

Heart and Soul

CASE tools for Real Time Systems on the Testbed

By Peter Maurer, Karlsruhe, Germany (translated from German with the author's permission. First published in *Embedded Engineering*, Nov. 2001, Carl Hanser Verlag, Munich)

Those who are seriously considering the introduction of an object-oriented development process will sooner or later want to get hold of a UML tool. In the meantime, a number of software development tools are available for embedded systems. For instance, the "Artisan Real-Time Studio 4.1" from Artisan Software, "Rational Rose RealTime" from Rational Software, "Rhapsody" from I-Logix or the "Together Control Center" from Togethersoft. The slight differences can have a great effect upon the implementation in projects using C and C++, on real-time systems such as "VX-Works" and "OS-9" and on Windows.

In the last few years, object-oriented methods and languages have been increasingly used for the development of embedded or real-time systems.

The makers of UML tools have been busy taking this fact into account and are turning specifically to the developers of industrial software. With the purchase of an UML tool, the decision-makers and developers promise an increase in productivity in software development as well as an improvement in the quality of the finished products. If the tool in question is properly introduced and implemented, these targets are possible. The implementation of a CASE tool, however, takes time and costs well beyond that of the purchasing price of the software. CASE tools amortize only in the middle and long term, the introduction therefore has to be covered and actively supported by the management, otherwise there is a great danger of the tool falling into disuse after only a short period.

Together with a CASE tool, a software development process needs to be implemented which is finely tuned to the tool used. Each UML tool strongly advises the use of certain integral parts within the development process because without the appropriate development process, the introduction of an UML tool will very likely fail. In choosing a CASE tool one should therefore take care that the specifications required by the tool for the development process fit into ones own organization and method of working.

Useful – but to be applied carefully

Every UML tool is complex, as are the methods used with UML in object-oriented analysis and design. A UML tool should not be introduced into a project already running on a tight schedule. The best thing is to implement a pilot-project using a CASE tool. The experience gained from this can then be assessed before the method is implemented in larger projects. One should be very aware that a model of the software is created in the UML tool and

not the software itself. The CASE tool is capable of transforming a part of this model into source code (i.e. the actual software) ("forward engineering"). The other way round, the CASE tool can analyze source code and include it into its model ("reversal engineering").

During the development of software, both ways must be taken into consideration ("roundtrip engineering"). The problem with this is that the synchronization via roundtrip engineering applies only to a part of the model and thus only to only a part of the sources. As a rule, the actual implementation is left out. Sequence and collaboration diagrams are not subject to roundtrip engineering. All the same, most of the tools required for the embedded area support roundtrip engineering for state diagrams.

Apart from this, there is no clear feature which differentiates one UML tool for real-time systems from another. Most of the tools offer further features of interest for this field of application which,

however, differ from manufacturer to manufacturer. The selection of tools presented here was made on the basis of practical experience as reported by customers who use these tools for real-time and embedded applications.

The Teamworker:

Real-time Studio

Real Time Studio is from Artisan Software Tools, a company fully focused on the market for real-time and embedded systems. The Artisan model is saved in an object-oriented database which runs on Windows NT/2000 or Unix-OS (Solaris). All the clients thus access a common model. Modifications are immediate and visible when working in a team.

the following tools: “PVCS” from Merant, “VisualSourceSafe” from Microsoft, “ClearCase” from Rational and “CM Synergy” from Telelogic (previously Continuous). Unfortunately, the support of “CVS” is still lacking although this tool is widely distributed in the embedded field.

The use of Artisan is clear and simple. Anyone who has already worked with a CASE tool will be able to use this one with ease. Artisan supports most UML diagram types and instead of deployment and component diagrams, the system architecture diagram is available, which can visualize the distribution of the software over individual boards (Fig. 2).

The sequence diagrams keep track of focus control and can record conditional statements and loops consistently. The possibility of animating sequence diagrams has, time and again, proved to be very valuable.

Artisan also has table relationship diagrams for data modeling as well as the “Storage Mapper” which can automatically transfer classes and relations into the relational model. SQL for various RDBMS can be generated from the relational model using the “SQL-Generator”.

As well as the UML diagrams, Artisan supports a few of its own types of diagrams.

