

Application Development Trends

DEVELOPMENT SOLUTIONS FOR CORPORATE SOFTWARE MANAGERS

WWW.ADTMAG.COM

Application Integration & Management

ACNielsen, Kesmai Studios and Loewe GmbH look to UML to bring order to complex development projects. While the technology is a big help, there is still a way to go.



BY DEBORAH MELEWSKI

the Unified Modeling Language (UML), created with great fanfare by three of the world's top methodologists, was adopted as an Object Management Group standard in 1997. By that time, UML had gained the support of virtually all development tool makers,

ranging from its creator, Rational Software Corp., to industry leaders like IBM and Microsoft Corp., as well as a slew of smaller suppliers.

The next step was to convince corporate development managers and their teams that the time had come to bring order to development projects that were becoming more complex as the World Wide Web was added to the mix. In many cases, development managers have seen the light. This story looks at how corporate development groups

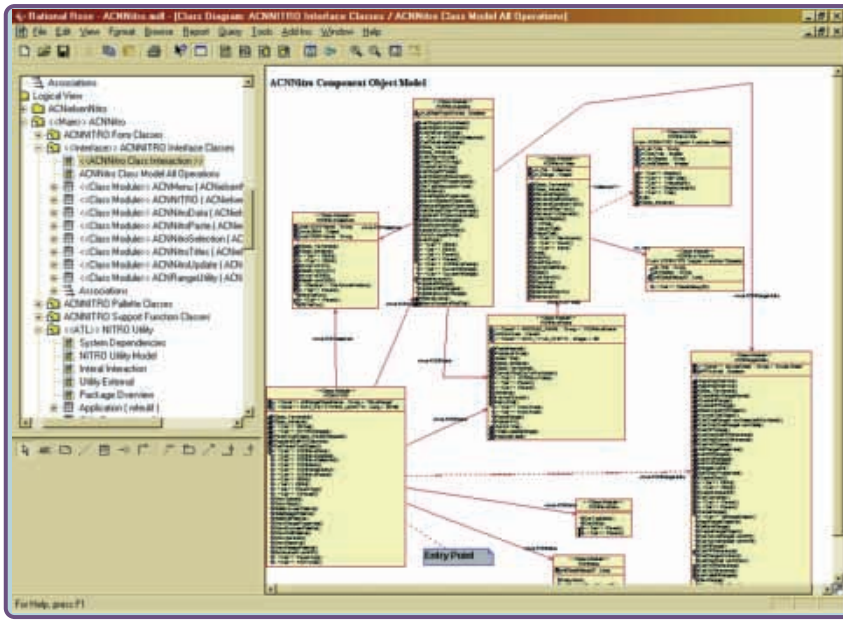
at ACNielsen, Kesmai Studios and Loewe GmbH have turned to the UML for strategic new projects.

At marketing research company ACNielsen, Schaumburg, Ill., all new software development projects begin with UML and the Rational Rose object modeling and design tool from Rational Software Corp., Cupertino, Calif.

In fact, ACNielsen Senior System Architect Joe Canon will not even OK the start of a project until the use cases have begun for its require-

ments portion. "Object modeling helps us design better systems, while also making them less error-prone and much more maintainable," said Canon. "The quality and stability of our software is of the utmost importance, and a problem with the software is unacceptable," he explained. "When you develop syndicated software and you ship 30,000 CDs, you don't want to be saying 'oops' afterwards."

ACNielsen, with 1999 revenue of \$1.5 billion, develops syndicated,



This screen capture from ACNielsen shows how the company did its design using the Unified Modeling Language (UML) and Rational Software's Rational Rose product.

in-house and custom applications software for measurement and analysis of marketplace dynamics and consumer behavior. Clients include leading consumer product manufacturers, retailers and service firms, media and entertainment companies, and the Internet community. The company's syndicated OLAP tools have migrated through the years all the way from main-frame processes to DOS, to Windows to 32-bit componentware.

Now in its fourth year, the UML has documented success stories in both large and small development projects. While the UML has been criticized by some in the industry for its complexities and size, these are non-issues at ACNielsen. More than 100 developers are working on various U.S. projects within the company, and many of the UML's components have served a significant purpose within the company.

At ACNielsen, the UML's use cases provide a means of identifying and clarifying requirements in the functional areas. Use cases have also been beneficial in serving as a common language between the company's business analysts, development teams, designers, domain experts and marketing personnel. As the team goes further into identifying how repetitive certain operations might be, and how often things may be used, use cases have also helped to determine what some system requirements might be. "How a system is used is

sometimes the best way to identify your technical needs," said Canon.

Following the use cases, the team moves through the Analysis Model to do a logical analysis of what needs to be done to fulfill the use case functionality. "The Rational Rose tool helps us a lot there because it's easy to use with its drag-and-drop capabilities," Canon said. "The tool also has a simple F5 [command] to create your collaboration diagram, which saves a lot of time."

