Book Reviews

Rainer Händel, Manfred N. Huber, Stefan Schröder

ATM Networks. Concepts, Protocols, Applications, Second Edition

Addison–Wesley Publishing Company, Inc., Wokingham, England, 1994, pp. v, 287, ISBN 0–201–42274–3.

This book is written by authors coming from the Public Communication Networks Group of *Siemens* in Munich, where they have been active in technical planning and standardization of Asynchronous Transfer Mode (ATM) for many years. The authors' education in ATM matters has also been fostered by their manifold activities in international standardization bodies (like ITU-T, ETSI, T1), the ATM forum and RACE project. This second edition is a completely revised up-to-date version of the authors' book on ATM based broadband networks featuring a topical survey of what is really happening with Broadband ISDN (B-ISDN). Since ATM is now an established technique, and the term ATM is well known to the public, the authors decided to give the book a new title stressing the presently hot issues in this area: *ATM Networks. Concepts, Protocols, Applications*.

The book is structured in 11 chapters. The first of them introduces the idea of integrated broadband networks and gives a survey of the current situation concerning B-ISDN. Chapter 2 gives an overview of possible broadband services and applications as presented by ITU-T as well as their technical characteristics. As B-ISDN implementations will, according to ITU-T, be based on the ATM, this transfer mode is briefly introduced in Chapter 3, its technical details being discussed in Chapter 4.

Broadband networks based on ATM cell transfer must meet certain performance requirements in order to be accepted by both potential users and network providers. In addition to performance parameters already introduced for existing networks, ATM related performance parameters and measures need to be specified. In Chapter 4, the authors only deal with ATM layer specific network performance.

Chapter 5 deals with the B-ISDN user-network interfaces (UNI) and with ATM based protocols. The first two sections describe the user network interfaces in general, along with their physical layer properties. Functions, codings and procedures for the adjacent ATM layer, ATM adaption layer and higher layer aspects to the user plane are described in the next sections. The last section on operation and maintenance problems of the user-network interfaces concludes Chapter 5.

Chapter 6 deals with details on signaling transfer and signaling applications. The first section discusses the effects on signaling of the proposed phased approach for the introduction of B-ISDN and gives an overview of the possible protocol architectures at the UNI and network-network interfaces (NNI). The next two sections deal with signaling transfer while the last two describe some details of signaling applications.

In Chapter 7 the authors discuss the basic elements of ATM switches and cross-connects. Particularly Section 7.2 deals with the general classification of switching networks.

In Chapter 8 the authors elaborate on the transmission aspects of ATM networks. After a brief overview of network elements, such as ATM multiplexers and cross-connects, transmission systems for ATM cell transport and network synchronization are discussed. The following chapter presents how to integrate existing networks and networks that will probably be implemented before B-ISDN, such as local area networks (LANs) and metropolitan area networks (MANs).

Some specific ATM problems have been covered in Chapter 10, including voice delay and echo in ATM networks, connectionless service provision and tariffing in ATM networks. The last chapter deals with future trends in high speed networks including optical networks and their relationship to ATM.

An overview of the current standardization situation is annexed. The comprehensive reference list will help readers to find more detailed information on available standards, network concepts, products and the like.

In my opinion, the authors have succeeded in keeping the book readable, thus it should be of interest to a broader audience including professionals involved in planning, development, implementation and sale of telecommunication networks and computer equipment, as well as students who wish to understand the principles of ATM as the basis of future high-speed networks. The book is written clearly and comprehensibly; I think that, for some time (at least), it is likely to be the standard introduction to broadband networks.

Željko Popović
Ericsson Nikola Tesla Institute
Zagreb, Croatia

Ravindra K. Ahuja, Thomas L. Magnanti and James B. Orlin

Network Flows: Theory, Algorithms, and Applications

Prentice Hall, Englewood Cliffs, N.J., 1993, pp. xv, 846, ISBN 0–13–617549–X.

