CASE Tools compared CRISIS MANAGEMENT

FRANK HAUBENSCHILD

Tools for computer aided software engineering are treated as resources to counter the software crisis. But another question to be answered is whether the time spent on breaking in is proportionate to the benefits.

Ever more powerful hardware, and the resulting possibility of being able to solve ever more complex problems with the aid of software, led in the 1960s to the term software crisis being coined. Software developers realised that development tasks could no longer be managed without the implementation and the support of powerful tools. At the start of the 80s CASE (Computer Aided Software Engineering) joined in battle against the bogeyman of the software crisis.

Software development problems

Studies proved that about 50 per cent of the errors detected during a software development process occurred in the analysis and specification phase. A further 26 per cent come in the design phase and only about 25 per cent of the errors found stemmed from faulty implementation.

Obviously, and especially in the initial phases of software development, people were not working with enough attention to detail. CASE tools offer, especially for these early phases of development, a transparent and visual method to enable developers to view the system being created as a whole. This means they will not lose themselves at the early phases of development in implementation details.

The CASE tools presented here use Unified Modelling Language (UML) for the notation of software models. UML diagrams copy the connections of object-oriented systems visually and thus increase an understanding of the system just before the actual implementation.

But CASE tools offer more than just a purely visual support for the development process. They can be used for documentation purposes and offer the option of creating code entities out of class diagrams (and vice versa) (forward and reverse engineering), and in roundtrip engineering modifications which are made in the source text act directly on the visual model.

End of the software crisis?

So does this mean that CASE tools are a powerful weapon against the much complained about software crisis? Will all software projects now be completed by the promised deadline and meet customer requirements with the system realised? Well, not exactly, no. Many developers see CASE tools as more of a blot on the landscape, which hinders them in their creativity.

In the case of smaller projects, with a few hundred lines of code, this may well be fine, but projects with over 10,000 lines of code and developers working in parallel can be very hard to control without CASE tools, so the CASE tools available under Linux will be looked at more closely below.

Class diagrams with Dia

Unlike the other test candidates, the GPL program Dia is not strictly speaking a proper CASE-Tool. It is really for drawing diagrams of all kinds. The prototype is the commercial program Visio, well known in the Windows world.

For hobby-developers who want to create class diagrams and application cases and do not need to use code-generation, reverse engineering and the like, Dia is a good choice. It impresses by its simple and intuitive user guidance. Compared with the byte code-interpreted tools Together and ArgoUML, its rapidity also stands out.

Dia supports the following diagram types: UML (use case, class, sequence), ER (entity relationship), SADT, flow charts, networks and integrated circuits. New types of diagram can be added using simple XML files. The program loads diagram types in advance on start-up or dynamically, as required. Dia can load and store diagrams in XML and exports the formats EPS, SVG, WPG, CGM, PNG and TeX macros. XML files can either be saved direct in the ASCII format or compressed.

Note: You can download the command line-oriented tool Dia2code, which converts the class diagrams created under Dia into corresponding C++ or Java classes from the fifth URL listed below.

Overall, Dia is a stable tool which is good to use for the software development process on a small scale. People who want to avoid long break-in periods, and do not need the function overkill of Rational Rose or Together, are well served. For professional implementation in software development, however, Dia is not suitable.

