

CASE Tools for ATLAS

Steve Fisher / RAL

UML CASE Tools for ATLAS

- For the full story:
 - <http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/OO/tools/case/>
- A questionnaire was prepared for ATLAS
 - UML CASE (Computer Aided Software Engineering) tools such as ObjectTeam, Object Domain, Rose, StP or Together
 - to find out what people want from such a tool.
- Few replies; 2nd mail resulted in a few more.
- Other experiments invited to contribute.
- 19 people offered their general thoughts
 - 15 from ATLAS
- 7 people commented on Rational Rose and 8 on StP.

Questionnaire Results - 1

- People wanted a simple tool as well as a big one
 - there was also a suggestion that one tool might be enough if it did not rely upon network connections.
- There was partial agreement that people should be able to choose their own tools
 - there was more agreement that a collaboration should standardise though many felt that it was not feasible to standardise on just one.

Questionnaire results - 2

- Nobody thought that we should have one CASE tool for HEP but rather that we should encourage at least one tool within HEP.
- All agreed at some level that the ability to customise the tool was important. There was some support for the vendor getting it right.
 - I interpret this as meaning that we should be able to customise the tool but should not need to do a lot of work ourselves.
- People want the tool primarily on PCs (NT and Linux) and on Solaris

Questionnaire results - 3

- Table showing use made of Rose and StP showed little significant difference between what people used the tools for however there appears to be a greater satisfaction with Rose.
 - But note curious markings
- People agreed on the right price for the tool with an average license fee of SF740 for Rose and SF800 for StP.

What are the best points about the tool?

● Rose

- Available on "all" platforms at CERN
- Fast and nice GUI (on NT)
- Knows about OO
- Rather good support by vendor
- Speed
- Very flexible and customisable
- Interconnection between static model and dynamic model
- Lots of diagram types built in
- Supports most ISO C++
- Analyser for C++ and Java available
- Semantics of model
- Online database

● StP

- Complete
- Good, accessible underlying data model
- Open for customisation and with many customisation possibilities
- Produce good documentation for design review
- Diagrams are the source, and DB just acts as a cache
- Keeps trace of the design processes
- Stable
- Support for multiple methods
- Check model coherency
- Support for multiple developers
- Supported in Atlas

What are the 5 worse points about the tool

● Rose

- Crashes often (on UNIX)
- Very high memory and CPU requirements - slow
- GUI doesn't look same on all platforms
- Price
- Some C++ features are missing
- UML not fully supported
- Not intuitive
- Difficult to customise
- Printing and saving into ps-files bad
- Reverse engineering is not simple
- Still some bugs in analyser (Reverse engineered files may not work)
- Unreadable generated C++ code

● StP

- Speed of execution
- Speed of startup
- Difficult to learn
- It didn't generate any useful code
- Difficult to use (bad GUI design)
- It could not make sufficient use of "procedural" input
- Needs significant support on-site by system managers
- Requires central DB
- Bugs that lose you hours of work
- Doesn't run on Linux
- Migrating data from one release to the next
- Miserable presentation quality

What added functionality should the tool have?

● Rose

- Activity diagrams
- Code repository missing
- Maybe a better integration with the/a compiler/debugger
- Code checking (of non generated code)

● StP

- Better search facilities and organisational-aids
- Documentation templates independent of format
- More freedom in diagrams (free drawing for illustration, colours)
- Import from simpler tool
- XMI support
- Nicely coupled to the code
- Possibility to edit the text (mainly for requirements) as text, not only via GUI

Evaluation of tools

- Looked at 12 tools all of which support UML and run on Unix and NT. They must also support C++ or plan to very soon.
- There is quite a lot in my report on the web, but for now I will skip quickly over most of them.

And in fourth place...

- GDPPro 3.0
 - Maybe worth coming back to
- Innovator 6.1
 - This does not seem to be suitable.
- MagicDraw UML 3.1
 - Until it has C++ support it has no advantage over Argo and Together except for speed. They are considering moving to Swing - this will slow it down considerably.

