

PRODUCT REVIEW

Reviewed by Donald G. Firesmith

ObjectMaker 2.0: the power of choice, customization, and extensibility

During the 1970s and early 80s, functional decomposition software development methods evolved slowly and computer-aided software engineering (CASE) tools could survive and even prosper for years supporting only one or two of the most popular methods. And what you saw was all you got; customization was not available. But during the last ten years, a great number of object-oriented development methods have been developed and are rapidly evolving. With so many new entries in the race, there is as yet no clear winner, and there may not be one for several years to come. To avoid obsolescence and compete in the entire market, CASE tools must support all of the major object-oriented methods, and CASE vendors must update their tools every three to six months as the methodologists advance the field.

With most CASE vendors coming from the COBOL and MIS/ADP community, support for today's numerous object-oriented methods is either minimal or lacking altogether. Such tools often only "help" developers obtain relatively poor or inappropriate designs more efficiently. "Object-oriented" seems merely a buzzword when used by CASE vendor salesmen who know that it increases sales, even if they do not quite understand what it means or how their tools do (or more likely, do not) support OOD. Because methods should drive tools (and not the other way around), developers should first determine the best OOD approach for their application and organiza-

Donald G. Firesmith is President of Advanced Software Technology Specialists. He is developer of the ASTS development method 2, a full-cycle, object-oriented software development method for large, real-time embedded projects.

tion and only then choose the tool that best supports the approach.

This is where ObjectMaker 2.0 comes in. It is currently unique. No other CASE tool provides as much choice when it comes to different state-of-the-art object-oriented development methods. These methods are supported today and are consistently mapped into a single repository schema, allowing developers to mix and match graphics from different methods without having to worry about inconsistencies in terminology among the different methodologists. ObjectMaker also allows the sophisticated user to customize features (such as pull-down menus and rule checking) and to extend the tool to support project- or company-specific or proprietary methods and rules.

I first evaluated Mark V System's ObjectMaker several years ago and was pleasantly surprised at its ability to support several methods. Although older versions of ObjectMaker were somewhat buggy and limited in capabilities, I have since observed ObjectMaker evolve into an excellent production quality tool. During this time, I have used it to develop support for over ten methods, as well as used it on several customer projects.

PRODUCT DESCRIPTION

ObjectMaker is a graphics-oriented CASE and meta-CASE tool developed and marketed by Mark V Systems Ltd, 16400 Ventura Blvd, Encino, CA 91436, 800.666.6232. It supports over twenty object-oriented and functional decomposition software development methods covering software requirements analysis, logical and physical design, automatic code generation, and reverse engineering. ObjectMaker is hosted

on numerous platforms and runs under the Microsoft Windows 3.0, X Windows X11R4, and Macintosh windowing environments. Its graphics can be pasted into analysis and design documents using multiple publishing systems. Mark V Systems offers ObjectMaker in three major configurations designed to meet specific project and budget requirements. The typical developer uses the complete (but uncustomizable) ObjectMaker including automatic generation of, and reverse engineering from, either Ada or C and C++. This configuration supports all of the nonproprietary methods and costs \$8,000 for a single seat with significant discounts available for multiple seats. One can add the other language(s) for an additional \$3,000. One can also obtain additional seats of the ObjectMaker configuration for analysis and design work without language support for \$5,000. Finally, for the sophisticated user with special needs, the Tool Development Kit (TDK) provides the critical ability to extend the tool to support project- or company-specific or proprietary methods and rules. The TDK also supports all of the standard graphical notations. The TDK costs \$25,000 (a bargain compared with the cost of developing even a fraction of the functionality in-house) and requires a three-day training course on the rule-based ObjectMaker Extension Language.

Each class of diagrams of an individual method is supported through the use of a menu file, one or more rule files, and part of a view file written in Mark V Systems proprietary rule-based language. With adequate training, owners of the Tool Development Kit can use this language to extend ObjectMaker by adding new graphics, tailoring pull-down menus, adding or modifying rules (eg, arc checking), and

.....

extending the tool to new graphics and methods.

OVERALL LOOK AND FEEL

ObjectMaker uses a windowing environment based on pull-down menus. When creating a new diagram, one first chooses the method and then the class of diagrams. The "icons" pull-down menu is then used to choose what diagram-specific nodes and arcs to draw. The other pull-down menus (eg, actions such as move, copy, delete, etc) are relatively straightforward and standard from diagram to diagram. I rate user friendliness in terms of how little (or how much) of the documentation one must read in order to do useful work. ObjectMaker passes this test, and most developers familiar with CASE tools and the Macintosh or Microsoft Windows will probably be able to use most of the tool with minimal reliance on written documentation. One minor limitation is the PC limit of eight-character file names, which can be annoying when trying to find the right method and diagram class.

PRODUCT CAPABILITIES AND LIMITATIONS

ObjectMaker version 2.0 has significant advantages over its competition that make it a unique CASE tool. The most important of these is the class of individual methods (and associated diagrams) that it supports today. Because ObjectMaker supports both object-oriented and functional decomposition methods, end users can move to more modern methods using the very same tool. This primary advantage, however, can be a double-edged sword. With so many methods to choose from and so many ways to mix and match graphics, the less sophisticated user may be at a loss at where to begin. Such power of choice demands better understanding on the part of the user if the full power of the tool is to be achieved.