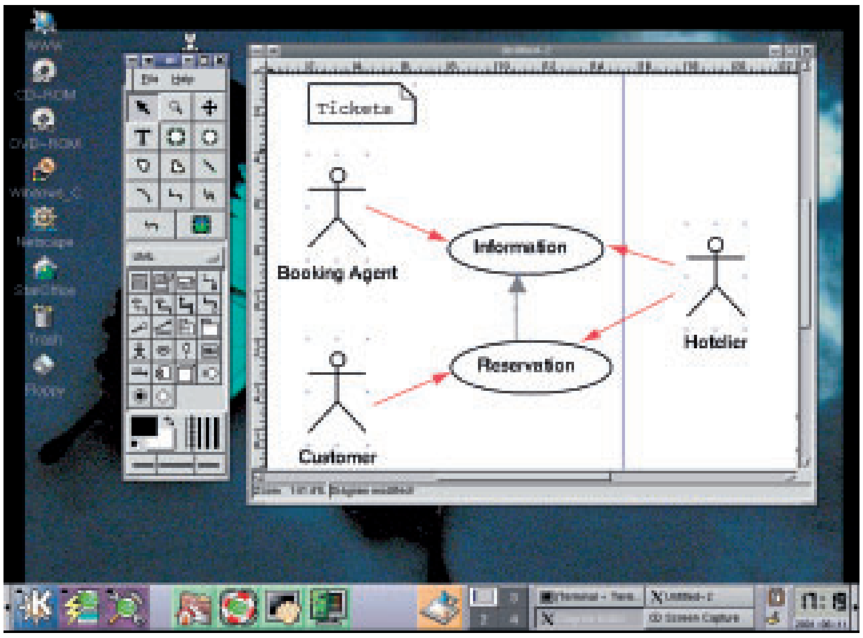


Figure 1: Dia in the Use-case-Modelling.

ArgoUML

ArgoUML is an open source project and completely implemented in Java. The tool also runs, due to this principle, on any platform with a virtual machine for Java 1.2. Since it is byte code-interpreted, though the speed of execution is not exactly electrifying.

ArgoUML meets the OMG standard for UML 1.3 and supports as diagram types class, state machine, use case, collaboration, activity and object/ component/ deployment diagrams. It is only sequence type diagrams that are not supported in the current version 0.8.1 – but these are planned for the next release. Also, ArgoUML supports the XML-based swap format XMI. ArgoUML uses it as standard memory mechanism and thus makes it possible to swap model data with other UML tools and in this way creates the basis for an open standard.

For code generation, ArgoUML supports only Java. UML diagrams can be saved in GIF, Postscript, EPS, PGML and SVG formats. One plus point is the wide variety of setting options for the print output.

After downloading and unpacking the tarball from the second URL listed below, all the necessary *jar* files are saved to the current directory. As long as you have installed JRE 1.2 or higher, ArgoUML can be started with the command

```
java -jar argouml.jar
```

After starting, you will be looking at a nice, tidy program interface.

The GUI of ArgoUML is split into four main parts. In the left upper corner there is the navigation panel in the form of a tree structure, via which one can access all previously installed elements of the model.

If you click on an element there, the properties of the element will be displayed in the detail panel (bottom right), and the element itself (for example a class) will be selected in the master window – the editing panel top right. The detail panel itself is in turn divided into eight index cards – for example under *Source* there is a preview of the generated Java source text of the selected UML element.

Developers who manage to-do lists with the aid of yellow Post-it stickers, and can hardly see their monitor for notes, will be glad of the to-do panel in the bottom left corner. Here the developer can manage to-dos sorted according to priority and thus has a constant overview of all items still outstanding. Apart from any to-dos you add yourself, ArgoUML also automatically adds to-dos to the list in accordance with design criteria and analyses of the model. These could be missing methods or class names.

If you go to *http://www.ArgoUML.org* and follow the *Tours* link, you'll find a good introduction to the most important features of the program. But even without a tutorial, thanks to the intuitive user instructions you will quickly get on your feet and can then defend yourself against the bogeyman of the software crisis. Apart from the speed of execution and the lack of one or two features, ArgoUML makes a very good impression and is therefore recommended for both hobby-developers as well as for the semi-professional domain. For the professional domain, there is no code generation for additional languages (especially C++), nor team support or reverse and roundtrip engineering.

