Tools Exhibits

Alberto Rodrigues da Silva

INESC-ID and Instituto Superior Técnico,
Rua Alves Redol, nº 9 – 1000-029 Lisboa, Portugal
alberto.silva@acm.org

Abstract. Live demonstrations of cutting-edge systems were an important and exciting part of the UML2004 conference. The tool exhibits session provided an excellent opportunity where participants analysed and viewed relevant UML and MDA related tools in action and discussed these systems with their creators or distributors. The tool exhibits session took place during the main conference, from October 13 to 15, and included the following live demos: (1) "seCAKE: A complete CASE tool with reuse support", by dTinf; (2) "Making UML diagrams accessible for visually impaired programmers", by FNB; (3) "TAU Generation2", by Telelogic; (4) "IBM Rational Rose XDE Products", by Sinfic; and (5) “BridgePoint Development Suite”, by Mentor Graphics. The tool exhibit contributions are described in this paper in the form of an extended summary. We briefly describe the related products according the data provided by their respective creators or distributors.

1 Introduction

In this paper we briefly describe the tools exhibits presented, from October 13 to 15, at the UML2004 conference in Lisbon. The tool exhibits session provided an excellent opportunity where participants analysed and viewed relevant UML and MDA related tools in action and discussed these systems with their creators or distributors.

The tool exhibits session included the following live demos:

- *seCAKE: A complete CASE tool with reuse support*, by dTinf;
- *Making UML diagrams accessible for visually impaired programmers*, by FNB;
- *TAU Generation2*, by Telelogic;
- *IBM Rational Rose XDE Products*, by Sinfic; and
- *BridgePoint Development Suite*, by Mentor Graphics.

The tools exhibits contributions are described in the following sections in the form of an extended summary. We describe the related tools according the data provided by their respective creators or distributors, or in same cases we also get information from the respective web sites. In the beginning of each section we provide contact information of related persons and organizations.

In *UML Modeling Languages and Applications: UML 2004 Satellite Activities*. LNCS 3297.

2 seCAKE: A complete CASE tool with reuse support (dTinf)

José M. Fuentes, Vicente García
dTinf, S.L. (Desarrollos para las Tecnologías de la Información), Madrid, Spain
{jmfuentes, vicente.garcia}@dtinf.es
http://www.dtinf.com

CAKE (Computer Aided Knowledge Environment) is a framework of tools, applications, and methodologies for identifying, classifying, retrieving, organizing, managing and reusing knowledge. The CAKE framework has been developed to manage, organize and reuse all different kinds of "knowledge assets" generated within an organization. The CAKE knowledge management and engineering environment, covering vertical applications, is based on a modern knowledge representation and classification schema called RSHP.

seCAKE is a computer-aided software engineering tool (CASE) designed for allowing software engineers to develop Information Systems using the Knowledge Management paradigm. Therefore seCAKE includes a whole set of enhancements for managing software as knowledge.

Tool Objectives

seCAKE has the following objectives: (1) A full coverage of the Software Development Process: unlike other CASE tools that are centered in UML, seCAKE covers other development stages such us requirements, estimation, testing… Furthermore, a trace system among all the project elements has been implemented. (2) Reuse support: by automatically index and retrieve software models using the RSHP [1] repository, seCAKE aims to open a new paradigm in software reuse. Together with the classical domain analysis and domain engineering techniques (also covered in CAKE) the indexing and retrieval capabilities of seCAKE enhance other classical classification schemas such as facets [2]. (3) A deep coverage of the UML standard: this objective is extremely important for retrieval and reuse purposes.

Main Features

The main features of the tool are the following:

- **High UML semantic accuracy:** in order to ease the retrieval of software models as much as possible, seCAKE tries to be complied with the UML standard. So, unlike many other tools, the core of the tool has been designed with the UML 1.5 metamodel in mind. seCAKE also includes a semantic checker that shows all the deviation of the current model with regards to the UML 1.5 metamodel and the UML wellformedness rules.
- **Advanced reuse system:** aside of the classical domain engineering techniques that are implemented in the CAKE tools, seCAKE includes a novel classification and retrieval module. This allows storing software models into the repository and

querying it. Those queries are not textual ones, but model sketches (pieces of UML diagrams, one or more requirements…). During the query, the system will look into the repository for those models that are close to the query. All those techniques allow an organization to enhance from an *ad-hoc* reuse maturity to a systematic one.