Because of the great extensibility built into the tool, relatively little time is required for a methodologist experienced with ObjectMaker's rule language to add the capability to draw a new diagram and to check arc consistency and node parenting. I was

able to add Chen diagrams (a simple diagram with few rules) in approximately 45 minutes, and it only took two weeks to add initial versions of nine methods to the tool with more time required for testing than development. Because of Mark V System's association with multiple methodologists and the relatively small effort required to add new methods and notations, Mark V Systems has been able to offer new methods rapidly. This is a major advance over most CASE vendors who often have little insight into methods research and therefore wait until a method is well entrenched before investing the effort to support the method. Mark V Systems has even allowed methodologists to write and maintain the files to support their own methods, ensuring that the method is supported as intended by the methodologist and also ensuring rapid support of new versions of methods.

Sophisticated users rarely use an entire method or use the method exactly as recommended by the methodologist. Methods need to be tailored for the application domain of the project as well as the corporate culture of the development team. Often project methodologists and lead software engineers would prefer to mix and match methods and notations to form a project- or corporate-specific method and notation set. Because Mark V Systems hires methodologists as consultants to ensure that ObjectMaker uses a standard mapping from the individual methods to a single semantic database, project data is consistently mapped to the database across method boundaries regardless of differences in method-specific nomenclature. If none of the currently supported notations suffice, users can even customize ObjectMaker to develop their own diagrams from its relatively rich set of primitive icons and rules.

In addition to graphics, ObjectMaker also supports (via Adagen for Ada and Cgen for C and C++) automatic PDL and code generation as well as reverse engineering (ie, automatic regeneration of graphics from modified code). These features were origi-

nally based on the diagrams and only implemented for some of the more important notations and methods (eg, Buhr structure graphs, Booch's 1991 object-oriented design). Now that the semantic database is stable, work is underway to support automatic code generation and reverse engineering for all graphics from the database.

In general, ObjectMaker is quite user friendly. Until recently, I was restricted to using relatively simple drawing tools. Even with version 1.8 of ObjectMaker, I found that I could draw and maintain diagrams at least three to four times faster than before. In fact, I will only draw such graphics by hand if I cannot get to my workstation or if I am using a blackboard to communicate with more software engineers than can fit around my workstation.

ObjectMaker (and Adagen, its earlier incarnation) has been in use since 1986, so it has gone through its initial debugging stage.

Like most CASE vendors, ObjectMaker's greatest current weakness is its minimal support for automatic deliverable documentation generation. They do offer a third-party template for DOD-STD-2167A's primarily functional deliverables, but no support for object-oriented documentation. One must manually insert individual diagrams (in various formats, such as Interleaf) into deliverable documentation and write one's own software to access the database for textual information. Mark V Systems recognizes this as a critical limitation to increased sales of their tool, and support for object-oriented documentation is currently planned.

User documentation is reasonable by industry standards, although ObjectMaker is so user friendly that I found I rarely used it. For the small percentage of users who will customize the tool, however, information on the rule language is sketchy, and the required training course is a good idea. Although the error messages are currently being rewritten, some may be too cryptic for beginning users.

Version 2.0 of ObjectMaker includes the code for locking and unlocking the repository to support multiple, interactive de-

velopers. The upgrade and test of the network is underway, and Mark V Systems is doing a native port to PCTE in September for demo at OOPSLA '91. A port to Softbench and Cohesion is also planned. Therefore, support for multiple users from the same database may be available by the time this review is published.

Because of ObjectMaker's evolutionary history, support for graphics is more advanced than support for textual information, PDL, and code. Mark V Systems has plans to integrate ObjectMaker with one or more language-sensitive editors by the end of the year. While it handles simple graphics consisting of icons, arcs, and labels very well, compound icons storing lists of information are less well integrated into the database. I am looking forward to the

time when CASE tools provide a more integrated approach based on the semantic database where the choice of text, graphic, PDL, or code becomes merely a question of preferred media. The foundation appears there in Objectmaker, and it is merely a matter of time before ObjectMaker achieves this goal.

Finally, Mark V Systems is client driven and has provided good technical support during the last year.

CONCLUSION

ObjectMaker 2.0 is an excellent CASE tool for modern software engineering. No other CASE tool provides users with nearly as much choice when it comes to object-oriented methods, customization of features, and extensibility.* It is wisely based upon

the truism that "methods should drive tools and not the other way around" and allows the sophisticated developer to mix and match methods and notations to meet project- and company-specific needs. Its limitations are few, being rapidly eliminated, and insignificant when compared with its strengths. In conclusion, ObjectMaker is my tool of choice. ☺

* Note added in proof: An impressive new tool, Paradigm Plus, was introduced by Prot.Soft at the OOPSLA '91 conference and will be reviewed in a future issue.

OBJECT REPRINTS

magazine

Reprints of any article in OBJECT magazine are available from Reprint Management Services™, the exclusive reprint agent.

High-quality article reprints can help your company in many ways:

- Increased **EXPOSURE** for your product or service
- Credible, believable information that consumers **TRUST**
- Great **SALES** tool for trade shows, mailings and media kits
- Powerful **EDUCATIONAL RESOURCE** for consumers and employees

We tailor the reprint layout and design to your needs.
Call Duane Dagen Today For Additional Information!

**REPRINT
MANAGEMENT
SERVICES™**

Duane Dagen
Reprint Operations Specialist
(717) 560-2001

505 East Airport Road, Box 5363, Lancaster, PA 17601. Phone: (717) 560-2001 • Fax: (717) 560-2063