5. Java and .NET Interoperability

Proposal for OpenMI and Java Interoperability

OpenMI version 1.4 will contain both C# and Java versions of Standard defined.

Java interfaces will be created using semi-automatic procedure and will represent exact mirror of C# version. Except language differences between two frameworks.

In this way both Java and C# users of OpenMI should be happy because they will use framework classes from the platform of their choice: System.* in .NET Framework or org.* in JDK.

For example:

C# version

```
using System.Collections.Generic;

namespace OpenMI.Standard
{
    public interface ILinkableComponent
    {
        IList<IExchangeItem> InputItems { get; }
        IList<IExchangeItem> OutputItems { get; }

        event LinkableComponentDataChangedEventHandler DataChanged;
    }
}
```

May look like this:

Java version

```
package org.openmi.standard;

import java.util.List;

public interface ILinkableComponent
{
    public abstract List<IExchangeItem> getInputItems();
    public abstract List<IExchangeItem> getOutputItems();

    public abstract void addDataChangedListener(LinkableComponentDataChangedEvent e);
    public abstract void removeDataChangedListener(LinkableComponentDataChangedEvent e);
}
```

After that the following directions have to be investigated in order to re-use components between different platforms:

1. Binary converison of Java components implementing Java version of Standard to .NET and adapting them to C# version of interfaces using Adapter design pattern. Conversion should be possible using ikvmc.
2. Binary converison of .NET components implementing C# version of Standard to Java and adapting them to Java version of interfaces using Adapter design pattern. Maybe OpenJDK version of ikvmstub.
3. Ingetration of two platforms using WebServices or other remoting methods using one of libraries available for both platforms. For example using Remoting in Spring and Spring.NET.

Please note:

- **Package names** are defined in a way they expected to be:
    - **org.openmi.standard** for Java
    - **OpenMI.Standard** for .NET
- Collection interface like **IList<T>** defined as corresponding **List<T>** interface in Java.
- **Properties** converted to corresponding **getXxx()** and **setXxx()** methods.
- **Events** are transformed to Java analogs using **addDataChangedListener()**, **removeDataChangedListener()** in a way they expected to be in Java world.

Conclusions

- Java developers will use JDK class libraries and .NET developers will use .NET Framework class libraries. The best from two worlds. We do the rest to make them work together
- When **functional** changes to OpenMI Standard are required - they should be discussed and accepted in both versions without any exceptions. Any deviations may not be called OpenMI Standard anymore.

Tools

Tool	Description	Comment
IKVM.NET	Binary Java <-> .NET conversion	Latest development version is based on OpenJDK, see blog
Grasshopper	Binary .NET -> Java convertion	Does it use ikvm internaly?
JAD	Converts .class into .java	Also contains GUI called FrontEnd Plus