essential memory are presented. Event partitioning is recommended as the strategy for identifying essential activities, and object partitioning is suggested as the means of structuring essential memory. Once events are specified, essential activities are modeled in terms of the planned responses and results of the system and essential memory accesses. Essential memory, on the other hand, is modeled in terms of objects and the intraobject relationships.

Having introduced the fundamental concepts of essential activities and memory, their representation, and basic strategies for modeling them, the authors offer new insights into systems analysis goals and some good ideas on deriving requirements for large systems.

The authors deal with the actual problem of deriving system essence. In the next four parts, techniques for deriving the essence of an existing system are discussed. The seventh part outlines a strategy for deriving the essence of a new system. The last part contains recommendations for managing the tasks involved in deriving the essence of large systems.

This book offers new insight on the goals of systems analysis and provides many good ideas for deriving the requirements of large systems. Since the book is packed with new terms and assumes a basic knowledge of the tools and techniques used in structured analysis, few practitioners could assimilate the essential systems analysis techniques by reading the book alone. I also feel that the book would not be a good text in a systems analysis course, since the authors give no end-of-chapter exercises or review questions.

Also, the authors appear to have written a monograph documenting their understanding of the subject. Instead of simply defining the concepts, illustrating them with examples, and then showing their application to a case, McMenamin and Palmer defend and justify their ideas of essential systems analysis as better than the analysis techniques oriented toward data flow structuring. Despite these shortcomings, the book makes a definite contribution to systems analysis literature, and anyone teaching a course on the subject should read it.

Donald R. Chand
Computer Information Systems Dept.
Bentley College
Waltham, Massachusetts

The Human Factor: Designing Computer Systems for People

Richard Rubenstien and Harry M. Hersch (Digital Equipment Corp., Burlington, Mass., 1984, 249 pp., $25.00)

Until recently, the application of ergonomics, or human factors, to the design of interactive software has been largely ignored. Numerous publications now discuss a variety of approaches to the problem, from intuitive strategies to theoretical models of human behavior. The authors have attempted to unify these diverse approaches under a philosophy of design for the user. As such, the book is a practical guide to the production of user friendly systems and is aimed at the software designer.

The book summarizes existing approaches briefly and then gives a well-rounded discussion of how to consider human factors in all design stages. The authors review each step of systems development, from information gathering and initial design to final testing in the field, and discuss the human factors implications for each. The book's contents are also summarized in a series of guidelines—a convenient reference for practitioners. The authors use their own experience in designing an electronic bank teller to illustrate the application of these guidelines.

The importance of the mythical conceptual model, which will inevitably be developed by the user, and its control by the designer are emphasized. A small number of highly visible states is recommended for this model. The use of language, including voice I/O, is discussed. The three human interface styles—command-driven, menu-driven, and direct manipulation—are compared, and the importance of consistency is stressed. The choice of hardware is also considered; available I/O devices are described with their advantages and disadvantages. The presentation of information and the use of graphics are also covered. One chapter is devoted to software testing, an area often neglected in computer texts.

The book is well-written and holds the interest of the reader. Some basic knowledge of design methodologies is helpful but not essential, since the authors develop their own design philosophy.

Clearly, the comprehensive coverage of human factors in all stages of software development makes this book worthwhile reading for all involved with the design of human interfaces for computer systems.

Jean Giblin
Voice & Data Systems
Nepean, Ontario
Canada

WANG INSTITUTE OF GRADUATE STUDIES
School of Information Technology
Software Engineering Faculty

Wang Institute provides a unique educational environment that combines the best features of academe and industry. Faculty members teach one graduate course per semester, interact closely with small numbers of graduate students who have industrial experience, supervise production quality software projects, and pursue their individual research interests. Positions are available for full-time faculty members and one-year visitors who have experience in using and teaching the technical and managerial principles of software engineering. Faculty members receive industrial-level salaries and are provided with sufficient equipment and support staff to achieve their professional goals. Each faculty member is expected to maintain a reasonable balance between research in software engineering and exposition of that material in the classroom. Teaching ability and an interest in graduate-level education are essential.

The campus of the School of Information Technology is located near the Boston high-tech and university areas which provide rich environments of academic, cultural, and professional activities. Instructional computing facilities at Wang Institute include a VAX 11/750 (VMS), a VAX 11/780 (UNIX), a Prime 750, a Wang VS100, and separate UNIX and VMS workstations. Apples, IBM compatible, and Hewlett-Packard personal/computer professionals. A growing collection of software tools for software engineering is maintained by the staff of the Wang Institute Software Environment project.

Applicants should submit the names of three references and a resume that indicates professional specialties, teaching experience, and experience in the practice of software development and software management to: Richard E. Fairley, Faculty Chair, Wang Institute of Graduate Studies, Tyng Road, Tyngsboro, Massachusetts 01879.

RATES: $4.50 per line, $45 minimum charge (up to ten lines). Average six typeset words per line, nine lines per column inch. Add $4 for box number. Send copy at least six weeks prior to publication to: Sandra J. Arteaga, Classified Advertising, IEEE SOFTWARE Magazine, 10952 Los Vaqueros Circle, Los Alamitos, CA 90720.

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, IEEE SOFTWARE may reject any advertisement containing any of these phrases or similar ones: "...recent college grad..." "...1-4 years maximum experience..." "...up to 5 years experience..." or "...10 years maximum experience." IEEE SOFTWARE reserves the right to append to any advertisement, without specific notice to the advertiser, "Experience requirements are suggested minimum requirements, not maximums." IEEE SOFTWARE assumes that, since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only.