Together Solo and Together Control Center

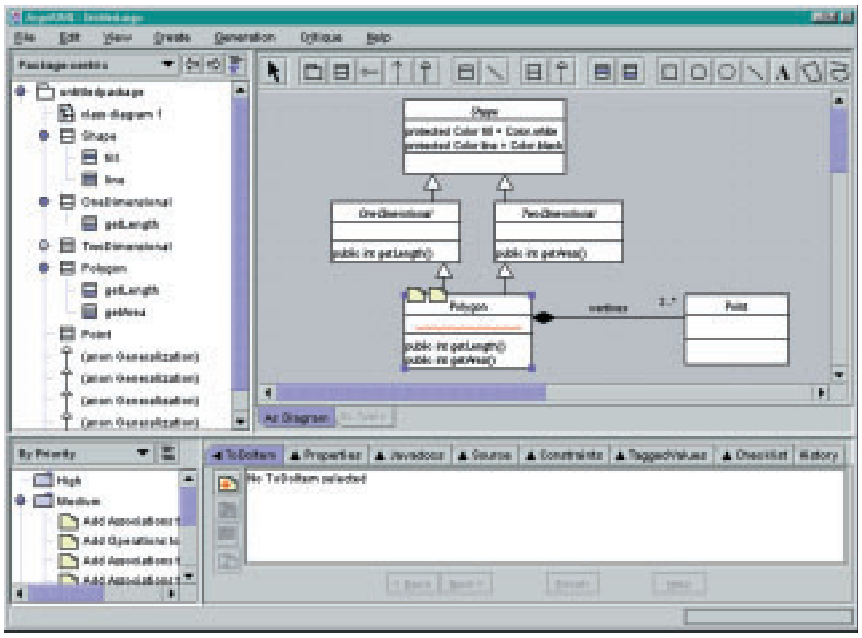
Together Soft offers its Together 4.2 product range in two versions. Both are byte code-interpreted, which, like ArgoUML, guarantees a sluggish rate of execution on ordinary hardware. Together Soft therefore also recommends P-III systems with 500MHz and 512MB RAM. As a virtual Java machine, JDK 1.3 is required.

Together supports class diagrams and UML diagram types (use case, sequence, collaboration, state, activity, component and deployment) for modelling. Code generation can be done in Java and C++, and reverse and roundtrip engineering as well as team support are provided. A project expert helps the developer to set up a new project. Here, for example, target language and directory settings can be adjusted.

The version Together Solo offers automatic documentation generation to HTML or RTF and supports the development of larger software projects via CVS. Together Solo also imports Rational Rose model files and exports UML diagrams as GIF or WMF. In addition to these there are EJB and forward and reverse engineering for sequence diagrams.

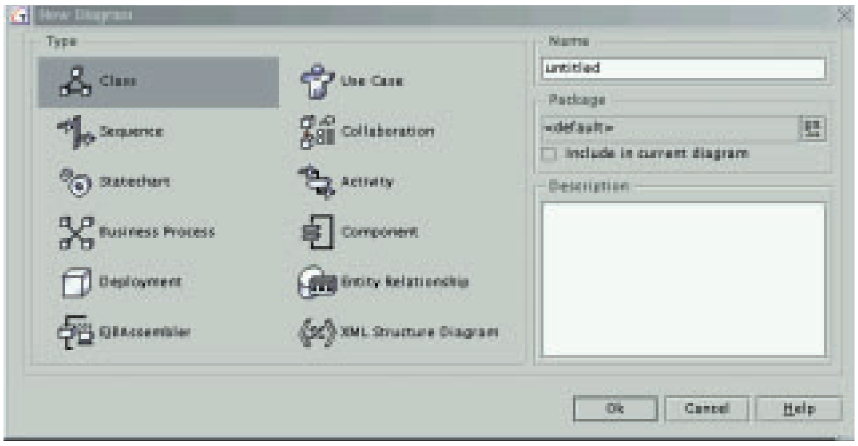
Diagrams and UML elements can be linked to each other. So for example classes can be linked to status diagrams, to get a better overview of the complete architecture. Apart from the target

[top]
Figure 2: Clear GUI from ArgoUML.



No ToDoItem selected

[below]
Figure 3: Together Control Center offers a wide variety of possible diagram types.



Name: untitled

Package: <default>

languages Java and C++, Together Solo supports the generation of IDL (Interface Definition Language) from class diagrams.

The high-end product Together Control Center supports, in addition to all the aforementioned UML diagrams, EJB Assembler and XML structure diagrams. It also excels due to the option of creating ER diagrams and offers JDBC roundtrip engineering for class and ER diagrams. Amendments to ER diagrams take effect directly on the database schemata of the DBMS below.

