

Notes from the NetKernel4 Frontier: Porting PoiNK to NK4

Tom Hicks

Tohono Consulting LLC

hickst@tohono.com

◇ About the Speaker

- Masters degrees in Computer Science..
 - Software Engineering & Language implementation
- ..and Cognitive Science
 - Computational Linguistics
- 29 years of software development experience
 - Wide variety of applications (info dissemination)
 - Using Java since 1997 (more Groovy now)
- Keeping an eye on NetKernel for several years
 - Developed several NetKernel3 modules
- Available for systems consulting, design, development, and training

◇ Outline

- Introduction to PoiNK
- Resource Models
- The PoiNK Resource Model
- PoiNK: NK3 to NK4
- New Documentation System
- New Unit Test Framework
- Additional Resources

◇ Introduction

- What is PoiNK?
 - A NetKernel module to **read** Excel spreadsheets
 - Provides access to spreadsheet "parts"
 - An example of a NK Resource Model
- Built on the Apache POI project
 - Open source project to..
 - "Develop pure Java ports of file formats based on Microsoft's OLE 2 Compound Document Format"
 - PoiNK using only the HSSF (Excel format) library
 - <http://poi.apache.org/>
- PoiNK uses the Apache License 2.0.

◇ Outline

- Introduction to PoiNK
 - Quick Demo - (circumstances permitting)
- Resource Models
- The PoiNK Resource Model
- PoiNK: NK3 to NK4
- New Documentation System
- New Unit Test Framework
- Additional Resources

◇ Examples

■ <http://localhost:1060/rowXML.groovy>

```
- <row row="18" sheet="1">
- <cell col="A" formatCode="0" row="18" sheet="1" type="S">
  <value>2.3.3.a.2.1</value>
</cell>
- <cell col="B" formatCode="0" row="18" sheet="1" type="S">
  <value>Identify constraints</value>
</cell>
- <cell col="C" formatCode="a4" row="18" sheet="1" type="F">
  <formula>G18/8/I18</formula>
  <value>0.3333333333333333</value>
</cell>
- <cell col="D" formatCode="a7" row="18" sheet="1" type="D">
  <value>04/10/06</value>
</cell>
- <cell col="F" formatCode="0" row="18" sheet="1" type="S">
  <value>sp,th,cs</value>
</cell>
- <cell col="G" formatCode="a4" row="18" sheet="1" type="N">
  <value>8.0</value>
</cell>
- <cell col="I" formatCode="0" row="18" sheet="1" type="N">
  <value>3.0</value>
</cell>
</row>
```

◇ Examples

■ <http://localhost:1060/colXML.idoc>

```

- <column col="B" sheet="2">
- <cell col="B" formatCode="0" row="1" sheet="2" type="S">
  <value>Test Cells: DO NOT CHANGE:</value>
</cell>
- <cell col="B" formatCode="0" row="2" sheet="2" type="B">
  <value>>true</value>
</cell>
- <cell col="B" formatCode="0" row="3" sheet="2" type="b">
  <value/>
</cell>
- <cell col="B" formatCode="0" row="4" sheet="2" type="F">
  <formula>1/0</formula>
  - <value>
    <error>Spreadsheet formula evaluation error 7</error>
  </value>
</cell>
- <cell col="B" formatCode="0" row="5" sheet="2" type="F">
  <formula>1+1</formula>
  <value>2.0</value>
</cell>
- <cell col="B" formatCode="0" row="6" sheet="2" type="N">
  <value>92.0</value>
</cell>
- <cell col="B" formatCode="e" row="7" sheet="2" type="D">
  <value>11/14/07</value>
</cell>
- <cell col="B" formatCode="0" row="8" sheet="2" type="S">
  <value>String</value>
</cell>
</column>

```

◇ Examples

- `http://localhost:1060/cellXML.bsh`

```
- <cell col="B" formatCode="0" row="9" sheet="1" type="S">  
  <value>Web Technology</value>  
</cell>
```

- `http://localhost:1060/cellVS.bsh`

```
System Engineering
```

- ...and also XML for Sheets and entire Workbooks
- Note: these scripts are calling PoiNK Accessors:
 - `active:cellXML`
`+wb@res:/data/tasks.xls+address@wbAddr:1!B9`

◇ Outline

- Introduction to PoiNK
- **Resource Models**
- The PoiNK Resource Model
- PoiNK: NK3 to NK4
- New Documentation System
- New Unit Test Framework
- Additional Resources

◇ Resource Models

- Central to Resource Oriented Computing paradigm
- ROC analog to the Object Models of OO paradigm
 - models relevant properties of an application domain
 - defines Information Types
 - set of values having a particular meaning or purpose
 - defines Services
 - which operate on the information types
 - independent of various physical implementations
- Netkernel has many built-in resource models
 - XML, JSON, PiNKY, Image, etc.