- **Requirements Support:** including version management, trace with the rest of elements of the model (risks, function point items, UML model elements, test suites and anomalies). Also, thanks to the NLP (Natural Language Processing) capabilities of the tool, seCAKE allows to incorporate requirements automatically from a textual document.
- **Risk Management:** aside of the trace, includes the management of the activities linked to risks, a priority model that shows the order of retirement and post-mortem information that gives an added value to the retrieval capabilities of the tool.
- **Project estimation Techniques:** including the function point technique, the postarchitectural model of COCOMO II, and the POP (*Predictive Object Point,* developed by PriceSystems*,* http://www.pricesystems.com) estimation method.
- **UML Models Comparer:** that allows to easily compare the similarities and differences of two UML models
- **Code and schemas generation:** including C# code generation, Java reverse and direct engineering and XML Schema generation
- **Test cases Management:** including test suits and test cases support, as well as anomalies support.
- **Report generation:** a project report could be generated with seCAKE. This report includes information about risks, requirements, estimation, test cases and especially UML.
- **Solution view:** that allows managing all the archives related to the project.
- **Modeling templates:** including the possibility of creating your own templates.
- **Knowledge sharing:** seCAKE will soon be deployed together with http://www.umlmodels.org. This web site will allow every seCAKE user to share their knowledge in two different senses: (1) seCAKE includes different forum threads; (2) the indexing and retrieval capabilities of seCAKE allow sharing software projects through the web site.

3 Making UML diagrams accessible for visually impaired programmers (FNB)

David Crombie, Sijo Dijkstra, George Ioannidis
FNB, Amsterdam, Netherlands
projects@fnb.nl, george.ioannidis@tzi.de
http://projects.fnb.nl/, http://www.tzi.de

Computer programming is one increasingly important area where blind and visually-impaired people have been able widely to participate in the employment market and in the educational sector [3]. Computer code has traditionally been text-based, and therefore accessible to visually impaired people using assistive technology (such as screen readers). In recent years the growth of software engineering has led to an increase in the use of programming tools that use visually rich presentation methods to facilitate development by sighted programmers. One of these tools is the Unified Modelling Language, a language used for modeling across many fields. It reflects the dominant object-oriented programming paradigm, and is increasingly popular in education and the workforce.

UML diagrams consist of nodes and connections between them. A great deal of information is contained in UML diagrams, much of it in graphical formats, such as the style of arrow-heads, and this remains inaccessible in tactile format. Furthermore, the text parts of the UML diagrams need to be converted to Braille to be understood as a screen reader cannot read text contained within diagrams. Often when a tactile version of larger diagrams has to be produced, the information needs to be redrawn over many tactile drawings and thereby the connections and coherence are easily lost. This imposes practical limits on UML representation for blind people [4]. In addition, UML is designed to be a co-operative modeling tool and a dynamic source of information for a development team to annotate, amend and change as the team designs a system. To take advantage of these features, interaction with the content is clearly necessary and without access to these diagrams, visually impaired people are excluded. The EU funded TeDUB project (Technical Diagram Understanding for the Blind) has been working on access to technical diagrams via interfaces to structured information [5,6] and has now developed a successful tool to access UML.

Technical Description and system architecture

The TeDUB system consists of two main parts, *DiagramInterpreter* and *DiagramNavigator*. DiagramInterpreter (semi-) automatically analyses existing diagrams from a number of formally defined domains and converts them into a representation that can be used by DiagramNavigator. DiagramInterpreter's core is the *knowledge processing unit*. It operates on a network of hypotheses and processes them incrementally until a semantic description of the whole diagram is found. The *image processing unit* analyses bitmap images and generates a first set of hypotheses based on the geometric information therein. Vector graphics files, which already contain explicit information about geometric primitives, can be used via DiagramInterpreter's SVG (Scalable Vector Graphics) import functionality. The *Annotator* allows a sighted user to interact with the interpretation process by inserting hypotheses manually and thus improving the quality of the interpretation as well as adding useful information not contained in the original diagram. All domain dependent aspects of DiagramInterpreter are externalised as formalised knowledge. Therefore, the system is designed to minimise the effort to incorporate a new type of diagram.

