

Interfacing WME Pages to Java Applet Manipulatives

Paul S. Wang 王士弘

Institute for Computational Mathematics

Kent State University

`pwang@cs.kent.edu`

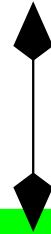
Motivation

- WME uses manipulatives in lessons to support interactive and inquiry-based learning.
- All current WME manipulatives are Javascript based.
- WME is establishing a uniform page-to-manipulative API known as the *WME manipulative architecture*.
- The National Library of Virtual Manipulatives (NLVM) offers a large collection of mathematical manipulatives in the form of Java Applets.
- We want to look into ways to make NLVM conform to the WME API so they become interoperable within WME.

WME Lesson Page and Applet Interactions

WME lesson

HTML + Javascript Code

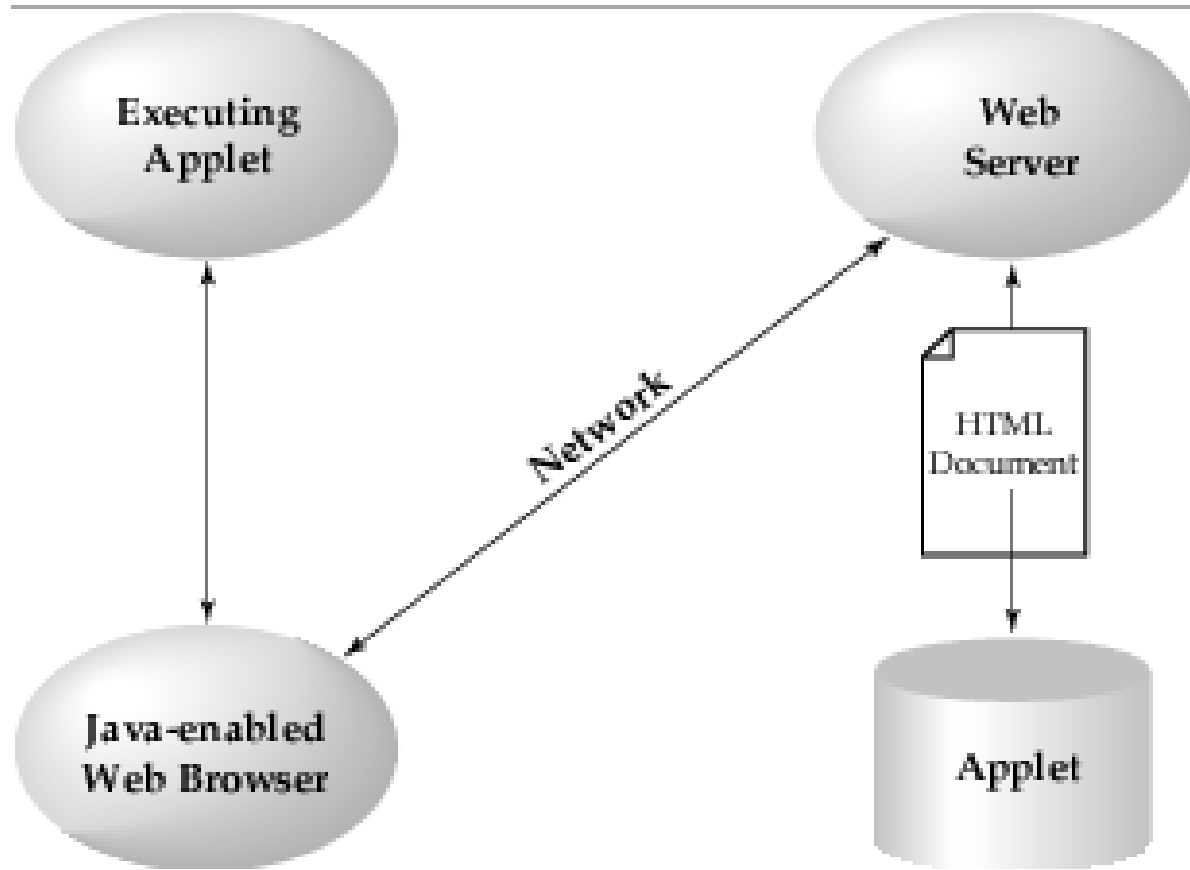


Applet Manipulative

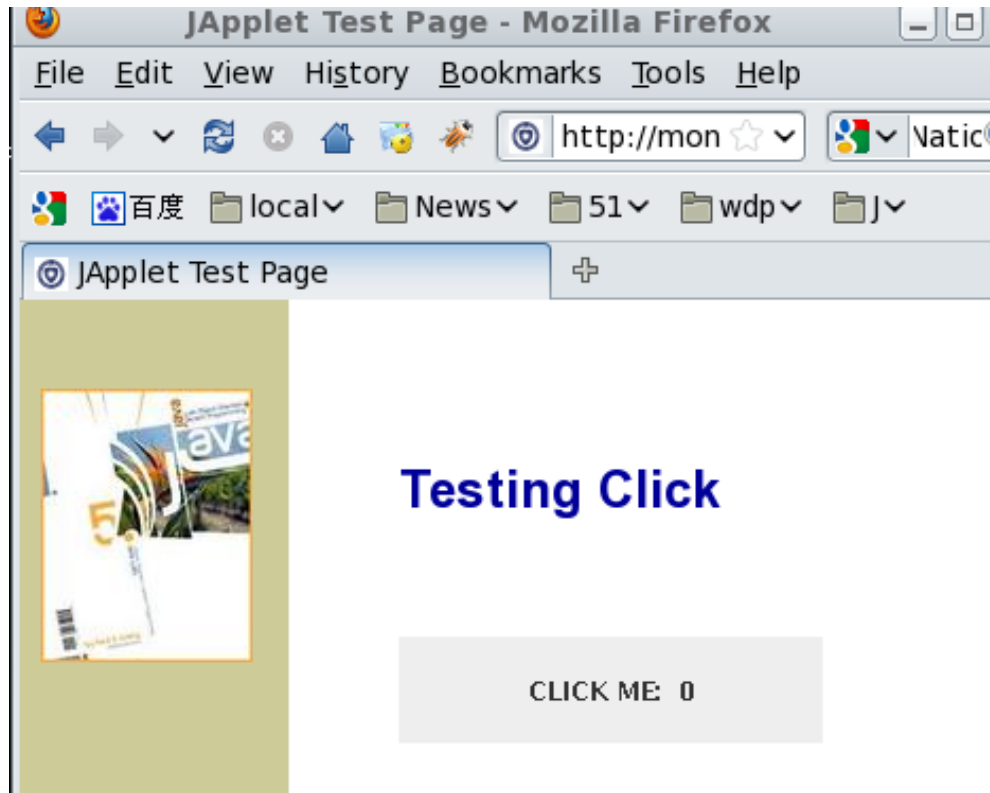
What is A Java Applet

- An applet is written in Java as an *extended class* of a Java-supplied class (usually `javax.swing.JApplet`).
- The compiled applet is deployed in a Web page via the HTML `<applet>` or `<object>` tag. The former being the preferred option.
- To run a Java applet, a Web browser requires the free Java Plugin.

Applet in Web Page



A Quick Applet Example



Counting Clicks

The Java Source Code

```
//////// Click.java //////////  
import java.awt.*;  
import java.awt.event.*;  
import javax.swing.*;  
  
public class Click extends JApplet  
{  
    public void init()  
    {  
        lb = new JLabel(c1 + n, JLabel.CENTER);  
        lb.addMouseListener(new ClickHandler(this));  
        getContentPane().add(lb, BorderLayout.CENTER);  
    }  
}
```

```
public void doClick()
{   n++;   lb.setText(c1+n);   }

protected String c1 = "CLICK ME: ";
protected JLabel lb;
protected int n = 0;
}
```



```
class ClickHandler extends MouseAdapter
{
    ClickHandler(Click ap)
    {
        app = ap;
    }

    public void mouseClicked(MouseEvent e)
    {
        app.doClick();
    }

    private Click app;
}
```

Javascript Access to Applet Methods

It is as simple as one-two:

1. Obtain the Applet Object (`jobj`) under DOM (Document Object Model).
2. Call any public method of the Applet through the `jobj`:
`jobj.methodName(...)`.