From there, the team takes the sequence diagrams that were completed in the analysis phase and uses them as the framework for the detailed design sequence diagrams. "Doing the collaboration diagrams along with the sequence diagrams also helps [in] mapping to the use cases," said Canon.

"Collaborations are good because things become time-independent ... giving you a straight feel for the different object interactions," he added. "This helps you with normalizing and de-normalizing your object models, and in figuring out which ones should be independent classes and which ones are actually the same objects."

Sequence diagrams define the ACNielsen team's classes, associations, generalizations and stereotypes. Canon also likes the way Rational Rose works with the UML to let him define his own stereotypes. "We fully round-trip engineer," he said. "Once we have a pretty good first cut

of our classes and definitions, we go through the component tool and assign different classes to the different languages necessary, be it C++, Visual Basic or Java."

On the Java front, the company recently started a new project by reverse engineering an existing Java application. During that process the Rational modeling tool was used to identify inefficiencies as well as to polish class interfaces.

"One of the stronger aspects of the UML and modeling is that you can visualize your system," said Canon. "Sometimes when you're looking at piles and piles of code you can lose sight of the overall picture and interactions," he added. "Just going through the code you don't always know how every object is used, so you can very easily fall into the mistake of changing a method or property, or an operation attribute, thereby affecting many things and not knowing about it right away. By doing all of your object manipulations through the models, you have a better chance of catching any mistakes up front."

ACNielsen uses the UML all the way down through the Software Quality Assurance (SQA) process. The company's SQA team creates test scenarios from a project's use cases. "When they find an issue in the system, we can track it down through the design module to see how the use case started; we can check Sequence Diagrams to see how the system is supposed to be flowing, right down to the classes that are implemented. We can debug our software much more quickly because of that connection," said Canon.

The UML has proven beneficial in many projects during the last few years at ACNielsen, with leading OLAP products such as Nitro and NetDispatch backed by UML and object modeling. Even the company's custom applications team has moved to the UML. "As people get over the stigma that object modeling slows you down, and that if you're in a fast-paced RAD development mode it doesn't lend itself to object modeling, they're finding that this isn't the case," noted Canon.

"You can move at a fairly good pace as long as you can keep people focused on the fact that it is an incremental and iterative approach all the way from the beginning, not just in the coding," he added. "The number one thing I watch out for is that we don't fall into an analysis paralysis mode. This is a major pitfall when people think they have to have the full system spec'd out to the most minute detail in

their use cases before moving to the next phase," he concluded.

Designed from the ground up

Another successful UML implementation comes from Kesmai Studios, the Charlottesville, W. Va., unit of interactive entertainment company Electronic Arts, which posted \$1.2 billion in 1999 revenue.

A pioneer in the online game industry, Kesmai Studios' earliest offerings were text-based, role-playing games. In 1987, Kesmai launched the world's first graphic multi-player online combat game, the Air Warrior flight simulator.

During the last two years, Kesmai Studios has been moving from C to C++ as its programming development language. "In that time, we've also seen the benefit of using object-oriented analysis and design in conjunction with C++ to get the most out of our project," said Tim Price, software engineer. "We then started looking at methodologies that would provide as good a design as possible before we started coding."

The code for the current flight simulator had been around for 15 years, with multiple patches showing its age. "We decided it was time to start over, from the ground up, with a real design from the beginning and no assumption that we could use any of the old stuff," said Price.

While Price was working specifically on the new generation of his company's Air Warrior Flight Simulator, other people within Kesmai were looking for tools and methods to help with a 3-D graphics engine project and a new host-to-client communication library.

"We started using the UML, which provided a very expressive way of showing what your design should look like and how things relate to each other," said Price. "From there, we also needed to decide what tool would best help us use the UML."

Kesmai chose the GDPro design and reverse engineering product from Advanced Software Technologies Inc., Littleton, Colo. As it turned out, GDPro was not chosen primarily based on the way it handled the UML. "We looked at Rational Rose and Visio as well," said Price. "They all had their pluses and minuses regarding UML support.

"GDPro won out because it allowed our group to work together on the same systems and see each others' work immediately," said Price, adding that the product's collaboration capabilities are very nice and

did not require any extra administrative work. "That was a big deal, because we would have seven or eight people all working at the same time on this project," he explained. "Each person would need a file for a while, then another file for a while, which looked like it would be a real headache for somebody. GDPro has a centralized server, so we can work on things at the same time with no interference."

In the future, Price is hoping to reuse code from previous projects. "We started doing detailed UML technical designing last summer, and finished that in March of this year. We used class diagrams a lot; that was our bread and butter. The biggest part of what we needed to do was to figure out what pieces make up the system, the database, the flight simulator, the world environment and the graphics," he said.

UML use case diagrams were also used. "We had a huge game design of 800 pages long," said Price. "We needed to figure out what the program needed to be doing, what sort of things we needed to support," he explained. "Later, as we got through more of the class diagrams, we were able to cross-reference completeness of the class diagrams and see if every use case had some correlation."

