The emerging Model Driven Architecture steers clear of the pitfalls that killed its predecessor, CASE. Its backers note that emerging MDA tools do work as promised, clearing a key hurdle in the effort to gain developer support.

MDA:

COVER STORY

Tools for the code generation

MODEL-DRIVEN ARCHITECTURE, or MDA, embodies the conundrum: Is the glass half full or is it half empty? Even though the MDA standard is still evolving, many products claim to be compliant with it and early adopters are developing apps with them.

MDA vendors claim that today's products can generate between 40% and 80% of the completed code for a given app based on models created with UML, and customers and analysts back up those claims. MDA's purported benefits go beyond automatic code generation and the reduction of development costs, but those advantages are longer-term and most have yet to be proven outside of theoretical conversations. They include factors like eventual code and model reuse, and more effective fulfillment of user requirements. One advantage touted by the MDA camp is the ability to swap out underlying technologies — OSs or languages, for example — by simply revamping the platform-specific model and then regenerating the applications.

Still, a split remains between current users of these products — mostly architects who speak UML or another modeling language — and the targeted group of developers who believe they can do a better job of writing apps than any code generator. And it is developers that need to be convinced that these tools can make their work lives more meaningful by allowing them to concentrate on the creative stuff.

Developer distrust

This distrust is not unearned. Developers have heard about magic bullets for years, and earlier attempts (remember CASE?) at this failed because they over-promised

BY JOHANNA AMBROSIO

and under-delivered. But this time around, advocates say, the products are based on an industry standard from the OMG, the originator of CORBA and other technologies that are well-known and well-loved by developers (see the sidebar "MDA: What is in the standard?," p. 32).

Another huge difference: Today's MDA products, while they have some holes and can be expensive, do work.

Another issue has to do with career prospects. Some developers feel that using MDA tools will make their language-specific knowledge and skill sets obsolete. According to some analysts, this is happening anyway.

"I expect that the success of MDA will follow the success of outsourcing," said Tim Sloane, director of business modeling at the Aberdeen Group in Boston. "I don't like the concept of IT jobs going offshore, but I'm hard-pressed to see how we're going to stop it. I honestly believe that the IT industry is more dysfunctional than the auto industry [was when it] faced restructuring and everything went to Taiwan."

In Sloane's view, software failures cost billions of dollars each year. And more importantly, the apps that result often bear little resemblance to the functionality business users originally requested from their development organizations.

"I ask IT managers whether they have a process that takes input from users about the success of a project — and most tell me 'no,'" said Sloane.

A forced march

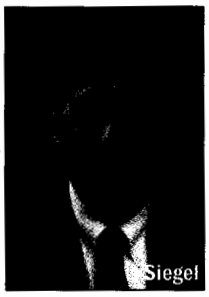
Given the potential savings and the linkage to requirements that MDA promises, many analysts say it is only a matter of time before MDA-like environments will be mandated by management. Michael Blechar, vice president and research director at Gartner Inc. in Stamford, Conn., calls it a "forced march." Right, wrong or indifferent, he said, "it's going to happen. It's effectively a done deal."

Still, none of this will happen overnight. "We're looking at five years at a minimum," said David Frankel, an independent consultant in Chico, Calif., and author of a book about MDA.

Industry watchers liken the use of MDA by developers to the earlier days of programming. Once upon a time, everything was hand-coded in Assembler. But then third-generation languages (3GLs) happened, and programmers resisted these because they felt — correctly — that they could generate more optimal code than a machine could. But compilers became better, with more efficient algorithms, and hardware became less expensive. As developers learned they could trust them, there was a gradual changeover to the new languages and compilers. Industry watchers predict a similar cycle for MDA tools.

Early adopters include Deutsche Bank, Austrian Railways and the Swedish Parliament. MDA is particularly prevalent with systems integrators — CGI, Lockheed Martin and others have projects ongoing.

Cris Kobryn, chief technologist at MDA vendor Telelogic, Irvine, Calif., said that there has been a lot of interest in MDA from the development organizations in engineering firms. These companies — in the aerospace, telecommunications, automotive and manufacturing sectors — typically long ago adopted computer-assisted manufacturing (CAM) and computer-assisted design (CAD), and are looking to apply similar discipline to their software process.