Javascript-to-Applet Demo



The Demo (works on IE, Firefox, and Safari)

The Applet Methods Called

```
// Method in class Click
public void doClick()
{   n++;   lb.setText(cl+n);   }

// Methods in extended class Clickjs0
public void doClickarg(int m)
{   n+=m;   lb.setText(cl+n);   }

public void doClicktext(String t)
{   lb.setText(t);   }
```

Calling Javascript from Applet

It is also possible to invoke Javascript functions from within the Java applet.

1. You need the Java library `netscape.jar` available at our WME site.
2. This library allows you to obtain the Javascript window object of its host Web page and use it to make Javascript calls.
3. Add `mayscript="mayscript"` to the `<applet>` or `<object>` tag.
4. This ability makes it easy for an Applet to update values in the Web page.

Applet to Javascript Interface Demo



The Demo (works on IE, Firefox, and Safari)

The Java Code

```
//////// Clickjs.java //////////  
import java.awt.*;  
import java.awt.event.*;  
import javax.swing.*;  
import netscape.javascript.*;  
  
public class Clickjs extends Click  
{ public void init()  
  { super.init();  
    win = JSObject.getWindow(this);  
  }  
}
```

```
public void doClick()
{
    super.doClick();

    // show_value is JS function
    String js_exp="show_value(" + n + ")";

    win.eval(js_exp);
}

protected JSObject win;
}
```


Java Code Location

On `monkey.cs.kent.edu` at

`/var/www/html/WME/javaJS`

`Click.java`

`Clickjs0.java`

`Clickjs.java`

Page-to-Applet Parameters

```
<applet id="cap" width="200" height="50"  
    code="tic.class" mayscript>  
    <param name="soundFile" value="beep.au">  
    <param name="oImageFile" value="o.gif">  
    <param name="xImageFile" value="x.gif">  
    <param name="name" value="tictactoe">  
</applet>
```

Applying These Techniques to NLVM

- Use Java decompiler to obtain VM source code. We are after its public and protected methods.
- Use Java class extension to add necessary methods to any target VM. Let's call these derived VM (dVM).
- Use applet parameters for manipulative customization.
- Write a *Javascript Wrapper* to envelop any dVM and deploy in a host WME page. This JS-Applet combination shall comply to WME manipulative architecture as much as possible.