◇ Resource Models - Abstract

- A Resource Model is comprised of:
 - Abstract Information Types
 - which characterize resource instances in the model
 - Model-specific Addressing Scheme
 - (optional) a way to identify resources in the model
 - Often have a domain-specific addressing grammar
 - Services
 - Operate on resources of the model's types

◇ Resource Models - Concrete

- Realization of Resource Models:
 - Abstract Information Type
 - Implemented by a Representation
 - Model-specific Addressing Scheme
 - Implemented as a "mini" resource model
 - Services
 - Operate on physical representations of resources
 - Operate on each other (composition)
 - Transreptors (transrepresentors)
 - Transform between physical representations
 - **Do not** change the information, only its form
 - Ideally, lossless and bi-directional

◇ Outline

- Introduction to PoiNK
- Resource Models
- **The PoiNK Resource Model**
- PoiNK: NK3 to NK4
- New Documentation System
- New Unit Test Framework
- Additional Resources

◇ The PoiNK Resource Model - (1)

- Abstract Information Types
 - Workbook
 - Sheets
 - Rows
 - Columns
 - Cells
 - Workbook Address
- Addressing Scheme
 - Workbook Address
 - really a "mini" resource model
 - defines a syntax for identifying parts of a Workbook
 - examples: 1!B9, H4, 2!AC24, 3!A

◇ The PoiNK Resource Model - (2)

- Services
 - get a Cell's value as a String
 - get a Cell's value as XML
 - get Column values as XML
 - get Row values as XML
 - get Sheet values as XML
 - get Workbook values as XML
- PoiNK is a limited Resource Model
 - **Could** define many other services within the model
 - services which manipulate the entities of the model
 - services which change values within the model

◇ The PoiNK Resource Model - (3)

- Transreptors

- Excel file --> Workbook*
- WorkbookAddress --> String
- String --> Workbook Address
- WorkbookAddress --> XML
- XML --> WorkbookAddress
- Workbook --> XML

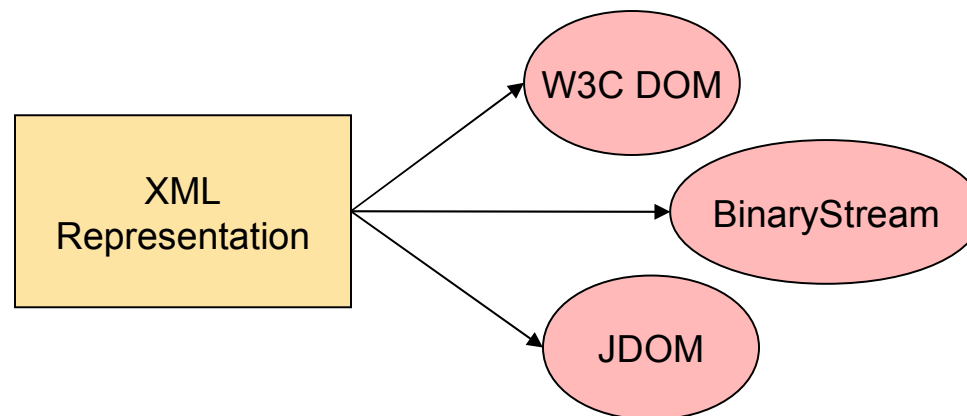
* not lossless nor bi-directional

◇ Outline

- Introduction to PoiNK
- Resource Models
- The PoiNK Resource Model
- PoiNK: NK3 to NK4
- New Documentation System
- New Unit Test Framework
- Additional Resources