- *Concurrency Diagram*: This diagram takes into account the problem that in standard UML, the representation of the synchronization of threads and tasks is very difficult.
- *Constraint Diagram*: Using this diagram, the non-functional demands can be visualized and managed more concisely.
- *General Graphics Diagram*: As well as the diagrams inherent to models, general diagrams can be drawn up and managed in Artisan. This enables the entire project documentation to be maintained within the CASE tool.
- *Text Diagram*: This applies also to text diagrams. Here, freely formatted texts can be managed.

Artisan can generate and analyze ANSI-C++, Microsoft C++, ANSI-C and Java and ADA code. The corresponding synchronization tools are mature and simple to handle. The code generated is tidy and contains almost no intrusive instrumentation. Artisan is in the position to generate implementation code from

state diagrams and is able to re-analyze this code when necessary.

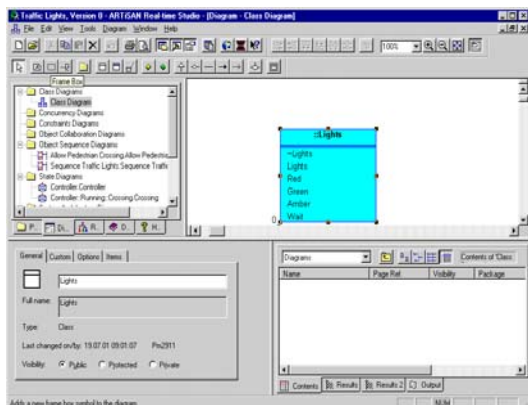


Fig. 1 : Using the properties dialogue box in Artisan, each active object can be edited

The clients run on the Windows desktop operating system from Windows 98 and Windows NT 4.0 upwards. As well as the “Professional Edition”, there is the “Modeler Edition”, which does not support code generation. Both editions can access the same repository. The repository can be integrated into different configuration management tools in order to guarantee a complete versioning of the model. Artisan supports

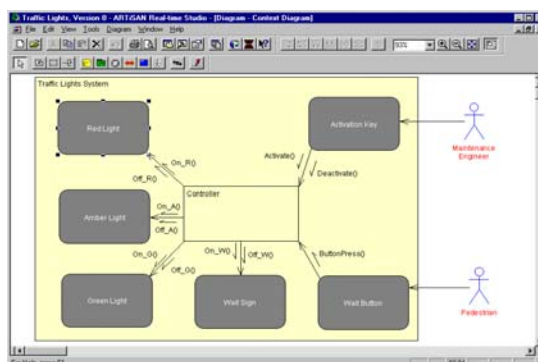


Fig. 2: With Artisan, the system architecture provides a bridge between soft and hardware

Using corresponding types and interfaces, Artisan supports “Corba” and can generate IDL code. Unfortunately, reversal engineering of IDL files is not possible. With its “Document Generator”, Artisan has at its disposal a mature tool for the creation of Word documents from the model. Updating of the generated documents is possible after modifying an existing document. With the help of various filters, all relevant documents for the model can be created. Unfortunately, apart from Word, Artisan does not support any further document formats such as HTML or PDF. Using the “Doors Synchronizer” Real-Time Studio offers a convenient connection to the Requirement Management System “Doors” (Dynamic Object Oriented Requirements Tracking System) from Telelogic.

**Powerful and Complex:
Rational Rose RealTime**

With Rational Rose, Rational offers a family of UML tools which, with “Rational Rose RealTime” and “Rational Rose RealTime Professional Edition” also has tools for embedded software development. Rational Rose RealTime runs on Windows and on various Unix platforms. Rational Rose saves its models in its own text-based format which can be directly transferred into a CM tool. Unfortunately, this tool only supports ClearCase (also from Rational) and Microsoft Visual SourceSafe (only on Windows NT/2000). The models are in text format so other CM tools can be used if required – albeit somewhat less conveniently. With Rational Rose RealTime, the entire development takes

place in the CASE tool including coding and testing. Rose RealTime therefore offers numerous features, although the interface does not always manage to hide the complexity of the tool (Fig. 3). The time taken getting used to this tool is relatively long. Apart from activity diagrams, Rose RealTime supports all UML diagrams, at the same time supporting some manufacturers’ extensions for the embedded area. The graphic representation in the printout and on the screen is very well done by Rational Rose (Fig. 4). An entity relationship diagram is not included.

Rational Rose RealTime can generate and analyze Java and ANSI-C code as well as ANSI and Microsoft C++ code. Reverse engineering in Java is not supported by Rose RealTime contrary to other members of the Rose family.

