Technical Assessment of the ARTIST Tool

Dr. Jakob Axelsson

18 May 2002

Executive summary

This is a technical assessment of the tool ARTIST. It has been done through study of documentation and through questions to the developers of the tool. Based on that information, the following main conclusions have been made. The tool provides hybrid (i.e. discrete and continuous) modelling and simulation capabilities to UML tools. It thereby addresses a clearly existing problem in many industries that develop products containing advanced embedded systems. It also fills a market niche where there are currently few alternatives. The technical risks of the product are limited, since it is based mainly on proven and mature technology, which has been packaged in an open architecture. The tool follows the UML standard, and is well aligned with existing tools, thereby reducing the hurdle for potential customers.

Introduction

This report describes a technical assessment of the ARTIST tool, made by Dr. Jakob Axelsson on request from Wassim Hafez and Paul Beckwith at ABB in Canada. The study was done in mid May 2002, and was based on written documentation and answers to my list of questions, all provided by Wassim Hafez. The total time spent was about 20 h, and did not include trials with the tool itself. However, this is not a major drawback, since the most important information in an assessment like this is not seen in a tool demonstration, but consists of the principles used for implementing it. The usability of the tool has also not been assessed, but although this is an important success factor, it tends to be an area that is superficial, and can quite easily be changed. The emphasis of the report is thus on the core technology and architecture, which remains stable over time.

The scope of the assessment was to judge the technical relevance of the tool based on the market needs, and to validate that the technology chosen to implement the tool is likely to reach the desired objectives. The assessment does not make any attempts to quantify the commercial potential or make marketing forecasts, but focuses on the technical issues. The business plans of the company have not been studied.

The report is structured as follows. In the next section, an overview is given of the industrial applications, i.e. the market for modelling tools for embedded systems. Some fundamental requirements and properties are identified. Then, it is discussed to what extent this tool fulfils those market needs. This is followed by an assessment of the technology used in the implementation. Finally, some conclusions are drawn.

Applications

Many technical products are today seeing a rapid increase in *complexity*, which is enabled by the introduction of embedded systems that allow new and highly integrated functions to be developed. Industries that are part of this strong trend include automotive, aerospace, automation, power, and medical systems. The integration is cross-disciplinary, involving competencies from, e.g., mechanics, electronic hardware, and software. It is generally believed that *systems engineering* approaches must be used instead of domain specific engineering.

A major goal of the companies is to speed up development and allow early optimisation, and therefore computer *modelling* methods are sought. Such models need to capture system aspects such as structure, behaviour, performance, cost, and safety. However, a major problem is due to the cross-disciplinary nature of systems engineering, and there are currently no widely used methods that allow the *information* to be captured in a consistent way when it comes from many different domains. Standard practice is instead to use different tools for each analysis task, where the tools use various formats for input data. Therefore, the same system information needs to be duplicated for each tool, leading not only to extra work but also to an immense problem of keeping the information *consistent*.

Instead, a *holistic* approach is needed to support systems engineering. In such a method, a database is used to collect all the information in a common *repository*. This information is thus captured in one place, and can be extracted by different analysis and synthesis (e.g. design optimisation or code generation) tools to perform the necessary tasks. Organising the product information in this way is not a new idea, but has been used for a long time in the mechanical engineering domain. There, product data management (PDM) is used as a repository to store computer-aided design (CAD) data, which can be used by different tools such as finite-element analysis (FEM) or computer-aided manufacturing (CAM). However, the ideas have only recently started to spread to other domains.

To allow the information in the repository to be effectively used in different ways, care has to be taken when organising the database. It has turned out that *object-oriented* methods provide a good way of structuring information to allow reuse, and it has been a trend for over a decade now to move to such methods in the software engineering domain.

In the European automotive industry, there is currently a very strong *market trend* to move to object-oriented tools in the area of software development for embedded systems. A natural extension to this, which is starting to gain momentum right now, is to extend it to also handle *mechatronics*, where mechanics, electronics, and software are designed together to form one system. However, this will not be possible using the software oriented tools, since they can only handle discrete behaviour. For mechatronics, and other physical systems, hybrid models consisting of *continuous and discrete* behaviour is necessary, and this is a non-trivial extension to the presently used tools.