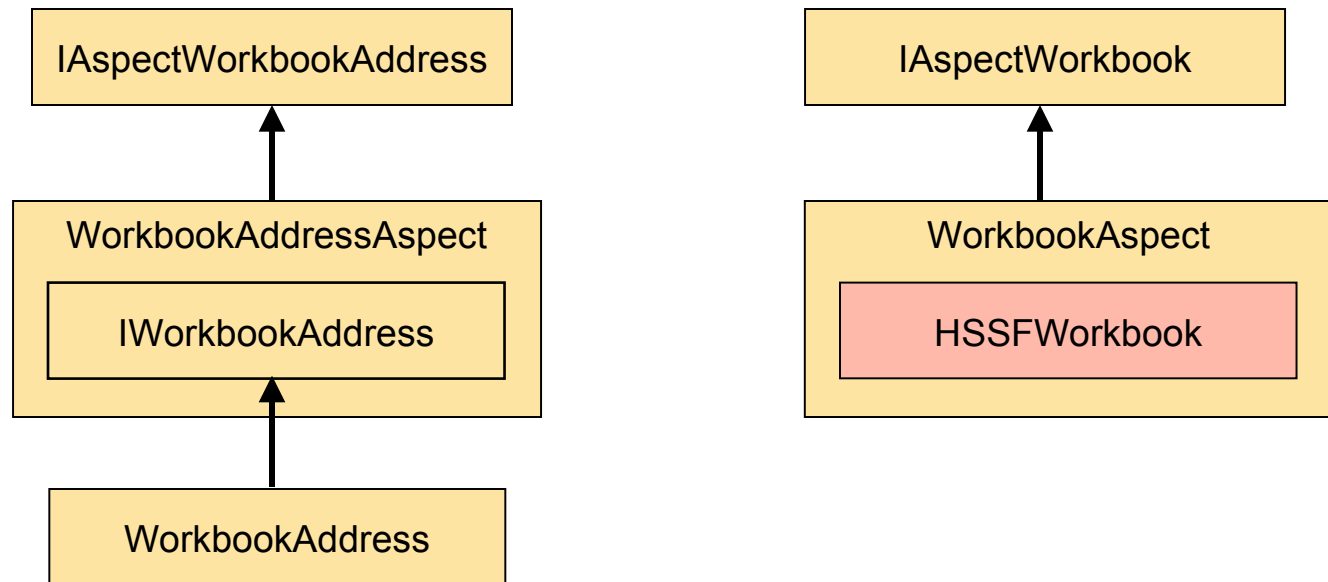
◇ NetKernel3 - Aspects

- Representations (of Abstract Information Types)
 - In NK3 each Representation has 1 or more **Aspects**
 - Alternate form (realization) of the Representation
 - Each Aspect is a different concrete data type
 - Often just a wrapper around an implementation object
 - Aspects used throughout the NK API



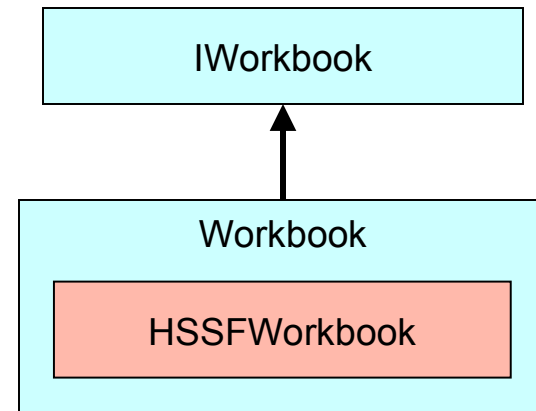
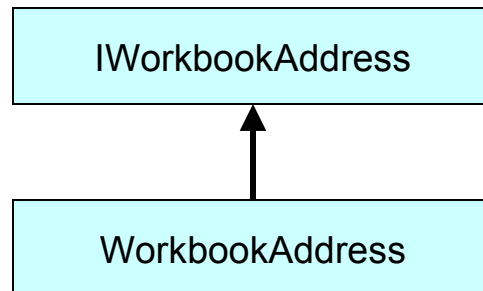
◇ PoiNK in NK3 - Aspects

- Aspects required to wrap domain classes
 - Sometimes creates an unnecessary layer
- WorkbookAddress aspect & Workbook aspect:



◇ PoiNK in NK4 - No Aspects

- Aspects gone: subsumed into Representations
 - Less code: smaller and cleaner
- WorkbookAddress & Workbook representations:



