

Bimonthly newsletter of Jozef Aerts - XML4Pharma  
Schlossbergstrasse 20, 78224 Singen, Germany  
Phone: +49 7731 975044 Mobile : +41 76 3088403  
Mail: [Info@XML4Pharma.com](mailto:Info@XML4Pharma.com)  
January 2004



### **Move of XML4Pharma to Germany in completion phase**

The move of Computer Chemistry Consultancy – XML4Pharma has now come into its completion phase. So please note our new address and phone numbers (see top of this page).

### **Bristol-Meyers Squibb chooses for XForms**

FormsPlayer, a producer of XForms technology, recently announced that Bristol-Meyers Squibb has chosen to use XForms for its web applications. XForms is the successor of the classic HTML forms for capturing data from web forms.

XForms have several great advantages over HTML forms. First of all, they are device-independent. The same XForms can be used as well in a browser as on a PDA or a mobile phone. In the past, the forms had to be adapted for each of these separate devices.

Another advantage of XForms is that they can be automatically generated from XML documents. For example, CRF definitions in CDISC form can be automatically transformed into a set of XForms. One XSL stylesheet generally suffices to do the transformation for all CRFs, independent of their content.

Furthermore, XForms can be configured to do on-the-fly validation, checking the validity (format, ranges, etc..) before the data is send to the server. Typically, validation is already performed when

one jumps to a next field in an XForm. Essentially, this means the user is saving a good amount of time when filling in an XForm. And for the receiving party, it means much cleaner data in the database. FormsPlayer did not make public in which applications XForms will be used, but clinical data capture is of course an obvious candidate.

### **Final ODM 1.2 released**

Just a few days ago, the CDISC ODM team released the final version of the ODM 1.2. The standard now has an XML-Schema available (allowing automatic code generation, automatic database creation etc.), and XML-Signature as the mechanism to electronically sign off full ODM files for transfer to another party.

The ODM 1.2 is almost completely downwards compatible with the 1.1 version, so that it will be straightforward to upgrade ODM files from version 1.1 to version 1.2.

Hereby, we would like the congratulate the whole ODM team for this major step forward !

The ODM 1.2 release can be ownloaded from the following address:

<http://www.cdisc.org/models/odm/v1.2/>

### **Java 1.5 is coming !**

Sun Microsystems is expected to release Java 1.5 within the next months. We took a preview on the new functionality and found many interesting new features

which will make Java developers life even more easy, and surely lead to even more robust software.

Many expert C++ developers were missing the template classes in Java. Sun has now decided to define a new Collection framework to allow generic programming, similar to what is done in C++ with the template classes.

Another improvement is that casting when reading from an object will no longer be necessary. This will save developers a lot of code, and decrease the amount of testing that is necessary. Iteration is becoming easier. Whereas in Java 1.4 one has to ask for the size of a vector, or define a special Iterator, in Java 1.5, one can simply write something like: "for (String str : allStrings) ...", meaning: do this for each element of allStrings. This is a feature that has been copied from C# (the Microsoft counterpart of Java) and from C++. Another feature coming from C++ that was absent in Java is the Enumeration (type enum). C++ developers have the good habit of defining all the possible values for a variable as an enumeration. This will now also be possible in Java. Using the internal field VALUES, one then immediately gets a list of all possible values of the variable, without any iteration to do.

A newbie is the use of Metadata, which can be added to the code, and which can be very useful for automatic code generators and other tools. Metadata will start with the '@' sign.

Working with threads is never easy. Sun now implemented a new framework (package java.util.concurrent) to make this easier.

Java 1.3 and 1.4 were a big step forward for working with XML. Whereas in Java 1.2 it still took about 30 lines of code to

transform an XML file into e.g. XHTML using a stylesheet (XSLT), it nowadays requires no more than 3 lines. However, XML has evolved further, especially regarding namespaces, so that the libraries for working with XML had to be updated.

The new Java release will make it even easier to develop robust code, and is especially interesting for C++ developers who did not want to give up some features that Java didn't have. Furthermore, the updates of the libraries for working with XML, further strengthen the number 1 position of Java as the language to work with when dealing with XML.

### **Transforming XML documents into Excel worksheet**

XML becomes more and more common as data exchange format also between departments in companies. In these departments, Excel is sometimes used to analyze and visualize the data, e.g. using macros. There, the question arises: how can I automate import of XML data in the worksheets ?

Therefore, it is interesting to know that Apache has a java library available that makes such transformations very easy. This Apache project is named POI ("Poor Obfuscation Implementation"). The POI library can also be used in Java Servlets and in Frameworks like Cocoon, so that on-the-fly generation of Excel worksheets at the server side is perfectly possible.

For transformation of the structure of the native XML document to the structure of the resulting worksheet, one can use an XSLT stylesheet or do the transformation in memory, e.g. using dom4j (Document Object Model for Java). More information about Apache POI is available at

<http://jakarta.apache.org/poi>.