Siegel

Vendors such as IBM (with Rational) and Borland (with TogetherSoft) have made acquisitions that will allow them to further incorporate MDA into their development tools. IBM and Borland acquired vendors that provide visual or other types of analysis and design tools — the front end of the MDA process — and are partnering with others to help on the code-generation end.

The tools

Currently, there are at least 40 tools that incorporate at least one of the major aspects of MDA: UML-based modeling; transformation between the app's overall design models and the models that are specific to the underlying computing architecture (.NET, EJB and so on); and the generation of code in a specific language.

Jon Siegel, the OMG's vice president of technology transfer, calls MDA adoption in the marketplace "extremely enthusiastic," and said there are another "couple of dozen" products in the works. The OMG does not necessarily know about them all, he noted, because vendors need not license or pay to use OMG standards. Still, he said, MDA adoption has caught on faster than any other OMG standard.

Tools generally fall into two camps. One is the "full-boat" environment that includes UML modeler, transformation engine and code generation. Only a few tools are in this space, among them ArcStyler from Interactive Objects Software GmbH, XDE from IBM Rational and OptimalJ from Compuware Corp.

The other camp, comprising the majority of the tools on the market today, are code generators that take input from existing UML modeling tools — Rational Rose or No Magic Inc.'s MagicDraw UML, for example — and then generate code from those models. Tools here include Codagen Technology Corp.'s Architect and Telelogic's Tau.

Another type of categorization could also be considered. These are the tools that existed before the MDA specification but which are still considered by their makers to be MDA-compliant, as well as "pure-play" MDA vendors committed to matching the official standard.

Vendors in each camp are happy to throw stones.

Richard Hubert, CEO and founder of Interactive Objects in Frieburg, Germany, said most of the tools "generate basic skeletal code from models." In comparison, his firm's ArcStyler provides multiple levels of modeling via pluggable "cartridges" that start with the high-level business model and then generate subsequent levels of abstraction from the UML technical model to the Web services model, and then the code.

"It's tuned modeling that's appropriate for each step of the software development life cycle," Hubert explained.

He called the code generator-only class of tool "sub-standard" because "it's very work-intensive to move back and forth between the modeling tool, generator and coding environment. Developers are used to making quick changes, hitting the compile button, seeing the errors and then going back into the code — all within the same environment," said Hubert.

Vendors on the code generation side of the house point to the expense of buying the full-boat environment and say it means customers will need to adopt an-

MDA: What is in the standard?

Model-Driven Architecture (MDA) is both an OMG standard and a generic way to develop software. The standard is currently a moving target, but it should be set in stone within the next year. The OMG standard incorporates, among others, Unified Modeling Language (UML), the Meta-Object Facility (MOF), the XML Meta-Data Interchange (XMI) and the Common Warehouse Meta-model (CWM).

The two most important elements to making the MDA plan work are UML — the modeling format introduced by Rational Software (now owned by IBM) and later adopted by the OMG as a standard — and MOF, which helps to translate the models into specific code. For its part, XMI is mostly about being able to share models and other development artifacts over the Web.

Both MOF and UML have new versions going through the standards process. The newest versions were due to be voted on by OMG members at the group's June meeting in Paris, and it was widely expected that both would be accepted at that point. From there, it will probably be between nine months and a year before all announced MDA vendors are up to speed with the standard, although some will be on board sooner rather than later.

Among its other features, UML 2.0 improves the support for component-based software, essential in the MDA world. The new version reportedly scales better, too.

Jochen Seemann, group product manager at IBM Rational, explained why UML 2.0 is so crucial for MDA. "It's not enough to draw some lines and boxes; all the tools must understand the reasons behind the lines and boxes." With Version 2.0, both developers and tool vendors will be able to use common semantics through integrated tools. "And UML 2.0 comes with technical defined compliance levels," Seemann said. "In the future, we will be able to precisely say that we fulfill a certain level of compliance."

In addition to adopting V2.0, Rational will be making related announcements this summer that Seemann could not discuss at press time. But one has to do with Rational Rapid Developer, a new product based on technology Rational received when it acquired NeuVis last year. Rapid Developer will allow users to create "large portions" of the application from models, said Alan Brown, director of product strategy at IBM Rational. "It will leap between platform-specific and platform-independent technologies," he said, "so you don't have to have large numbers of developers coming up to speed in the servlet technology in WebLogic," for instance, he explained.