In this book, Ravindra K. Ahuja (Indian Institute of Technology), Thomas L. Magnanti (MIT) and James B. Orlin (MIT), three leading researchers in the field of network flows, provide an integrative view of network optimization, concentrating on network flow problems and, in particular, the design and analysis of efficient algorithms. It should be noted that the 1993 Frederick W. Lancaster Prize for the best English language publication in operations research was awarded to the authors for their book.

Network flow problems have provided an important domain for operations research applications, as well as significant theoretical developments, which, among other things, yielded the historical bridge between linear programming, combinatorial optimization and computer science. Several high quality books on network flows have already been written, unfortunately dealing mainly with traditional operations research. The book under review is distinguished by its inclusion of many contemporary algorithms and applications, and the way it categorizes algorithms and highlights ways to design efficient solution procedures.

The book begins with three chapters on applications and some key ideas from applied mathematics, operations research and computer science that underlie the design and analysis of many algorithms. The three fundamental problems of network flows, namely the shortest path problem (Chapters 4 and 5), the maximum flow problem (Chapters 6 to 8) and minimum cost flow problem (Chapters 9 to11) are presented in-depth. The remaining eight chapters cover other network flow related problems: assignments and matchings (Chapter 12), minimum spanning trees (Chapter 13), convex cost flows (Chapter 14), generalized flows (Chapter 15), Lagrangian relaxation and network optimization (Chapter 16), multicommodity flows (Chapter 17), computational testing of algorithms (Chapter 18) and additional applications (Chapter 19).

This book is an excellent study of network flow problems and a rich collection of efficient algorithms with a number of additional features: it integrates many classical results and recent developments, includes a comprehensive and fascinating set of applications in a variety of engineering, management and scientific domains, provides over eight hundred exercises, approximately four hundred references and a rich and well-structured index. Moreover, the writing style of the book is clear and acceptable for readers with limited background in computer science and optimization. On the whole, this is a book which I enjoyed very much and I highly value having it on my shelf. I recommend it to anyone interested in the theory and practice of network optimization.

Mladen Kos
Faculty of Electrical Engineering
and Computing
University of Zagreb
Zagreb, Croatia

Michael Santifaller

TCP/IP and ONC/NFS. Internetworking in a UNIX Environment

Addison–Wesley (Deutschland) GmbH, 1994, pp. xi, 288, ISBN 0–201–42275–1; second edition translated from the German edition: *TCP/IP und ONC/NCF in Theorie und Praxis: UNIX in lokalen Netzen*

As the author stated himself, he got acquainted with TCP/IP and NFS in 1984, so we can look at his book as the offspring of ten years of experience. The text of the book was compiled from a course material prepared for iXOS Software GmbH and could thus be, as a textbook on TCP/IP and NFS, very useful for students, UNIX system administrators, integrators or even managers. Although it is not meant to be a complete reference or replacement of a manual, those who want to improve their understanding of everyday chores done with the help of manuals may find it a valuable accompaniment. Michael Santifaller succeeded in writing a book which is easily readable and understandable to a wide range of readers, and maybe this is the reason why this book has been reprinted in the first edition twice, and is now presented as the second edition.

The main body of the book consists of thirteen chapters. Chapter 1 is the introduction to the concept of protocol, and discusses the connection-oriented and connectionless protocols. It also includes a briefing on the ISO Open System Interconnection Reference Model. Chapter 2 explains the genesis of the TCP/IP architecture, from the beginnings of ARPANET to the influence of various UNIX implementations. Chapters 3 and 4 provide a detailed discussion of the most important protocols that make the TCP/IP suite. Chapter 5 explains Berkeley r-utilities, while Chapter 6 is devoted to TCP/IP administration. The part of the book dealing with TCP/IP concludes with Chapter 7 which elaborates on internetworking and briefly touches network management. Chapter 8 begins the part about ONC/NFS, discussing the origins of this technology as well as its alternatives. Chapter 9 explains ONC protocols, and Chapter 10 covers NFS implementations. Network Information Service (NIS), previously known as Yellow Pages protocol (YP), is described in Chapter 11. Chapter 12 briefly (3 pages only) refers to some performance aspects of the NFS and the file system organization. Chapter 13 addresses the programmer interfaces to TCP/IP and NFS including the socket interface, TLI, XTI and RPC/XDR programming. At the end of the book there is a glossary and a well compiled index. Included also are some contact addresses to which a reader can apply for more referential information. The bibliography mostly consists of the standards and RFCs suggesting that the author has studied the topic to the roots.