◇ PoiNK in NK4 - No Aspects

- Services Simplified:

```
IAspectWorkbook wbAspect = (IAspectWorkbook)
    context.sourceAspect("this:param:wb", IAspectWorkbook.class);

IAspectWorkbookAddress address = (IAspectWorkbookAddress)
    context.sourceAspect("this:param:address", IAspectWorkbookAddress.class);

// extract workbook and address of desired workbook component:
HSSFWorkbook wb = wbAspect.getWorkbookReadOnly();
IWorkbookAddress wbAddress = address.getWorkbookAddress();

//....use the extracted domain objects in the service code....
```



```
// get workbook and address of desired workbook component:
IWorkbook wb = (IWorkbook) context.source("arg:wb", IWorkbook.class);

IWorkbookAddress wbAddress = (IWorkbookAddress)
    context.source("arg:address", IWorkbookAddress.class);

//....use the domain objects in the service code....
```

◇ NetKernel4 - Endpoints

- Everything is an Endpoint (well...almost)
 - All logical addresses resolve to an Endpoint
 - Endpoints can be categorized by function:
 - Transports
 - Transreptors
 - Accessors:
 - SOURCEs, SINKs, Translators, etc...
 - Terminology: Killers and Existentialists (?)
 - Space manipulators:
 - Fileset, Import, Mapper, Overlay, Private, Rewrite

◇ NetKernel4 - Endpoints

- NK API provides hierarchy of Endpoint classes:

```
EndpointImpl (org.netkernel.layer0.urii)
  NKFEndpointImpl (org.netkernel.layer0.nkf.impl)
    StandardEndpointImpl (org.netkernel.layer0.module.standard.endpoint)
      StandardMonoEndpointImpl (org.netkernel.layer0.module.standard.endpoint)
        StandardAccessorImpl (org.netkernel.layer0.module.standard.endpoint)
        StandardTransportImpl (org.netkernel.layer0.module.standard.endpoint)
        StandardTransreptorImpl (org.netkernel.layer0.module.standard.endpoint)
-->      YourOwnTransreptorImpl (com.yourcompany.endpoint)
```

- Eases implementation of your own endpoints
- Allows implementation of multi-purpose endpoints
 - An "Accessor-Transport", for example

◇ PoiNK in NK4 - Endpoint Example

- Simple Transreptor: Workbook Address to XML

```
public class WorkbookAddressToXML extends StandardTransreptorImpl {

    public WorkbookAddressToXML () {
        // method inherited from EndpointImpl:
        declareThreadSafe();
        // methods inherited from StandardTransreptorImpl:
        declareDescription("Transrept a Workbook Address to XML representation.");
        declareFromRepresentation(IWorkbookAddress.class);
        declareToRepresentation(DOMXDA.class);
    }

    // implement abstract method inherited from StandardTransreptorImpl
    public void onTransrept (INKFRequestContext context) throws Exception {
        IWorkbookAddress wba =
            (IWorkbookAddress) context.sourcePrimary(IWorkbookAddress.class);
        String xmlStr = wba.toXMLString();
        Document doc = XMLUtils.parse(new StringReader(xmlStr));
        DOMXDA xml = new DOMXDA(doc);
        context.createResponseFrom(xml);
    }
}
```

◇ PoiNK in NK4 - Addressing Scheme

- Workbook Address Scheme
 - really a "mini" resource model
 - Has own representation, endpoint, & transreptors
 - defines a syntax for identifying parts of a Workbook
 - Accessor endpoint declared in module.xml file
 - Grammar argument defines base URI syntax:

```
// accessor endpoint declaration in module.xml file:  
...  
<accessor>  
  <id>WorkbookAddressScheme</id>  
  <description>The PoiNK Workbook Address Scheme accessor</description>  
  <class>com.tohono.poink.endpoint.WorkbookAddressScheme</class>  
  <grammar>wbAddr:<group name="addr"><regex type="anything"/></group></grammar>  
</accessor>  
...
```

◇ PoiNK in NK4 - Addressing Scheme

- Workbook Address Scheme
 - Endpoint implementation is simple
 - Just constructs new WorkbookAddress from argument:

```
public class WorkbookAddressScheme extends StandardAccessorImpl {  
  
    public WorkbookAddressScheme () {  
        declareThreadSafe();  
        declareDescription("Implements wbAddr: scheme.");  
    }  
  
    public void onSource (INKFRequestContext context) throws Exception {  
        String addr = context.getThisRequest().getArgumentValue("addr");  
        IWorkbookAddress wbAddr = new WorkbookAddress(addr);  
        context.createResponseFrom(wbAddr);  
    }  
}
```