DiagramNavigator is the user interface component of the system and provides blind users with an interface to navigate and annotate these diagrams. It presents the diagram content obtained by DiagramInterpreter to the user. It also performs XSL transformation of XMI-format UML diagrams exported from UML design tools like

Rational Rose or ArgoUML into the same TeDUB form, presented by the same user interface. The great advantage of this latter approach is that the information contained in the diagram is converted perfectly into the TeDUB format: the variable result of image analysis of bitmaps is avoided.

The information of the drawing is modelled as a hierarchical tree structure of nodes with information attached to each node. A node might be a Class in a Class diagram or an Actor in a Use Case diagram. The nodes can be navigated either hierarchically or as a collection of connected graphs (depending on the type of diagram), using standard application components so as to be screen-reader independent. Input is via the keyboard or an optional tactile tablet. This is combined with a number of alternative interfaces, such as 2D and 3D surround sound (optional) and a game force feedback joystick. The joystick is used as a simple input and output tactile device and allows representation of the spatial information in the diagram, such as the position of nodes or the connections between them. Access to the information is also allowed by means of a representation of the content as a text-only internally-hyperlinked document, which can be navigated with the cursor keys as a standard text document. Another approach to give access to the information is a set of functions based on task analysis of the diagrams' types and use that structure and represent information according to user needs. For example, a Use Case diagram allows the presentation of all the Use Cases separate from the diagram, useful for developers checking the required Use Cases against implemented functions. To reflect the co-operative and modeling needs of UML users, simple editing functions are available, including annotating, renaming and limited editing of node information. All these components are designed to form a cohesive whole, usable when the additional hardware is not available but consistent when it is. The devices used are widely-available, low-cost devices rather than relatively expensive special purpose devices. The system is intended to be affordable and usable without special equipment by users with their own familiar screen reader.

Application and demonstration

Following extensive user studies across four countries, most participants expressed a very positive response to the system and felt it would be of significant value in educational and vocational environments. An interesting example was given by one participant, a software engineer, who had been made redundant when her department switched to UML, as she was unable to visualize the diagrams. A system like TeDUB would have resolved that problem. This stresses the importance of a tool to provide access to these kinds of diagrams. Taking the suggestions from this user study into account, the current version of the TeDUB system has significant potential to become a truly effective UML tool for visually impaired people.

4 TAU Generation2 (Telelogic Iberica)

Niklas Lagerblad
Telelogic Iberica, Madrid, Spain
niklas.lagerblad@telelogic.com
http://www.telelogic.com

TAU Generation2 (TAU G2) [7] is a family of model-centric and role-based tools that are among the first to implement the recently adopted UML 2.0 standard. The tool family consists of TAU/Developer for Software Engineers, TAU/Architect for Systems Engineers, and TAU/Tester for Test Engineers. TAU G2 builds on the model driven compilation technology perfected in TAU SDL Suite (a.k.a. TAU G1). TAU G1 proved that real-time software development can be automated using mature specifications languages such as Specification and Description Language (SDL) and Message Sequence Chart (MSC). Given that many of the advanced language features offered by SDL and MSC were adapted and incorporated into UML 2.0, there were compelling technical and market reasons to combine TAU G1's model driven compilation technology with UML 2.0 to produce TAU G2.

Main Features

TAU G2 provides the following features:

- **Precise and unambiguous system specification** – Engineers can visually specify systems using the precise, standardized and non-proprietary language of UML 2.0. This results in easy-to-understand, clear and unambiguous specifications.
- **Specification of behavior** – Whereas most system modeling tools allow only the specification of the system's architecture or structure, TAU G2 also allows engineers to visually specify the dynamic aspects of the system's behavior.
- **Automatic application generation** - TAU/Developer is the only tool that supports executable UML 2.0 models with behavioral specifications. Developers have access to pre-defined, verifiable code patterns that ensure high quality standards. With these capabilities, developers can automatically generate complete applications.
- **Dynamic model verification** - With fully controllable model simulation, engineers can verify their work in the analysis, design, and implementation phases. As a result, they can quickly locate and remove errors early when corrections are relatively easy and inexpensive.
- **Scalability** - Large scale systems can be specified and models can be mapped to how teams want to work, rather than having restrictions imposed by the tool. System architecture and behavior also can be modeled and viewed at the appropriate level of abstraction for the user.
- **Integrated requirements management via Telelogic DOORS®** - TAU G2 is integrated with Telelogic DOORS, the market leading requirements management solution.

