMessage("Hello world");
    }
    public Object getServiceObject(IMyxName interfaceName) {
         return null;
         }
}
```

Assign Myx Components to Implementations - ArchEdit

- ComponentType
  - VariantComponentTypeImpl
  - AddImplementation
    - JavaImplementation
      - Main Class – JavaClassName:
        - » helloworld2.MessageSourceComponentType
        - » helloworld2.MessageSinkComponentType
  - Signature
    - SignatureImpl
    - AddImplementation
      - LookupImplementation
        - » Name: messages
    - ServiceType
      - MessageSourceComponentType: requires
      - MessageSinkComponentType: provides

Package & Class name only (no ext)

Assign Myx Components to Implementations - ArchEdit

- InterfaceType
  - InterfaceTypeImpl
  - AddImplementation
    - JavaImplementation
      - Main Class
        - JavaClassname: helloworld2.MessageType

Instantiate your architecture

- Drag the HelloWorld2.xml to AIM Launcher
- Select Hello World 2 from the outline view
- Click on Instantiate

Troubleshooting

- `Invalid operation: Could not find method getAllImplementations on class edu.uci.isr.xarch.types.SignatureImpl`
  - Check that all signature nodes have lookup implementations
  - Make sure the lookup implementations match the name of the interface in your Myx Component
  - Make sure that InterfaceTypes have Java implementation

- `MyxBrick could not be instantiated (or found)…`
  - Check that the Java class names are correctly spelled
- `Unable to see the link from internal component to outer component`
  - Display the ComponentType that contains the subarchitecture in Archipelago
  - Right-click on the signature and select “New Signature-Interface Mapping”
  - Drag it to the interface of the internal component

Environment setup

- Follow the instructions on setting up your environment
  - http://www.ics.uci.edu/~hasuncio/classes/in4matx119/HelloWorldTutorial.pdf

Other examples

- Chatsys – two component with one Event Pump connector:
  - http://tps.ics.uci.edu/svn/projects/archstudio4-examples/chatsys/trunk/
- Lunar Lander
  - http://tps.ics.uci.edu/svn/projects/archstudio4-examples/lunar-lander/trunk/edu.uci.isr.lunarlander/

Other examples

- Design
  - ArchStudio
  - http://tps.ics.uci.edu/svn/projects/archstudio4/trunk/edu.uci.isr.archstudio4/src/edu/uci/isr/archstudio4/archstudio4.xml