◇ Outline

- Introduction to PoiNK
- Resource Models
- The PoiNK Resource Model
- PoiNK: NK3 to NK4
- **New Documentation System**
- New Unit Test Framework
- Additional Resources

◇ PoiNK in NK4 - New Documentation System

- Module documentation automatically discovered
- Configured by XML files in <MOD>/etc/system
- Built on a Wiki-interpreter engine
 - Speaks MediaWiki, Textile, Confluence, TracWiki
 - Supports HTML tables:
 - Supports extensible macro engines
 - 4 built-in macros: {xml}, {java}, {literal}, {svg}

◇ PoiNK in NK4 - New Documentation System

- Configured by XML files in <MOD>/etc/system
 - Docs.xml - defines individual documents:

```
<docs>
  <!-- PoiNK4 Application Guide -->
  <doc>
    <id>doc:com:tohono:poink:mod:docs:guide</id>
    <title>PoiNK4 Guide</title>
    <desc>Documentation for the POI for NetKernel4 Application</desc>
    <uri>res:/docs/guide/title.mw</uri>
    <keywords>PoiNK,guide</keywords>
  </doc>
  ...
  <!-- endpoints -->
  <doc>
    <id>doc:com:tohono:poink:mod:docs:guide:cellxml</id>
    <title>CellXML</title>
    <desc>CellXML endpoint</desc>
    <uri>res:/docs/guide/endpoints/cellxml.mw</uri>
    <keywords>PoiNK,guide,endpoints</keywords>
  </doc>
</docs>
```

◇ PoiNK in NK4 - New Documentation System

- Configured by XML files in <MOD>/etc/system
 - Books.xml - groups documents into books:

```
<books>
  <book>
    <id>book:com:tohono:poink:mod:docs:guide</id>
    <title>PoiNK4 Guide</title>
    <desc>Guide to the "POI for NetKernel4" Module</desc>
    <author>Tom Hicks</author>
    <publisher>Tohono Consulting LLC</publisher>
    <date>9/14/2008</date>
    <icon>res:/docs/doc-icon-violet.png</icon>
    <keywords></keywords>
    <toc>
      <item id="doc:com:tohono:poink:mod:docs:guide">
        <item id="doc:com:tohono:poink:mod:docs:guide:overview"/>
        <item id="doc:com:tohono:poink:mod:docs:guide:license"/>
        <item id="doc:com:tohono:poink:mod:docs:guide:reps">
          <item id="doc:com:tohono:poink:mod:docs:guide:wb"/>
          <item id="doc:com:tohono:poink:mod:docs:guide:wba"/>
        </item>
      </item>
    </toc>
  </book>
</books>
```

◇ PoiNK in NK4 - New Documentation System

- Built on a Wiki-interpreter engine
 - Speaks **MediaWiki**, Textile, Confluence, TracWiki

```
=CellXML=  
  
<blockquote>  
The cell is selected by the required <code>''address''</code> argument,  
which is a Workbook Address resource.  
</blockquote>  
  
<blockquote>  
For more information on the Workbook Address Scheme, see the  
[[doc:com:tohono:poink:mod:docs:guide:wbas|Workbook Address Scheme]] endpoint.  
</blockquote>  
  
==Module==  
  
: ''urn:com:tohono:poink:mod''  
  
==Syntax==  
  
===URI:===  
: ''active:cellXML''
```

◇ PoiNK in NK4 - New Documentation System

- Supports HTML tables:

```
===Arguments:===  
<blockquote>  
<table border="1" cellpadding="5">  
<tr bgcolor="#E0D7FF">  
<td align="center">' 'Argument' '</td>  
<td align="center">' 'Rules' '</td>  
<td align="center">' 'Description' '</td>  
</tr>  
<tr>  
<td>wb</td>  
<td><font color="red">mandatory</font></td>  
<td>A Workbook resource</td>  
</tr>  
<tr>  
<td>address</td>  
<td><font color="red">mandatory</font></td>  
<td>A Workbook Address resource which selects a single cell</td>  
</tr>  
</table>  
</blockquote>
```

◇ PoiNK in NK4 - New Documentation System

- Supports extensible macro engines
 - You can create your own formatting macros
- 4 built-in macros: {xml}, {java}, {literal}, {image}

```
==Example Usage==
```

```
===DPML:===
```

```
{xml}  
<closure>  
  <request assignment="response">  
    <base>active:cellXML</base>  
    <argument name="wb">res:/data/tasks.xls</argument>  
    <argument name="address">wbAddr:1!B9</argument>  
  </request>  
</closure>  
{/xml}
```