Fig. 3: The overview becomes somewhat clouded with Rational Rose RealTime.

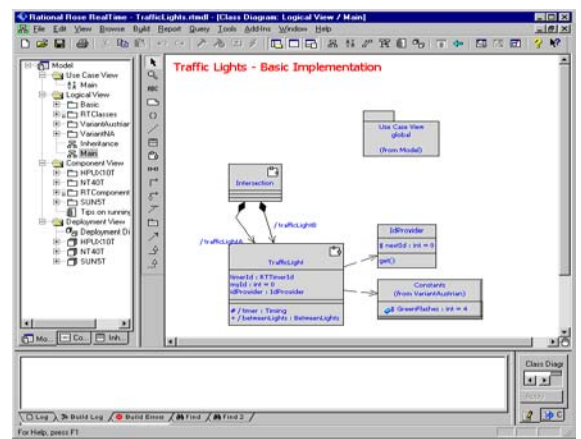


Fig. 4. Rational Rose deals with the UML diagrams very competently

Rose RealTime supports debugging of the software on the host system and in the professional edition also on the target. Support of Corba and COM is not available with Rational Rose RealTime, with other Rose tools it is. Rational’s “RequisitePro” is a smart tool for requirement management, and “SoDA” likewise as a documentation tool.

These tools are necessary in order to create Word documents from the model and for effective requirement management. Rational Rose provides interfaces for both products which belong – together with other Rational tools – to the “Rational Suite DevelopmentStudio RealTime”.

**The Classical Tool:
Rhapsody**

Rational Rose supports algorithmic code generation from State diagrams for C and C++. Reverse engineering is, however, not available. The code generated is easily readable for all supported languages and, in contrast to the earlier Rational Rose versions is minimally instrumented. The standard edition of

Rhapsody from I-Logix was the first UML Tool which specialized in embedded software development (Fig. 5). Rhapsody’s model is managed using single text-files, so-called units. Multiple users work together by exchanging units via a CM tool. Rhapsody connects CM tools using the “Microsoft Common

Source Code Control Interface". In this way, practically all CM tools running under Windows are available, including CVS. Besides the commercial "Developer Edition", there is also a free "Modeler Edition" of Rhapsody which offers limited code creation but does not include reverse or roundtrip engineering. Repositories from the modeler edition can be transferred onto the developer edition – the other way round is, however, not possible.

diagrams. Sequence, activity and state diagrams can be animated – a very useful feature. The focus control in the sequence diagram is missing, thus hindering the representation of time-dependant sequences.

Rhapsody can generate ANSI-C++, Microsoft C++, ANSI-C and Java code. Similar to Rational Rose Real Time, Rhapsody supports development completely in the CASE tool. Accordingly, there are interfaces for a whole range of target systems such as "OseDelta", "VxWorks", "pSOS+" and also "Windows CE". The corresponding IDEs can be integrated into Rhapsody.

Apart from the class and state diagrams, activity diagrams are also involved in code generation. Roundtrips are also possible with this diagram type. The code generated is clearly constructed, however, it contains instrumentations which have to be noted while editing. Rhapsody has its own runtime library (OXF) which requires code generated by Rhapsody. Rhapsody enables Corba modeling and can generate well-structured and readable IDL code.

Reverse engineering from IDL is not possible. Microsoft COM/DCOM is also supported by Rhapsody, COM interfaces can be generated and type libraries can be imported for the reverse engineering process.

With the "Rhapsody Reporter", Rhapsody has at its disposal a mature and user friendly tool to generate

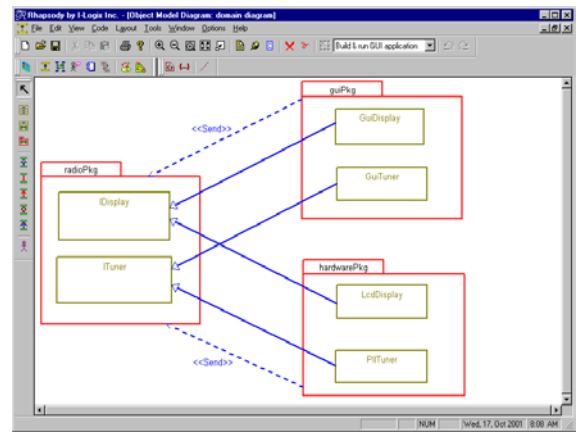


Fig. 5: The object model diagram in Rhapsody is a concise representation of the UML class diagrams

documentation from the model. Similar to Artisan, all the relevant documentation can be generated from the model without having to buy any extra tools. Rhapsody offers an interface to Doors so that a convenient tool for requirement management is at hand with which Rhapsody can work together.