For its part, Compuware is not promising full support for every MDA-related standard right off the bat. "This is a process," said Wim Bast, the Amsterdam-based chief architect for OptimalJ. And even while Compuware is involved in "all aspects" of that process, he said, "we can't push OptimalJ as being standard in all aspects. There will be some changes and additions we'll implement as those standards evolve." There is a new version of OptimalJ due this summer, but a Compuware spokesman would not talk about specific features that would be part of that release.

In the meantime, though, Bast said MDA is helping UML to become more "real" to developers. "In the past you'd have this nice picture to help communicate with other people," he said. "Then the 'real' work of coding had to be done. MDA makes modeling concrete — UML is becoming a programming language again, something that really executes."

Still, until all the best-laid MDA plans become reality, Otis Kobryn, chief technologist at Teleologic, cautioned users about different levels of standard compliance. "MDA is a wonderful conceptual architecture, and it's evolving in terms of the standards that are part of this architecture." To his way of thinking, UML 2.0 is "essential" to fulfill the promise of MDA. "If the modeling language has limitations, it's not complete enough or precise enough to specify everything" needed to make the vision of MDA truly viable, he said.

MDA is already in danger of going off the hype-meter, Kobryn warned. He likened MDA today to object-oriented technology when it became hot, and then "everything became an object. There's a point where good ideas become diluted. We need to be on solid ground with a more mature standard and more mature tools."

That is what the OMG is hoping to do, said Jon Siegel, vice president of technology transfer at the organization. "By putting this into the spec for the MDA, we're giving the industry a direction to follow — so all tools that implement MDA will work in the same way."

For more specifics about the standard and information on which products their makers already consider to be MDA-compliant, check out the OMG Web site at www.omg.org/mda.

— Johanna Ambrosio

other UML modeling tool instead of the one they already use. Working within your existing modeling environment also reduces the learning curve for MDA in general, assuming that everyone on the project is up to speed in UML (which is, admittedly, a big assumption).

In addition, customers want fast payback on their MDA tools, pointed out Michel Brassard, chief technology officer and founder of Codagen in Montreal.

"Our customers said they didn't want to develop a model, develop another model, develop another model and then generate code," said Brassard. "They wanted to do a model, then do the code. For commercial applications, ROI is key."

The holes

None of the tools is perfect; all have some holes because it is still early going in MDA-land. For example, there are certain kinds of apps that have UML profiles completed, said consultant Frankel, including real-time embedded systems and certain types of distributed systems. But there are many others that do not yet have specific UML profiles, and these need to be complete to be able to do widespread modeling of a wide variety of apps.

"This is a huge part of the maturation process," Frankel explained. "But there are plenty of applications that can be generated today — this is the low-hanging fruit — and part of being an early adopter is knowing where to use MDA successfully."

Another issue, Frankel said, is that it can be hard to keep all the various models in sync with any hand-finishing work that is later done on the code. "If you don't have management procedures in place, you can get out of sync," he said.

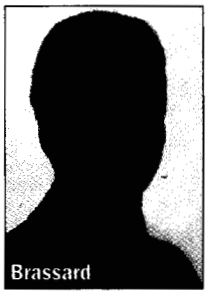
The real world

But customers are finding their way. AIM Investments, based in Denver, is an 11,500-employee company that owns Invesco and

CONTINUED ON PAGE 34

CONTINUED FROM PAGE 32

other financial concerns. Wes Williams, a software architect at AIM, said he has been using MDA "for just about everything I do." His current project is an integration effort that includes drawing UML domain models, applying J2EE patterns and then applying design patterns on top of that. He is using IBM Rational's XDE, among other tools.


Brassard

"UML will never generate the functional code," Williams said, "but it does generate the architectural shell" that he then hands over to the coders on his team. When they are done, he reverse-engineers what they have done and then adjusts the model to reflect their changes or additions.

He likes the approach because "when we do generate code, it's more likely to be correct. And as an architect, I can ensure that the code is being developed according to good quality principles." In addition, Williams feels that it shortens the overall development time (although he has no metrics on the specifics of that).