◇ Outline

- Introduction to PoiNK
- Resource Models
- The PoiNK Resource Model
- PoiNK: NK3 to NK4
- New Documentation System
- **New Unit Test Framework**
- Additional Resources

◇ PoiNK in NK4 - New Unit Test Framework

- Module tests automatically discovered
- Configured by XML file in <MOD>/etc/system
- XUnit test engine supports:
 - Tests and Testlists (suites)
 - Setup and Teardown code
 - User-defined Assertions
 - A set of built-in Assertions

◇ PoiNK in NK4 - New Unit Test Framework

- Configured by XML file in <MOD>/etc/system
 - Tests.xml - defines Tests and Testlists
 - **Example:** PoiNK4 Tests.xml file with 1 top-level testlist:

```
<tests>
  <test>
    <id>tests:com:tohono:poink:test:xunit</id>
    <name>PoiNK4 XUnit Tests</name>
    <desc>XUnit tests for the PoiNK4 Module</desc>
    <uri>res:/xunit/Testlist.xml</uri>
    <icon></icon>
  </test>
</tests>
```

◇ PoiNK in NK4 - New Unit Test Framework

- XUnit test engine supports:
 - Tests and Testlists
 - **Example:** PoiNK4 top-level testlist:

```
<testlist title="PoiNK XUnit Tests">
  <desc>
    <div>
      This is a suite of XUnit tests for the PoiNK Library.
    </div>
  </desc>

  <group title="CellXML Accessor Tests">
    <uri>res:/xunit/CellXML.xml</uri>
  </group>
  ...
  <group title="Example Scripts">
    <uri>res:/xunit/ExampleScripts.xml</uri>
  </group>
</testlist>
```

◇ PoiNK in NK4 - New Unit Test Framework

- XUnit test engine supports:
 - Tests and Testlists (suites)
 - **Example:** Test included from top-level testlist:

◇ PoiNK in NK4 - New Unit Test Framework

```
<testlist title="Cell Accessor XUnit Tests">
... <!-- important details omitted (which allow this testlist to really work) -->
  <test>
    <request>
      <base>active:cellValueString+wb@res:/data/tasks.xls+address@wbAddr:2!A1</base>
    </request>
    <assert>
      <class>java.lang.String</class>
      <stringEquals>The first cell</stringEquals>
    </assert>
  </test>

  <test>
    <request>
      <base>active:cellXML+wb@res:/data/tasks.xls+address@wbAddr:1!B9</base>
    </request>
    <assert>
      <xpath>count(/cell)=1</xpath>
      <xpath>/cell/@col='B'</xpath>
      <xpath>/cell/@row='9'</xpath>
      <xpath>/cell/@type='S'</xpath>
      <xpath>/cell/@formatCode='0'</xpath>
      <xpath>/cell/value/text()='Web Technology'</xpath>
    </assert>
  </test>
</testlist>
```

◇ PoiNK in NK4 - New Unit Test Framework

- XUnit test engine supports:
 - User-defined Assertions
 - **Example:** Assertion definition omitted from previous:

```
<testlist title="Cell Accessor XUnit Tests">
...
  <!-- load assertion definitions used in this test -->
  <group title="include xpath Assertion">
    <uri>res:/xunit/assertions/xpath_AD.xml</uri>
  </group>
...
</testlist>
```

```
<!-- user-defined assertion definition, externally defined to share -->
<assertDefinition name="xpath">
  <endpoint>GroovyRuntime</endpoint>
  <argument name="operator">res:/xunit/assertions/xpath.groovy</argument>
  <argument name="result">arg:test:result</argument>
  <argument name="xpath">arg:test:tagRef</argument> // or arg:test:tagValue
</assertDefinition>
```

◇ PoiNK in NK4 - New Unit Test Framework

- XUnit test engine supports:
 - User-defined Assertions
 - **Example:** Groovy implementation of Assertion:

```
// A custom assertion: test an xpath expression against an XML result
import org.netkernel.layer0.nkf.*
import org.ten60.netkernel.xml.xda.*

// Use the following if the tag argument is passed by-value:
// String xpath = context.source('arg:xpath', String.class)
String xpath = context.getRequest().getArgumentValue('xpath')
xda = context.source('arg:result').getRepresentation()
if (!(IXDAReadOnly.class).isInstance(xda)) {
    throw new IllegalArgumentException(
        'xpath assertion can only test results compatible with type IXDAReadOnly')
}
Boolean xeval = new Boolean(xda.eval(xpath))
INKFResponse resp = context.createResponseFrom(xeval)
resp.setExpiry(org.netkernel.layer0.nkf.INKFResponse.EXPIRY_ALWAYS)
```