Sequence diagrams were used to show a flow of control. This showed how the use case would be implemented, and what class and methods would be used to accomplish the task. Price noted that things were identified fairly early on to ensure that there were interfaces between the various major subsystems.

Even with a fairly good process in place, the team learned a great deal from the project. "Nobody in the company had used the UML in-house to this degree before," said Price. "We had hundreds and hundreds of pages of diagrams by the time we were done."

One lesson learned, according to Price, was that the team could have implemented the use cases much earlier. "This would have better identified what we would need later, such as helping to define what the test plan should have, and a bigger correlation about how to schedule what pieces to work on when we go to implementation."

Kesmai had a few very knowledgeable people in-house who had studied the UML and were able to help educate people. It was determined that GDPro was not up to the latest and greatest version of the UML standard, but not many tools are. "GDPro still has some quirks in its

behavior that have nothing to do with the UML, but which can interrupt the way you want to work at times," said Price.

At the same time, GDPro's collaboration capabilities proved to be invaluable, especially so as more people were added to the project. Additionally, GDPro's reporting capabilities enhanced communications and simplified design reviews.

With experience, the team now knows how to take advantage of different parts in the UML to achieve greater rewards in productivity. "This is a huge benefit, because we were able to express and understand the game and the programs to a great degree before we even started programming," said Price. In the end, this made the programming stage much easier for the company.

By way of project status, Kesmai is just starting to define what will be put into the next Air Warrior Game, the WWII version. "We're not quite ready to address the technical parts through the UML as yet, but we will begin that in less than a month," said Price. "We can also do a staged deployment over the Internet, adding more features to the latest version."

The UML in Germany

Loewe GmbH, Kronach, Germany, is a consumer electronics firm that manufactures television sets. Loewe's latest technological advance is a fully integrated Digital TV module that provides users with convenient access to digital programs.

The new TV set had to be developed with an embedded digital TV receiver. From a hardware point of view, this entailed two separate devices: a TV set and a digital receiver set-top box. The major work within the project was the software, however, because it would require the same look and feel of the initial user interface: consistent handling of usage scenarios such as having only one channel list, the same EPG in analog/digital mode, and recording/timer/event bookmarking control.

The low-level drivers were developed in C, while the application was written in C++. The team decided to bring in the UML and the UML-based Real-time Studio from Artisan Software Tools Inc., Portland, Ore.

Two team members had a Booch and OMT background, but most of the members were "OO newbies," according to Michael Zeitler, software engineer at Loewe GmbH. After a two-day course on OO/C++ and a week-long course on the

UML from Artisan Software, the Loewe team began its project.

Requirements engineering began with use cases, class/package diagrams, sequence/collaboration diagrams and state/activity diagrams. In addition, the Artisan tool provided constraints, concurrency and architecture diagrams that are beyond the scope of the UML.

The scope diagram together with the use case diagrams helped the team understand the system by first finding the system boundary and grasping the black-box view of the system. "For more complex systems, we do scope diagrams in conjunction with use case diagrams to show particular aspects of the system, for example, event flows," said Zeitler. Non-functional requirements are collected in the constraints tables, and major system behaviors are detailed in system/subsystem state diagrams.

The Loewe team began by modeling the hardware, which included boards, ports and buses, from a software point of view. This provided software developers with online access to the latest hardware-software interface requirements, and allowed them to begin low-level tasks such as device driver development.

The Real-Time Perspective Mentor, Artisan's online reference tool, was used to

perform requirements capture, define use cases and constraints, develop system scope and execute architectural modeling. The use cases were then created to develop action models and to build complete object sequence diagrams.

The project was completed in January, with some ongoing maintenance activities. A follow-on project has begun. "The UML was a strategic decision we made," noted Zeitler, "rather than to go with other methods such as ROOM.

"The UML's combination of different system views, including static, dynamic and interaction modeling, strengthened our understanding of system analysis and design," he added. "And the use case scenarios, or sequence diagrams, were a natural fit for our understanding of how the system should interact with the user."

The Artisan tool was also a major advantage. In particular, the team found the Artisan Mentor very powerful, and the mapping and inter-dependence diagrams to model elements easy to understand. "The additional diagrams were very useful, and lightened the changeover from our traditional methods. The clear separation of diagrams and dictionary reduces mistakes such as unintentional deletion," said Zeitler.

Too much, too little, just right

In each of these UML spotlights, companies were able to easily combine the language with their chosen object analysis and design tools. Projects have been successfully completed, oftentimes with fewer errors due to the UML approach to modeling. While some people are still concerned about the overall size of the language, it would appear that many companies have been able to successfully choose and implement the pieces they require.

ACNielsen's Canon makes a good point about the comprehensiveness of the UML. "People may say there is too much detail in the UML, until the one time something different comes along and there's no good way of modeling it."

At that point, people will say the language is "weak," and not comprehensive or flexible enough. "While different areas may get into too much detail and try to cover everything, you can choose to use only the parts of the UML that are applicable to your needs," he said.

But Canon would be more frustrated if people said the UML is "good enough." It is when things are considered good enough that they are eventually retired. ●

