

Tools & Java



Steve Fisher / RAL
LBL - 13 May 2000

Contents



- XML tools for documentation
- Code checking
- “Together”
- Java
- ART

XML for documentation



- Apache are making new releases of Xalan
- DocBook also advancing
 - Style sheets produce reasonable HTML but the PDF is still poor.
- Lassi has left a good requirements DTD and XSL style sheet.
 - Transforms to HTML rather than via DocBook
 - Should we generate documentation DTDs
 - If so should we transform via DocBook?

Code checking



- CodeWizard from ParaSoft
 - Possibly not needed as expect this functionality soon with “Together”.
 - SDT at CERN will soon be making it available.
 - Version just released claims to have more checks and may have a properly supported RuleWizard.
 - The list price is \$995 node-locked on Windows.
 - It is available on Linux.
 - **Suggest we get some experience with existing (e.g. CERN) installations and find out the planned availability of the corresponding facilities in “Together” before purchasing licences.**

Together 3.2



- 3.2 functionality is good for C++ and excellent for Java.
- Now good price in the US for non profit use.
- Good results on interworking with Rational
- The web page reached from Software Tools page has a list of “features” to watch out for:
 - > 128M for decent performance
- Use Javadoc style comments for C++ to benefit from “Together” generated documentation.

Java



- Used some of the GEANT4 examples to try out Java C++ interface.
 - Build serialised Java events with collections of hits
 - very easily extended
 - Looked at both CORBA (ILU) and JNI.

CORBA (ILU)



- Xerox not investing much in ILU currently
 - they are not over interested in C++ but more in Java and Python.
- Defined an IDL "Outputting Interface".
- My first interface was too complex - I was not able to get it to work.
- With a very simple interface all was well and I had full type checking on each side.
- Issues about garbage collection not very clear - quite a lot of mystery left.

JNI



- JNI seemed much easier
- You can create a JVM
- Then access the full functionality of Java and work at the level of objects to which you can save "references" on the C++ side.

Conclusion on CORBA vs. JNI



- If you start from scratch and include CORBA as part of the architecture it might be very interesting - but it is not very easy to add in later.
- With the Athena “Algorithm” it should be relatively easy to write modules in different languages as the interface which an algorithm has to implement is very simple.
 - Plan to try it - starting with one of the Athena tutorial examples - probably the LAr.

Java Postscript



- All modifications to Athena should go in the direction of **enabling algorithms to be written in Java**.
- The interface to the services should be kept simple.

Java and the technical group.



- Julius Hrivnac and I are are preparing two documents:
 - How to use Java in ATLAS (almost ready for general comment)
 - Coding standards
 - Java C++ inter-working
 - Pros and Cons of Java for ATLAS (only ideas so far)

ART



- We want to make sure that we have a good “release” tool strategy.
 - This does not necessarily mean having the best release tool
- As mentioned at previous meetings we have a requirements document, which has been reviewed.
- Meeting which was widely announced but not too well attended where we tried to identify the next steps. The meeting was more emotional than I had expected and is summarised on <http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/OO/evaluations/art/>
- Request for extra ideas on good and bad features of existing SRT produced only a few replies.
 - The add-ons (e.g. web interface to submit multi-platform test builds) were liked
 - People do want an easier way to cooperate when working on related packages.

ART - Next steps



- Hope to get agreement by e-mail or another meeting if necessary on how a tool would be evaluated.
 - An important consideration, which is not covered by the requirements document is the ability to work with SRT for at least some change over period.
 - I believe we should try to change as fast as possible once we decide to do so to avoid the confusion of having two systems.
 - Will use mailing list of those who have expressed interest to discuss evaluation and then the developers list.
 - CMT appears to be a strong candidate to be evaluated according to the agreed criterion.
- Hope we can move quickly now so that we understand ASAP if we want to develop SRT further or not.