SPOTLIGHT ON SPARX – AND ENTERPRISE ARCHITECT, ITS NEW GENERATION UML MODELING TOOL

Software systems are inevitably becoming more complex. As designers seek to build more efficient, more powerful and more far reaching systems, the need for tools to visualize, manage and control that complexity increases. With larger teams, the need to share a common language and quickly work from sketched models through to more structured domain and implementation models is essential. TONY STEVENSON speaks to Sam Mancarella, Chief Technical Officer at Sparx Systems, about how UML, and in particular a tool like Enterprise Architect from Sparx Systems, is a great way to get this complexity under control.

International Developer: What's the role of UML in the SDLC (System Development Life Cycle)?

Sam Mancarella: The Unified Modeling Language (UML) has emerged as an industry standard for modeling systems today. One thing to keep clear from the outset is that UML is a modeling “language” and not a process. It defines a clear and well understood dictionary of modeling terms and constructs which allow software and other models to be clearly defined. To paraphrase the OMG, who develop the standard, the language is used to ‘model application structure, behaviour and architecture as well as business processes and data structure.’ As a graphical language, UML provides 13 types of diagrams for describing different aspects of systems throughout the various stages of the SDLC.

ID: What are the advantages of this type of modeling over traditional modeling techniques?

SM: One of the key concerns we have in development today is the need to manage complex software and system projects. The UML facilitates this by providing different abstract views of a system, and a rich, well understood language for communicating each of these views to team members and other players.

For example, Use Case models provide a high-level description of system functionality, whereas Activity diagrams can be used to analyze detailed system behaviour. As a graphical language, UML tends to be fairly intuitive, so it’s easy for new-comers to get the sense of UML models. Given its prevalence across the industry, you’re also likely to reach a wider audience when modeling with UML.

The UML has also matured a lot over the years, so it has strong tool support for automating many tasks throughout the SDLC, such as generating documentation, code stubs and database schemas directly from the model. These productivity boosters can bring a substantial return on investment.

ID: Enterprise Architect is Sparx Systems's own UML modeling tool. What can be achieved using Enterprise Architect?

SM: As a comprehensive UML modeling tool, EA allows you to specify systems from the early stages of requirements gathering, right through to the analysis and design stages, testing, deployment and maintenance. This is important because it also allows you to achieve direct traceability from one stage of the SDLC to the next.

One of the great things about a tool like Enterprise Architect is the extent to which you can manage and control very large projects. When you are dealing with thousands of requirements, classes, components and behavioral specifications, the need to have automated traceability and fast intuitive tools is paramount. Add version control, quality assurance and a large development team to the mix and it is immediately apparent why some of the more simplistic modeling tools begin to suffer badly.

Beyond pure UML modeling, EA provides a rich set of tools for taking advantage of the model, including the ability to forward and reverse engineer code and database schema, create high-quality, customizable HTML and rich-text documentation. Coupled with Model Driven Architecture (MDA) transforms and UML Profile support, Enterprise Architect is a highly extensible platform for modeling and building software systems.

ID: Across a company or organization, who are the people that should be using Enterprise Architect?

SM: Enterprise Architect is definitely targeted at all of the above roles. A typical scenario we observe prior to companies standardizing on EA is that an individual within a department will pick-up the tool for a specific task, say a developer who wants to reverse engineer and document a legacy system, or an analyst who needs to formulate a requirements specification. Then an architect or team-lead discovers the resource, planning and reporting features and so on. Gradually the tool filters through to the rest of the organization as awareness of EA's rich feature-set grows.

What we have been committed to from the start is that EA is a tool for all team members, from the analyst, through designers, developers, testers, project managers, QA teams, deployment team, project managers and finally maintenance. By linking these disparate roles through a single, shared information repository, communication and collaboration are greatly improved.

ID: From a developer perspective, what are the benefits to be gained from using technology such as Enterprise Architect?

SM: From a developer perspective it's all about the code! So developers will naturally take advantage of the code engineering framework to automatically generate code from UML class diagrams. They can also analyze legacy systems by reverse engineering entire code bases into a structured, graphical model. Most importantly, by working from the same model, developers stay on the same page as the rest of the team – with direct traceability between model and code, it's easy to verify that implementation matches design.

By using Model Driven Architecture transforms, it is possible for developers to quickly set up language neutral models and generate platform specific artifacts, such as database models and DDL, code models and generated code. It is even possible to generate WSDL and XSD models/files from very high level (abstract) models.