Consistent, runs on all platforms: Together

Togethersoft offers with its product "Together" a CASE tool which, although not primarily applying to embedded developers, can be effectively used in this field. The tool has been developed solely in Java and therefore can be used on many platforms: Microsoft Windows, Linux, Sun Solaris, HP UX and Compaq Tru64 Unix are supported by the manufacturer. The resource consumption is, however, not to be underestimated: on Windows 512 MB working memory is needed with Together in order to ensure smooth operation. Together comes in the form "Together Control Center", and with reduced features as "Together Solo".

Tool Providers

Artisan Software

50933 Cologne
Tel. 02 21 / 4 85 22-60
Fax. 02 21 / 4 85 22-61
www.artisan.com

Rational Software

82041 Oberhaching
Tel. 089 / 6 28 30-0
Fax. 089 / 6 28 38-269
www.rational-software.de

I-Logix Deutschland

85591 Vaterstetten
Tel. 0 81 06 / 3 79 66-0
Fax. 0 81 06 / 3 79 66-6
www.ilogix.com

Togethersoft

70565 Stuttgart
Tel. 07 11 / 78 19 25 -50
Fax. 07 11 / 78 19 25 -99
www.togethersoft.com

Rhapsody can be run on all Windows desktop operating systems from Windows 95 upwards. Parallel to this, the Developer Edition also includes a variant for Solaris though only as an older version and only for C++. Rhapsody supports all UML diagram types apart from the deployment

Together saves the model in text-based files. Classes are stored directly in the source code (Java and C++) and class diagrams are directly modified along with the code. Working in a team is possible with Together using the CM tool. Together has command line interfaces leading to a whole range of CM tools, amongst others CVS, PVCS and ClearCase. Together offers all UML diagram types as well as a whole range of its own diagram types. Worth mentioning are Together's sequence diagrams for the real-time area which have excellent focus control as well as being able to visualize the generation and deleting of objects correctly. In addition to the class model, the Together Control Center supports data modeling for relational databases using ER diagrams and SQL generators. For J2EE, the Together Control Center offers a wide range of extensions and tools which are, however, hardly relevant to software development for embedded systems. Together supports generation from ANSI-C++ and Java but not ANSI-C. The C++ code generated is well-structured and readable and the instrumentations used by the tool are saved in commentaries which correspond to "javadoc". Together offers an extensive range of source code metrics to evaluate the Java and C++ sources. Together supports Corba in the model and is able to generate IDL code. The ensuing results are very satisfactory. Reverse engineering is also possible. With Together, documentation can also be generated from the model. There is a choice of data format between HTML, RTF and PDF. All documents relevant

for development can be generated via filters. The handling of the tools is simple and clear.

Conclusion:

Four well-developed tools

All the tools introduced here can be practically employed in embedded software development. Artisan RealTime Studio leaves the best impression due to the optimum support of teamwork via the repository. This tool also edits large models on averagely equipped systems without too much delay. The representation of the UML is also well done, all the tools are easy to use and enable uninterrupted workflow. The inclusion of CVS in the range of supported CM tools would, however, be an improvement.

Rhapsody is also an excellent software tool which, due to its close connection to the target, requires a completely different style of working than that of the Artisan tool. Rhapsody also represents the UML very well and concisely. The handling of the repositories, however, requires discipline among developers which, unfortunately, is rare in practice.

Rational Rose RealTime is a very powerful tool which leaves hardly anything to be desired. It is, however, very complex thus more taxing on personnel resources.

Peter Maurer is the managing director of Maurer & Treutner in Karlsruhe, Germany and is mainly occupied with the architecture and design of distributed systems in technical software applications

Contact

Maurer & Treutner

D-76135 Karlsruhe

Tel. 07 21 /933 49-13

Fax 07 21 / 933 49-29

www.maurer-treutner.de

This software and consultancy company specializes in the development of object-oriented software for customers from the industrial, trade and service sectors