... fourth place continued

- ObjectDomain 2.0
 - Until it supports associations it is not worth considering further.
- Rational Rose 98
 - The interface is quite nice but the forward engineered code is ugly and the reverse engineering is inadequate so we get a very expensive drawing tool which supports only the OMT/Booch subset of UML. It is not worth considering at the moment.
- Objecteering 4
 - The basic design of the tool is fine but not the implementation. Not yet suitable

... fourth place - final episode

- COOL:Jex
 - Even with big discounts this would be expensive. It is not clear that they have fully embraced UML and the central repository could be a problem (for remote working, and cvs)
- ORCA, SDT
 - I do not plan to consider this further.
- Opentool v3
 - An interesting product but they do not appear to have addressed reverse engineering at all.

StP UML 7.1

- Supports UML 1.1 The one we know and love - no extra cost
- Have installed version 7.1
 - they have made major improvements
 - only NT at the moment
 - < 10% of new customers want Unix
 - Next release (with XML generation) will initially be just on NT.
 - They plan to have a Unix version in September/October
- It is now a "real" Windows program -
 - no X
 - no MKS
 - either Sybase (shareable) or Access for the data base.

StP - continued

- On a 200MHz Pentium with 128M
 - it starts in about 3 seconds
 - each editor starts in about 3 seconds plus the time to load the diagram.
- Rebuild repository with 15 classes, 12 diagrams and 53 annotations.
 - local Sybase - 2mins 15 seconds
 - local Access - 2 mins 15 seconds
 - at CERN (client at RAL) > 30 mins
 - So we pay dearly for shared repository at CERN.
- The interface has been greatly improved,
 - change details of a method in a class table, then when saved, the changes appear in class diagrams automatically.
 - Several annotations can be worked on at once, and the whole lot saved together.
- Diagrams print out neatly without the jagged lines we are used to.

Verdict on StP

- They seem to have fixed the problems - on NT. If you are on NT you can easily be quite independent and move the system to a shared area as needed, but the Unix position looks gloomy.

Together

- Round-trip is automatic for both Java and C++.
 - The model and the code are always in step - this is rather strange at first - but seems to work very well.
- Performance is fine on a 450MHz Pentium - no swing.
- Easy to use
- You can define your own way of mapping the various UML constructs onto code.
- The model is split into many small text files which facilitates group working.
- Code held in memory?
- Scripting is with either Java or Python
- The whiteboard edition is free - but already useful.

Together continued

- Document generation is excellent - it can produce HTML using frames with all diagrams and code hyperlinked.
- Colours can be associated with different stereotypes.
- Adapted it to our code structure and found few problems. We may need a small script to periodically shift any new files to their proper home and change the #includes. Together will notice the changes and carry on happily.
- It is not currently possible to hide individual operations. (this might discourage people from writing monster classes).
- With the Blackdown JDK 1.1.6 on Linux quite a number of Java errors appear but the program carries on without crashing and it never mangled any code.
- No support for namespaces nor nested classes but they are working on it.

Verdict on Together

- With the proviso that you like diagrams to precisely reflect your code, this is an excellent and reasonably priced tool. They have a new version coming out at Java One which it has been suggested will have support for inner classes, partial support for namespaces and XMI (so we should be able to share models easily with Argo and the next release of StP)

Argo

- This is "... a domain-oriented design environment that provides cognitive support of object-oriented design. Argo/UML provides some of the same automation features of a commercial CASE tool, but it focuses on features that support the cognitive needs of designers. These cognitive needs are described by three cognitive theories: reflection-in-action, opportunistic design, and comprehension and problem solving." It is based directly on the UML meta-model and makes use of another UCI package, GEF, An Open Source Java Library for Connected Graph Editors.

Argo continued

- It generates Java and will shortly read Java.
- It is FREE (Open Source)
- It supports XMI.
- C++ support is planned in about 3 months time.
- It has lots of good ideas in the interface.
- It is still rather buggy and suffers from the slowness of the Java Swing GUI.
- There are about 80 people signed up as contributors so it could take off.

Verdict on Argo

- Until it supports C++ this has no major advantages compared to Together unless you want to work on design not linked to code. However as it is Open-Source, seems well designed and supports lots of notions for collaborative software development. It is worthy of consideration today where a close link with C++ is not required.

Recommendation

- So today for a heavy tool, I would recommend staying with StP, but as light tools Together and Argo are both very attractive. They are both portable Java applications which do not rely upon an external server and support (or will soon support XMI) I think that all 3 should be supported for the moment by ATLAS but that perhaps either Together or Argo will be dropped in the future.