- **Automated documentation via Telelogic DocExpress®** - TAU G2 is integrated with DocExpress, which provides automatic extraction and formatting of system or software application documentation.
- **Change and configuration management via Telelogic SYNERGY™ -** SYNERGY provides change and configuration management for TAU G2 and related products.

Discussion

It's inevitable that the software industry will eventually mature, and catch up with other industries based on engineering and automation, such as the computer hardware industry. At some point during this maturation process, it will become common practice for software engineers to specify their products using an architectural blueprint language, such as UML 2.0.

During this evolution it will also become common sense for engineers to apply a model driven development approach, such as MDA. This approach will need to be supported by power tools, such as TAU G2, that faithfully and efficiently implement the blueprint language, so that it can automate the mapping transformations across the models that represent the various process phases.

What should we expect from Model Driven Architectures during the next decade? We should expect them to evolve from conceptual architectures into technical architectures that solve complex business and technology problems.

What should we expect from MDA tools, such as TAU G2? In general, we should expect progressively tighter integration with traditional Integrated Development Environments (IDEs), and improved integration with requirements management and testing tools. In the case of TAU G2, this means seamless integration with DOORS and TAU/Tester. DOORS requirements can already be visualized as UML elements and TAU/Tester test scripts are being updated to align it with the recently adopted UML 2.0 Profile for Testing.

This future model driven IDEs will allow developers to efficiently shift and downshift through all the abstraction gears associated with a full application lifecycle. In these high productivity development environments, programming code will likely devolve into a machine readable artifact that is rarely viewed by humans. Released from the drudgery of producing and maintaining low-level implementation code, software developers will be able to pursue more creative activities that return greater business value, such as architecture, analysis and design.

5 IBM Rational Rose XDE Products (Sinfic)

Paulo Figueiredo
Sinfic, Lisbon, Portugal
pfigueiredo@sinfic.pt
http://www.sinfic.pt, http://www.ibm.com

The IBM Rational Rose XDE product family [8, 9] combines the rich heritage of the award-winning IBM Rational Rose family with IBM Rational XDE, which extends your IDE with the world's most advanced software modeling capabilities. Though packaged and purchased together, Rose and XDE are installed separately. They can be used in combination, with some limitations, but most users will benefit from primarily using one or the other.

IBM Rational Rose XDE Developer editions offer software designers and developers a rich set of model-driven development and runtime analysis capabilities for building quality software applications. They offer complete visual design and development environments that address the needs of organizations targeting both J2EE-based and .NET-based systems. Our solution allows users to work inside the included Eclipse IDE, or it can be installed into the IBM WebSphere Studio Application Developer and Integration Edition IDEs, and Microsoft Visual Studio .NET. Rational Rose is also included to integrate with Microsoft Visual Studio and other leading Java platform IDEs. The IBM Rational Rose XDE Developer products extend your development environment or integrate with the one you are already using.

Rational Rose XDE Modeler enables architects and designers to practice model-driven development with the Unified Modeling Language (UML). Such users can produce platform independent models of software architecture, business needs, reusable assets, and management-level communication. Industry standard UML support and a powerful pattern engine allow users to create a semantically rich application architecture that meets business needs and is readily understood by the development team. Architects and designers can use Rational Rose XDE Modeler's multi-model support to separate concerns of analysis, architecture, design and implementation. Developers can use architectural models and patterns as the basis for implementation, thereby accelerating the development of applications to conform to their architecture. Further, features such as free-form modeling, Web publishing and reporting allow users to share architecture and designs with all stakeholders, whether or not the stakeholders use Rational Rose XDE Modeler.