Williams estimates that MDA generates about 50% to 60% of the application code, but it depends on the application.

Success with MDA depends a great deal on the coder's fluency in UML. And it is "like learning a foreign language," Williams said. "It's one thing to know the vocabulary; it's another to think in the language and speak it fluently."

The second is what developers need to be successful with MDA. "We require class diagrams and collaboration diagrams" from coders, he explained. "It's painful at first, because they're learning to think and speak in a new language. But when that happens, you get the benefits of the model-driven architecture."

Gartner's Blechar estimates that 15% to 20% of organizations are using integrated model-driven development. These are the ones using models throughout the development life cycle as opposed to those that use models for requirements analysis, for example, and then put aside the models once the "serious" coding happens.

Organizations in this 20% group are most likely to move to MDA the fastest, as they already have the people and processes to understand modeling in place.

"Unfortunately," Blechar said, "the majority of companies don't have these people. A lot of companies will assume the everyday developer will know how to design reusable components."

But, he said, that has not traditionally been a programming discipline. Reusability is "about pattern recognition, resolving disputes over politics and cultures, and managing versions and configurations of multiple releases. It requires teaming in ways that programmers aren't used to."

Many programmers today are focused on re-assembling new services or apps from existing components, he noted, or they are doing integration projects.

Although the logical conclusion of MDA — if it is implemented fully — will mean vast changes to a development organization, there is no need to start there. Aberdeen's Sloane suggested that a more intuitive place to begin might be for an architect to use MDA tools to design models and generate the business logic and technical architecture models, and to then generate the stubs of the source code.

Coders are then handed this, and can modify or add to the source code as needed. The code is checked against the models, which validate whether the code has lived up to the architectural guidelines.

Bob Carasik, an enterprise architect at Wells Fargo in San Francisco, said that going with MDA is "a matter of the project leadership making the decision to do so, and then making the tools available." Wells Fargo is using MDA to convert interfaces written in CORBA to interfaces written in J2EE with XML messaging. Almost all the bank's apps use these interfaces, including mainframes, client apps that run various online apps and the corporate telephone response center.

In the bank's case, in-house developers handcrafted the tools used to switch from CORBA middleware. "It has paid off because we have so many interfaces to convert," Carasik said. And "it wasn't necessarily a case of using MDA — it was more about how do we convert the interfaces, and what are the appropriate tools.

"You don't change the development cycle overnight. You introduce concepts one at a time, with a clear rationale of why people are asked to do things," Carasik said. "I'd tell developers to go for it because there's increasing demand for these skills, and these techniques are needed." •

Johanna Ambrosio is a freelance writer in Marlborough, Mass. She can be reached at jambrosio@earthlink.net.

IBM Rational unveils RAD tool

Like others, IBM's Rational Software Division is returning to a concept from an earlier era — the Rapid Application Development, or RAD, tool. IBM is categorizing Rational Rapid Developer as an "Architected Rapid App Development" environment (ARAD). The visual, model-driven development environment is aimed at making life easier for developers working on J2EE business apps, according to IBM reps.

The new ARAD tool targets enterprise developers, enabling them to build apps that have a strong underlying architecture, reps said, and making it easier for them to build high-quality, standards-based apps that can scale.

Forrester analyst Liz Barnett sees the tool as part of a trend to give programmers model-based development environments. "The focus today is on model-based development, which has the potential to provide some real productivity gains," she said.

Time pressures and budgetary concerns have been driving software development toward tools that emphasize productivity, Barnett noted. "Managing costs and timing are absolutely critical today," she said. "Tools like this are addressing those issues in a direct way by automating the development of a lot of code."

Widely seen as the first fruits of IBM's purchase of Rational Software, the ARAD tool has been in development since before the acquisition, initially by Shelton, Conn.-based NeuVis Inc. Rational acquired NeuVis in 2002.

"This [tool] has been in the works for a long time," said Forrester's Barnett, "both at NeuVis and at Rational. And it has been in the hands of beta testers for a while. I think it's safe to say that it has been fully baked."

IBM is pricing its new Rational Rapid Developer tool at $5,995 (U.S.) per user license.

— John K. Waters