For jUnit and nUnit enthusiasts, EA also allows tests to be executed from within the modeling environment and results stored back into the model, effectively keeping a record of tests run and their results. It is possible to then use the querying facilities within EA to search for failing tests and to even trace back those failures to use cases, requirements and other parts of the model.

ID: There is a strong emphasis in UML on a visual approach to modeling, and as a result, Enterprise Architect supports a diverse number of diagrams like package diagrams; object diagrams; component diagrams; use case diagrams; activity diagrams; and so on. Why are diagrams like these such an important element of UML and Enterprise Architect?

SM: Each of the diagram types allow different aspects of the system design to be modeled. For example structural diagrams (class diagrams, component diagrams) allow the system's form to be modeled – behavioral diagrams (activity diagrams, sequence diagrams) allow the system's function to be modeled.

In addition to these structural and behavioral aspects, the ability to model the system in different paradigms (conceptual vs concrete) is also possible in UML. For example the Use Case diagram is used to conceptualize the interaction of the system with the outside world. This is in turn, is used to drive concrete design of the system.

ID: What sort of platforms and computing environments is Enterprise Architect best suited too? For example, is it a requirement that an IT department be totally "object oriented" in its approach to development work in order to make full use of Enterprise Architect?

SM: One of the great strengths of EA is the flexibility of the tool. An organization can use as much or as little of the classic "object-oriented" functionality as is appropriate or desired. UML and EA are all about modeling the fundamentals – behavior, process, structure, hardware components, interacting systems, messages, signals, actions and more. Object-oriented programming and design are a part of UML and EA, but they are not pre-requisites. Requirements, use-case modeling, database models, component and deployment models for example may not always fall directly under the traditional "OO" banner.

ID: Can you supply some practical examples of how Enterprise Architect is being used by your clients?

SM: Because of the wide applicability of UML and modeling in general, and Enterprise Architect's wide range of features, we see it being used in an incredible number of ways – in a surprisingly diverse set of environments and industries. We have clients in manufacturing, automotive, aerospace, telecommunications, entertainment, software design, finance, banking, government, hardware and embedded systems, to mention a very few. Many of these will be using EA for full lifecycle system design. Some will be using it primarily for requirements and business modeling, others for code generation and visualization of their existing code base.

ID: The latest version of Enterprise Architect has recently been released? What new features have been incorporated into that release?

SM: Version 6.5 of EA introduces support for the latest UML and XMI 2.1 specifications as well as many other enhancements to the core feature-set. This includes a revamped HTML documentation generator that allows users to publish navigable content that mirrors the actual model hierarchy- a great way of keeping clients and stakeholders in the loop.

One of my favorite features in 6.5 is Object Workbench facility, which lets Java and .NET developers create ad hoc instances of their model objects and probe their functionality by invoking methods at will. Coupled with the ability to record the results of this activity in visual form as a sequence diagram, this is a really nice way of exploring and validating the functionality of code as it is being developed.

Also included in 6.5 is a new Pan & Zoom window for easy navigation of complex diagrams and a raft of other enhancements that make this a really exciting release!

ID: If a company or an organization wants to try out the Enterprise Architect UML modeling tool, what do they need to do, and most importantly, what can they expect to achieve?

SM: Sparx provides a downloadable and fully-functional trial of Enterprise Architect from the web-site, which can be used for 30 days without limitation. The trial version comes with the full range of UML constructs and a team-enabled model repository at their disposal. An example model with a lot of descriptive information is also included.

Like many things, the pay off comes from the effort you put in. Immediate benefits may include the ability to visualize code – both static and executing (Java and .NET) code, reverse engineer database models, reverse engineer WSDL and XSD files. Likewise, MDA transforms, code generation, document generation and other easily accessible features are a great productivity boost. However, the longer term benefits come when you think globally about your project and start to tie together all the bits and pieces that traditionally end up in widely separated places. By exposing and connecting all these fragments, the concept of a shared vision and collaborative development become a reality.

ID: In conclusion, what types of training and support from Sparx Systems are available to companies and organizations wanting to incorporate Enterprise Architect into their SDLC?

SM: The Sparx Systems Partners Program has attracted over 250 firms around the globe in providing sales, training and consultancy services to new and existing users; many of our partners provide services that cater for product training, but also can assist in a consultancy basis to ensure the success of a project.