Rational Rose XDE Developer also allows data architects and DBAs to create logical and physical data models for DB2, Oracle, Sybase, and SQL Server databases. Architects can follow a top-down approach, creating a logical model of data requirements, transforming it into a physical database design, and then deploy that design to a database. Or, start from an existing database and reverse engineer the schema into a physical data model. Sophisticated "Compare and Synch" capabilities allow you to compare a physical data model to a database, and reconcile differences. Further, since the data modeling capabilities share the same environment as the application modeling environment, it is easy to keep data models and application models in synch.

No need to switch between tools for design/development

IBM Rational Rose XDE products extend your development environment or integrate with the one you are already using. The developer doesn’t need to switch between the IDE and the visual modeling tool. Purify plus it’s also integrated into websphere studio an Microsoft Visual Studio .Net, that allows use the capabilities of this performance and runtime analysis tool without leaving the IDE.

Main features

The relevant features of the IBM Rational Rose XDE products are the following ones:
- Model-driven development with UML support
- Roundtrip engineering Java,C++, and Visual Studio languages
- Automatic or on demand model-code synchronization
- User-definable patterns and code templates
- Runtime analysis including visual execution trace
- Assisted modeling
- Multiple model support for Model-Driven Architecture
- Free-form diagramming
- Logical and physical database design
- Web publishing and reporting

6 BridgePoint Development Suite (Mentor Graphics)

Thomas Ulber
Mentor Graphics, Munchen, German
thomas_ulber@mentor.com
http://www.mentor.com

The Nucleus BridgePoint Development Suite [10] accelerates the development of real-time, embedded, technical, and simulation systems. Nucleus BridgePoint provides the most complete and productive environment for Agile MDA (Model Driven Architecture) [11] and the development of Executable and Translatable UML models. It has been used to develop hundreds of the most demanding systems including flight-critical launch vehicles, life-critical medical systems, large fault-tolerant distributed telecom systems, highly resource-constrained consumer electronics, and large-scale distributed discrete-event HLA simulation systems.

Project Technology founders pioneered the concepts of Executable and Translatable UML and have shaped the OMG standards that make UML executable. Accelerated Technology continues Project Technology's leadership role by delivering the benefits of MDA automation to development teams today.

Agile MDA – Executable and Translatable UML

Agile MDA provides a unique opportunity to accelerate the development and improve the quality of real-time, technical, and simulation systems. OMG-compliant

Executable and Translatable UML (xtUML) provides the basis for Agile MDA and its significant benefits.

xtUML Platform Independent Models (PIMs) completely and concisely describe what the system does and are fully testable and executable. The three orthogonal system aspects are defined with Class Diagrams (Data), State Charts (Control) and OMG-compliant Object Action Language (Processing). Domain Package Diagrams provide support for effective subject-matter partitioning and system scale-up. Through early PIM testing and defect elimination, system quality is dramatically increased, and downstream integration, test and maintenance activities are streamlined.

xtUML PIMs are automatically translated, by customizable model compilers comprised of translation rules and patterns, to generate 100% complete target code. The generated code directly reflects both the application behavior defined and tested in the PIM, and the design and implementation specifics defined and tested in the model compiler. Changes to the application defined by the PIM or to the software architecture defined in the model compiler are automatically reflected in the system's generated code.

Benefits of effective automation and implementation of the Agile MDA process include: accelerated development and maintenance, greatly increased system quality, effective performance and resource optimization, streamlined platform migration, and large-scale reuse.

Nucleus BridgePoint Development Suite

The Nucleus BridgePoint Development Suite provides complete support for Agile MDA and the construction, debug, test, management, and translation of Executable and Translatable UML (xtUML) PIMs. The Nucleus BridgePoint Development Suite provides:

- Guided development of high-quality xtUML PIMs.
- Early (pre-code) execution, debug and test of xtUML PIMs.
- Customizable translation of xtUML PIMs into target-optimized 100% complete code.
- Model-level test and debug of complete or partial systems comprised of generated and non-generated code.
- Powerful performance-tuning and system-resource optimization.
- Effective reuse of xtUML PIMs and PIM components across multiple releases, products and product lines.
- Robust model configuration management including concurrent branches, overlapping configurations, and versioned domains, subsystems, and class statecharts.
- Multi-user, heterogeneous network support.
- Multi-level subject matter partitioning for effective project scale-up and accelerated iterative development.
- Nucleus BridgePoint Model Builder
- Building quality into UML models that execute and translate

6 Conclusions

The tools exhibits session, integrated in the main activities of the UML2004 conference, was a key opportunity to gather both the tools creators or distributors and interested people from the academic and industry world. It was an exceptional occasion to see live demonstrations of cutting-edge systems.