Another option is that of direct import of existing relations from a database as ER diagrams. Via a dialog window, the necessary settings (server type, database name, host, port, username and password) can be made for database communication. Databases supported are Oracle 7.3.x/8.x, DB2, MySQL, MS SQL, Cloudscape, ODBC/Access 97 and SequeLink/Oracle. Together Control Center also includes a debugger for Java – and is thus maturing into a complete IDE.

Overall the Together products make a very good impression. In terms of user guidance and handling Together leaves nothing to be desired and in this field is definitely, together with ArgoUML, ahead by a nose. Together is obviously intended for the professional domain, because of the enormous range of functions – which is also underlined by the respectable price of around £2800 (inclusive of one year's support) for Together Control Center. Version 5.0 is launching just as we go to press.

Rational Rose

Since the end of March the CASE tool Rational Rose, which stems from the Windows world, has been available for Linux.

What matters when it comes to CASE tools?

Depending on the size of the system to be realised and the number of developers involved in it, requirement profiles always differ. A hobby-developer, who wants to write a little tool with just a few hundred lines of code, will certainly not want to spend several thousand on a high-end product. The list below shows a few requirements, depending on the domain of application.

Main CASE-Tool features

Minimum Requirements
Support for common UML diagram types
Simple, intuitive user guidance
Easy-to-use diagram-layouter
Semi-professional requirements
Reverse Engineering
Roundtrip Engineering
Flexible documentation creation (for example in HTML)
Code generation (support for several target languages)
Professional requirements
Database support
Team support for larger projects
Open architecture for any/ potential expansions
In heterogeneous environments, support for as many platforms as possible

After registering, you can download the roughly 80 MB TGZ file. You will then receive, via email, a 15-day licence key and can test the program without restrictions. Rational specifies as platform Red Hat Linux 6.2 with Kernel 2.2.12.20.

After unpacking Rose is installed with the installation script *rs_install*. The binary *rose* starts from the directory *./bin*. Developers who have already worked with Rose under Windows will immediately feel at home, because the Linux GUI matches the one from the Windows world.

In comparison with ArgoUML and Together, the user instructions and the handling are limping behind – many features and setting options can only be found after a long search. For example the source text generated from the diagrams cannot be viewed directly, but has to be created manually by means of non-intuitive dialogs.

Of the CASE tools in the test field, Rose supports most languages: Java, C++, ADA 83, ADA 95 and CORBA IDL and DDL for database applications. Rose offers both roundtrip and reverse engineering. While reverse engineering a small Java sample project, Rose abruptly crashed, despite a correctly set *CLASSPATH*, without an error message – this is something the manufacturers must fix. Diagram types supported are class, use case, collaboration, sequence, component, state chart, deployment and activity. For notation UML, Booch and OMT can be used.

As to be expected for implementation in the professional domain, Rose has multi-user capability

