

Preface

The focus of this issue is AD/Cycle™, the new application development (AD) capability for the IBM Systems Application Architecture™ (SAA™). The papers describe the AD/Cycle architecture, tools strategy, Repository Services, end-user interface services, work management process model and its use, and some of the tools that will be available through the AD/Cycle platform. We are indebted to F. J. Terrio of the IBM Santa Teresa Laboratory in San Jose, California, for his contributions to the planning and development of this issue, and to B. F. Meyers, also of the Santa Teresa Laboratory, for her extensive contributions to the technical review and factual coordination of the papers.

Last September, IBM and its Business Partners announced AD/Cycle as a new approach, framework, and set of products for application development. Many other companies also announced their intention to produce tools for the AD/Cycle environment. AD/Cycle is designed to support the development of application software and to help reduce the backlog of needed applications through the use of advanced automation techniques. This vital area has most recently been the target of systems and tools that are known generically as computer-aided software engineering, or CASE. AD/Cycle is of interest to the CASE community because it advocates and addresses the use of a common framework across the software development life cycle, including support for the management of the life cycle itself. Much of this approach has, of course, been anticipated by earlier work, including the paper on business systems planning by J. A. Zachman in the *IBM Systems Journal*, Volume 21, Number 1, 1982, and the paper on automating the software development process by G. F. Hoffnagle and W. E. Beregi in the *IBM Systems Journal*, Volume 24, Number 2, 1985.

In the first paper of this issue, Mercurio, Meyers, Nisbet, and Radin reveal the approach and framework that underlie the actual products in AD/Cycle. In so doing, they set the stage for the rest of the papers, while providing the historical motivation, goals, technologies, architecture, components, and the connection to SAA. Special attention is paid to the aspects of AD/Cycle that support management of the software life cycle and integrate both the data and functional aspects of software development tools.

Management of the software development process throughout the life cycle is critical to the production of quality software. To accomplish that management, there must be a methodological approach to the development of software, a process that embodies that approach, tools that support that process, and people who control it. The process involves so many events, pieces, tools, and people that automation is required to ensure its success. In their paper, Chroust, Goldmann, and Gschwandtner show how such automation is accomplished in AD/Cycle through process models, work management, and the automated support of both.

The concepts and technology behind the Repository Manager™ are the primary subjects of Sagawa's paper. The extended three-schema approach to tools support—conceptual view, storage view, and logical view—is presented during both specification and execution of tools. He further describes how the Repository Manager provides services and management facilities for entity-relationship and object concepts of data and function.

Matthews and McGee build on the previous paper to provide a suggested approach to modeling the

data that is useful to software development in particular. They demonstrate how such a model can be formulated and suggest criteria that such a model should meet, including ease of use, robustness, and extensibility. Most of the paper is on the options available for representing software development data in this new environment.

Turning to the user aspects of AD/Cycle, Artim, Hary, and Spickhoff present the objectives, posture, and services provided on the user interface and at the user's workplace. These capabilities are supported through extensions and refinements of the context and constructs in the SAA Common User Access (CUA) rules and guidelines. The paradigm present on the interface is that of actions on objects. The paper provides a description of the look and feel of the workplace under that paradigm, along with a view of the changes necessary to evolve tools so that they fit into and present that look and feel for the user.

Hein states the frustration of the user community with the long lead times for application development and the difficulties faced by companies in attempting to coordinate and control the dispersal of information across many applications and user groups. He describes the use of DevelopMate™ on AD/Cycle to provide the necessary centralized understanding of the enterprise model and its data, without which the applications cannot share data effectively. The steps required to implement this approach are discussed, and the resulting benefits for users and for organizations are described. In terms of the focus of this paper, the primary benefit is the foreshortened application development cycle and the likewise foreshortened maintenance efforts.

The participation of Cross System Product/Application Development (CSP/AD) in AD/Cycle brings the professional software developer into the environment through the support of such standard methods as structured analysis, structured design, and structured programming. Dewell's paper shows how application design is performed under CSP/AD in all of its facets: data design, application program design, and overall design of the application structure. Recommendations are given for the best design choices in each category for the AD/Cycle environment.

The last paper on AD/Cycle describes the provision of knowledge-based technology in this environment. The integration of knowledge-based systems into

AD/Cycle allows them to continue their evolution to full participation in the mainstream development of software applications. Many of the integrating features of AD/Cycle set the stage for this participation, and it comes at a moment when knowledge-based technology is poised to take advantage of such integrated environments. Hembry introduces the technology and its history, discusses the process that surrounds knowledge-based application development, and shows how it fits within SAA and AD/Cycle.

On another subject, Faulk and Gustavson give probabilistic methods and experimental results from their work on segmenting continuous speech into lexical units. This step in the analysis of speech is essential to the recognition of words, roots, stems, and endings. The experiments dealt with English, French, German, and Russian. This work forms one part of the necessary chain that would link humans and computers in natural-language discourse.

The next issue of the *Journal* will be a special issue on the coordinated use of images and data, with particular attention on the family of IBM products known as ImagePlus™.

Gene F. Hoffnagle
Editor