In an overview analyse we verify that the number of features and level of complexity of the most part of the shown tools is very high.

For example, the tools suite from IBM (e.g., Rational Rose XDE [8], Rational Software Architect [9]), from Telelogic (e.g., TAU G2, Doors, Synergy) or even from dTinf (e.g., seCAKE) support several activities of the software development process. Predominantly, they better support the following activities: requirement engineering, visual modelling and development. A common aspect shared by these tools is their support for visual modelling in UML and also an emergent set of features concerning the MDA concept.

The sophistication level of the current state presented in these tools is something remarkable, which raises several questions, such as: What is the necessary effort to learn how to use them productively? Is it possible and how can we tailor or customize these tools according business needs or specific projects needs (e.g., team size, project complexity)? Is it possible to integrate these tools with others (such as IDEs or project management tools)? How effort does this require?

In a different way, the BridgePoint Development Suite addresses the precise domain area of real-time, embedded, technical, and simulation systems. BridePoint applies innovative and interesting ideas around the Agile MDA and xtUML (executable and translated UML) model concepts. Based on these concepts, BridgePoint provides a set of tools that supports the design, debug, test, animation, and translation of xtUML platform independent models.

Finally, the TeDUB system is the result of the EU funded TeDUB ("Technical Diagram Understanding for the Blind") project and addresses the ability to make UML models accessible for visually impaired people (particularly, technical people). Definitely, TeDUB is a system for a very narrow but important group of people, and we believe that it is still in an initial stage of development.

References

1. Lloréns, J., Morato, J., Génova, G.: RSHP: an information representation model based on relationships. Springer Verlag in the LNCS series; Soft-Computing in Software.
2. Prieto-Díaz, R., Freeman, P.: "Classifying software for reusability". IEEE software (January 1987) 6-16.
3. C. Baillie, O. K. Burmeister & J. H. Hamlyn-Harris, "Web-based Teaching: Communicating Technical Diagrams with the Vision Impaired" (available at http://opax.swin.edu.au/~303207/OZeWAI20031.html).

4. M. Horstmann, C. Hagen, A. King, S. Dijkstra, D. Crombie, D. G. Evans, G. T. Ioannidis, P. Blenkhorn, O. Herzog, Ch. Schlieder (2004) 'TeDUB: Automatic Interpretation and Presentation of Technical Diagrams for Blind People', Proceedings Conference and Workshop on Assistive Technologies for Vision and Hearing Impairment, University of Glasgow.
5. P. Blenkhorn, D. Crombie, S. Dijkstra, G. Evans, B. Gallager, C. Hagen, M. Horstmann, G. Ioannidis, A. King, M. Magennis, H. Petrie, A. O'Neil, C. Schlieder & J. Wood, Access to Technical Diagrams for Blind People, AAATE, In Craddock, McCormack, Rielly & Knops (Eds.), Assistive Technology – Shaping the Future (Proc. AAATE 2003), pp 466 – 470, 2003.
6. H. Petrie et al, "TeDUB: A System for resenting and Exploring Technical Drawings for Blind People", In K. Miesenberger, J. Klaus, & W. Zagler (Eds.), Computers Helping People with Special Needs, Proc. 8th ICCHP, Lecture Notes in Computer Science, No. 2398, Springer, pp 537 - 539, July 2002.
7. Telelogic *TAU Generation2*. http://www.telelogic.com/products/tau/tg2.cfm
8. IBM *Rational Software Architect*. http://www-306.ibm.com/software/awdtools/architect/swarchitect/index.html
9. IBM *Rational XDE Developer*. http://www-306.ibm.com/software/awdtools/developer/rosexde/
10. Mentor Graphics *BridgePoint Development Suite*. http://www.mentor.com/products/embedded_software/nucleus_modeling/nucleus_bridgepoint/index.cfm
11. OMG. 2003. *MDA (Model Driven Architecture) Guide Version 1.0.1*. www.omg.org/mda.