◇ PoiNK in NK4 - New Unit Test Framework

- Framework provides a set of built-in Assertions:
 - maxTime - request must not exceed this time
 - minTime - request must exceed this time
 - class - representation must be instance of the class
 - mimetype - response must have this mimetype
 - expired - response must be expired
 - notExpired - response must not be expired
 - minTotalCost - response total cost, including all sub-request costs, must exceed this cost
 - maxTotalCost - response total cost, including all sub-request costs, must not exceed this cost
 - minLocalCost - response total cost, excluding any sub-request costs, must exceed this cost
 - maxLocalCost - response total cost, excluding any sub-request costs, must not exceed this cost
 - scope - response must have this scope depth
 - exception - response must be an exception and has this id

◇ PoiNK in NK4 - New Unit Test Framework

- XUnit test engine supports:
 - A set of built-in Assertions
 - **Example:** Test using built-in Assertions:

```
<test>
  <request>
    <base>active:sheetXML+wb@file:/tmp/wb.xls+address@wbAddr:2!</base>
  </request>
  <assert>
    <minTime>0</minTime>
    <maxTime>50</maxTime>
    <class>org.ten60.netkernel.xml.xda.DOMXDA</class>
    <mimetype>application/xml</mimetype>
  </assert>
</test>
```

◇ Outline

- Introduction to PoiNK
- Resource Models
- The PoiNK Resource Model
- PoiNK: NK3 to NK4
- New Documentation System
- New Unit Test Framework
- **Additional Resources**

◇ Resources

- NetKernel Sites and Forum:
 - <http://www.1060.org/> and <http://www.1060research.com/>
 - <http://www.1060.org/forum/>
- Apache POI Project
 - <http://poi.apache.org/>
- TheServerSide.com Articles:
 - <http://www.theserverside.com/tt/articles/article.tss?l=ARESTfulCorePart1>
 - <http://www.theserverside.com/tt/articles/article.tss?l=ARESTfulCorePart2>
 - <http://www.theserverside.com/tt/articles/article.tss?l=ARESTfulCorePart3>
 - <http://www.theserverside.com/tt/articles/article.tss?l=ARESTfulCorePart4>

◇ Questions

- Is there any time left for Questions?
 - There's always time for questions, cheeky monkeys
 - (apologies to Craig)
- Available for systems consulting, design, development, and training
- Feedback: hickst@tohono.com

◇ EXTRA SLIDES

◇ PoiNK in NK3 - Aspect Example

- WorkbookAspect in NK3:

```
public class WorkbookAspect implements IAspectWorkbook {  
  
    /** The underlying POI HSSF Workbook. */  
    private HSSFWorkbook workbook;  
  
    /** Constructor to create a new instance of Workbook Aspect. */  
    public WorkbookAspect (HSSFWorkbook workbook) {  
        this.workbook = workbook;  
    }  
  
    /**  
     * Return the underlying POI HSSF Workbook, which must not be written to!  
     * @see <tt>org.apache.poi.hssf.usermodel.HSSFWorkbook</tt>  
     */  
    public HSSFWorkbook getWorkbookReadOnly () throws Exception {  
        return workbook;  
    }  
}
```

◇ PoiNK in NK3 - Aspect Example

- WorkbookAddressAspect in NK3:

```
public class WorkbookAddressAspect implements IAspectWorkbookAddress {  
  
    /** The underlying address. */  
    private IWorkbookAddress wbAddress;  
  
    /**  
     * Constructor to create a new instance of the workbook address Aspect.  
     */  
    public WorkbookAddressAspect (IWorkbookAddress wbAddress) {  
        this.wbAddress = wbAddress;  
    }  
  
    /**  
     * Return the underlying address, which must not be written to!  
     *  
     * @see <tt>com.tohono.poink.util.IWorkbookAddress</tt>  
     */  
    public IWorkbookAddress getWorkbookAddressReadOnly () throws Exception {  
        return wbAddress;  
    }  
}
```