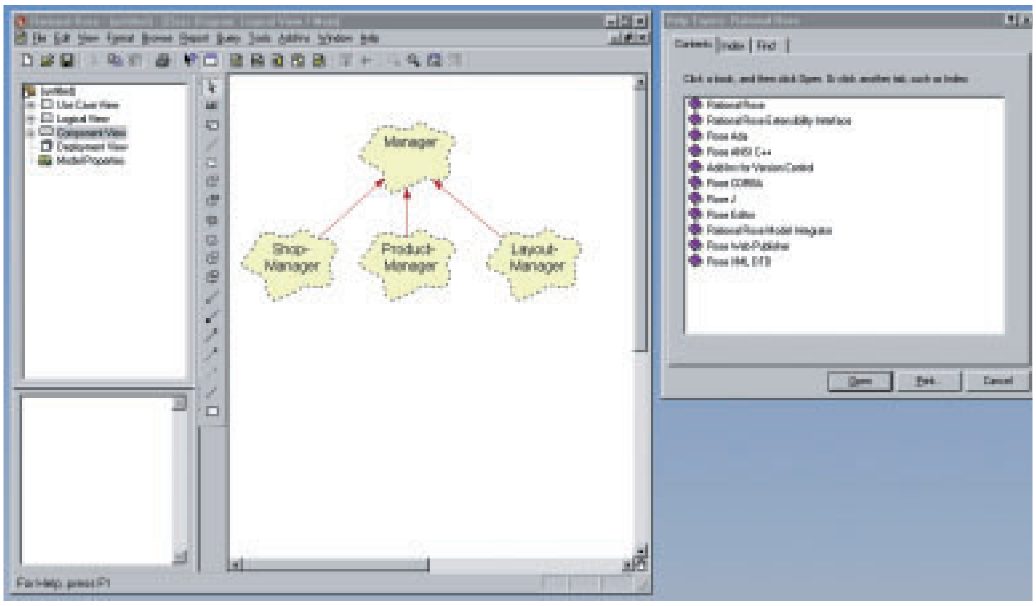


Figure 4: It is only the look-and-feel of KDE that hints at Linux – otherwise Rational Rose's Linux GUI matches the one from the Windows world.

CASE tools in overview

Product	**Dia 0.86**	**ArgoUML 0.8.1**	**Together Solo 4.2**	**Together Control Center 4.2**	**Rational Rose 2001**
Manufacturer	Alexander Larsson	University of California	Together Soft	Together Soft	Rational
Internet	www.lysator.liu.se/~alla/dia	www.ArgoUML.org	www.togethersoft.com	www.togethersoft.com	www.rational.com
Price (approx.)	GPL	free	£1600	£2800	£5500
Diagram types					
Use case	+	+	+	+	+
Class	+	+	+	+	+
State	-	+	+	+	+
Deployment	-	+	+	+	+
Activity	-	+	+	+	+
Collaboration	-	+	+	+	+
Sequence	+	-	+	+	+
Entity-Relationship	-	-	-	+	+
Code generation	only via Dia2code (C++, Java)	Java	Java, C++, IDL	Java, C++, IDL	C++, IDL, Java, Ada 83, Ada 95
Team support	-	-	+	+	+
Reverse engineering	-	-	+	+	+
Roundtrip engineering	-	-	+	+	+
Other	expandable by XML	open source, XMI support, needs JRE 1.2	Version 5.0 just out Version 5.0 just out	Java debugger,	Look & Feel matches Windows version

and supports developer groups. Rose makes a private working area for all developers, in which each has an individual via of the whole model. Modifications are thus restricted to the private working area until they are checked in to the CMVC (Configuration Management and Version Control System).

Models created with Rose can be put on an intranet or the Internet via a Web publisher as HTML files. As for notations, there is a choice of UML, Booch and OMT. Diagrams automatically integrate themselves as JPEG graphics. So for example for an API the complete documentation can be place don the Net. Using a preview function the result can be approved before the actual HTML generation.

Netscape Navigator 4.74 comes with it as Web browser. As to be expected, model data stored under Windows (MDL files) can also be used under Linux without any problems.

Overall Rose for Linux gives a stable impression – apart from the problem of reverse engineering of Java. In terms of speed it leaves the byte code-interpreted CASE tools ArgoUML and Together far behind. Because of the enormous range of functions of Rose and the wide variety of platforms supported (Windows, Sun Solaris, HP-UX, AIX, Irix, Compaq True 64 Unix) it is recommended for the pro.

But a minus point for the product is its high price of around £5,500 per commercial single user. Here's a thought for the manufacturer: A free version for non-commercial use would be a good idea.

Conclusion

Each of the CASE tools presented here has both advantages and disadvantages – whether one or two missing features or ergonomic weakness in the interface. ArgoUML and Together are in the lead in the field of operability. Together and Rose are restricted by the very cost of their licences to the professional domain. The free ArgoUML, though, scarcely needs to hide behind these programs, but the professional one is certainly lacking a few features.

All in all, CASE tools should be used more frequently in development. Because once they get used to them, developers will see them not as a blot on the landscape, but rather as helpful colleagues.■

Info

Dia: http://www.lysator.liu.se/~alla/dia
ArgoUML: http://www.ArgoUML.org
Together Soft: http://www.togethersoft.com
Rational: http://www.rational.com
Dia2code: http://dia2code.sourceforge.net

■