Market niche

Currently, industry uses a number of different tools in the design of embedded control systems. The tools, and tasks they can perform, include the following:

- **Control design:** Usually, tools based on continuous dynamics are needed, and the leading tool is *Matlab Simulink*.
- **Software design:** Discrete, logical behaviour is the focus of software design, and main tools include *Statemate, Rational Rose, Rhapsody, Artisan, Telelogic Tau,* and *ETAS.* The tools can handle both system structure and state machines, and some of them are based on the standard modelling languages UML and SDL.
- **Information structuring:** The above-mentioned UML tools also do information handling to some extent. Alternatives are the systems engineering tools that focus more on requirements managing and modelling, such as *Doors, RTM,* and *Core*.

Some tools, such as Matlab Simulink, do handle hybrid system modelling to some extent. However, they are not object-oriented, which means that the models become hard to use for other purposes than simulation, or for other systems than they were originally intended for. Reuse is on the other hand a strong point of the UML based tools, which are also good at information structuring. Their drawback is that they are mainly intended for non-embedded software-only systems, and do not handle modelling of the physical world (plant) very well, nor the interface between the controller and the plant. Some of them do not handle simulation of the systems at all.

There has been a market battle for some years between tools for system development, and there is now a clear tendency that UML based tools come out as winners. Vendors whose tools do not support UML are converting them to handle that language (an example of this is Telelogic Tau, who was originally an SDL tool, and Rhapsody, which was developed based on the older product Statemate).
Lots of companies are investing in UML technology, and are building up product information and large component libraries in that format. Many of them will move on from using the tools for embedded software, to also cover mechatronics.

The *ARTIST* tool is intended to fill the market niche of giving UML tools capabilities for modelling and simulation of hybrid systems. It does so by providing an open architecture, which makes it possible to include many other analysis techniques, although simulation of behaviour is currently the main focus. The tool does not compete with the existing UML tools, but relies on their capabilities for issues such as editing and documentation handling, and extends them to handle physical systems together with the embedded controllers in a holistic approach. By making alliances with established tool suppliers, it is not necessary to develop everything from scratch. It is also easier for customers to introduce the ARTIST tool in their existing tool chain, since it is an extension of what they already have in place. This reduces the risk to the customer. The tool is not domain specific, but is based on universal principles, which makes it applicable in many areas. By providing model libraries, it can however be tailored to specific domains, thereby increasing the efficiency of the modelling process.

Technology

After having established the existence of a market for tools like ARTIST, we now turn to the question whether the chosen technology is suitable to fulfil the market requirements. A number of key topics have been identified, that are described next.

Architecture: The typical use of ARTIST would be to work in a host UML tool that provides many of the editing and code generation capabilities. ARTIST is then used for simulation purposes, and accesses the information in the repository of the host tool. The aim is to support several hosts, and a potential problem of this is the lack of standards for the repository interface. However, it appears that the open architecture of ARTIST deals with this issue as well as possible. The interfaces rely on standard technology, e.g. ActiveX components, which is a fairly mature technology.

Modelling language: As mentioned above, the tool primarily uses UML as a modelling notation. The extensions necessary to provide the ability to model physical systems are modest, dealing mainly with physical equations and interfaces, and are rigorously described in a meta-model, thereby providing the semantic crispness necessary to implement them efficiently in analysis tools. The diagramming capabilities of UML are supplemented with an extended context diagram that goes beyond the standard. The need for this type of view has already been recognise by other tool vendors, and is therefore not dramatic.

Model generation: Before simulating the system in the repository, it has to be transformed into an executable model. In some cases, additional information is needed to tell the simulator more about the characteristics of the system, and this is handled in ARTIST by adding flags in the UML model. Wizards are used to add the necessary interfaces between model objects and the simulation kernel, thereby minimising the effort of the user. Graph analysis algorithms are used to organise the model in an efficient way for simulation, thereby removing the need for the user to think about the efficiency of the equation, and allowing him to focus on a clear description of the problem.

Simulation: The simulation engine is based on state-of-the-art numerical solvers. Key questions here are the speed of simulation, and the ability to handle difficult numerical problems. It appears that ARTIST would provide performance at least on par with other commercial tools, and the architecture would allow future development to incorporate new and more efficient algorithms. The tool also allows distributed simulation, whereby the burden of computation can be spread across several CPUs.

Libraries: Currently, the tool comes with basic physical libraries, similar to those of the standard modelling language Modelica. This is sufficient for initial trials, and can later be complemented with other libraries as the customer needs become clearer.

Since the tool is at an early stage of development, and is expected to be continuously extended and improved over many years, the most important aspect today is its ability to withstand changes. It appears that the chosen concept has that capability.

Conclusions

Based on the information provided, this assessment concludes that the ARTIST tool has a clear market niche in an area of large importance to many industries. The technical risk of the enterprise appears to be fairly low, since the tool is based on proven technology which is integrated in a good way in existing tools. This will also make it easier to introduce the tool in the organisation, where a UML based tool chain is likely to already exist. The open architecture of the tool gives it a good potential for future additions and improvements, and hence for an enterprise growth. The ultimate success of a product like ARTIST is a combination of many factors, including marketing and company management as well as competitor development, but from the information provided it seems unlikely that the enterprise would fail due to the technology not living up to the expectations.

To further increase the chances of success for the tool, it is recommended that the following actions be considered:

- There is currently an activity going on, aiming at defining an extension of the UML standard to cope with systems engineering. This is done within OMG in co-operation with INCOSE, and more information can be found on the Internet at http://syseng.omg.org. It would be beneficial to participate in this effort, or at least monitor it closely, to ensure that the forthcoming standard is compatible with the capabilities of ARTIST.
- A good tool is not a guarantee of success, if the users do not know how to use it efficiently. It is vital to also describe the process to be used, from initial requirements to completely verified models, when using the tool for product development. A good way could be to participate in the production of a professionally written textbook that can be used for training and that describes the methodology. The book could be sold with a complimentary CD containing a restricted version of the tool that could be used for exercises.
- A plan should be made for how to extend the tool with other capabilities, such as formal validation, hardware performance models, and cost models, that are important to the embedded systems designer for achieving a well-balanced solution. The benefit of this plan is that it can be verified already at this stage that the current tool architecture is capable of supporting such extensions, and thereby is future-proof.
- If the company intends to get involved with the automotive industry, establishing a European presence should have a high priority. The reason for this is that the European high-end car manufacturers, such as BMW, Mercedes, and Volvo, lead the development in advanced automotive electronics, and therefore they have the largest need for tools like ARTIST. North American OEMs are lagging behind in this field.

About the author

Jakob Axelsson studied computer science at Linköping University in Sweden and Ecole Polytechnique Fédérale de Lausanne in Switzerland. He received the M.Sc. and Ph.D. degrees from Linköping University. His doctoral thesis was on systems engineering of embedded real-time control systems, and focused on modelling, analysis, and synthesis of hardware and software architectures. Dr. Axelsson has been with Saab Dynamics in Linköping; ABB Corporate Research and ABB Power Generation in Baden, Switzerland; Volvo Technological Development, Carlstedt Research & Technology, and Volvo Car Corporation, all in Göteborg, Sweden. Positions include research programme management, project management, technology strategist, consulting, technical due diligence, and software tool development. He has published about 30 scientific conference and journal papers and technical reports, including work dealing with the extension of UML to cover non-discrete behaviour, architectural modelling, and cost analysis for embedded systems. He is a member of several international professional organisations, including the IEEE and the INCOSE, where he serves on the board of the Swedish chapter. He has held board positions at the Department of Computer Science of Linköping University and the School of Mathematics and Computer Science of Chalmers University of Technology in Göteborg, and has been in the program committees of a number of international scientific conferences. He has also received several professional awards in Sweden.