Somebody could think one hasn't read the book carefully if he hasn't found any error in it, so there it is: I would expect the second edition to be without such plain errors as is the one in Fig. 3.13 (page 38). The figure representing the urgent-data mechanism is clearly in contradiction with the accompanying text, yet luckily enough, the matter is understandable. Also, I would have given more space to certain issues (network management, for example), but of course this is the author's liberty to choose how much to write on a determined topic. Nevertheless, I could not agree with the author's sentence that "Experience shows that most UNIX networks currently used in industry are too small to justify the installation of network management and routing".

The book is translated to English by Stephen S. Wilson and I have to notice that he excelled in his job. In contrast to some other translations I have had the opportunity to read, this book is fluent and feels like the author had originally written it in English.

Dragutin Vuković
MicroLAB Computer Consulting &
Engineering
Zagreb, Croatia

John J. Barton, Lee R. Nackman

Scientific and Engineering C++. An Introduction with Advanced Techniques and Examples

Addison–Wesley Publishing Company, Reading, Massachusetts, 1995, pp. xv, 671, ISBN 0–201–53393–6.

C++ language has become the most popular object-oriented language. Its widespread is based on popularity of C language and availability of commercial compilers. Although C++ was originally designed for operating systems programming, today it is increasingly used in numerical and scientific applications as well. The fact is that today most of scientific and engineering programs are written in FORTRAN or C, without the benefit of any particular programming terminology. For small programs these languages may be adequate, but as programs grow, the code becomes difficult to understand, revise and improve. The need for object-oriented strategy becomes obvious. According to the authors, the purpose of the book is to teach the reader how to produce better-quality programs, with emphasis on scientific and engineering programs. Its main audience should be engineers and scientists fluent in FORTRAN or C, professional programmers using C or C++ who are looking for a new, systematic discussion of C++ for object-oriented programming, and advanced programmers who are interested in sophisticated C++ programming techniques.

J. J. Barton is a research staff member at IBM Thomas J. Watson Research Center in Hawthorne. His research interest include experimental and theoretical surface physics and chemistry, and programming concepts and technologies. L. R. Nackman is a research staff member and senior manager of Programming Environments and Compilation Systems at the IBM Thomas J. Watson Research Center in Hawthorne. His research interests include applied computational geometry, C++ programming environments, and software design. Both authors write a column for C++ *Report*. Obviously, none of the authors is strictly a system programmer, but are mainly interested in applied programming. This fact has a strong influence on the concept of their book.

The book is divided into three parts: *Getting Started, Expressing Commonality* and *Application and Techniques*, containing 19 chapters. The first part is a quick tour of C++'s most basic concepts and features for those not familiar with the language. The remaining two parts are dealing with the design and programming approach to more complex problems. Instructive source-code examples are included and each chapter ends with useful notes and exercises for a reader. There are also marked hints that warn the reader of common mistakes and problems.

In the introductory chapter the authors try to answer the common questions of object-oriented programming and C++ language. Chapter 2 and 3 (*Basics for FORTRAN Programmers* and *Basics for C Programmers*, respectively) give alternative introductions to C++ for readers who have managed programming in FORTRAN or C. Chapter 2 is a unique and very useful introduction for experienced FORTRAN users. It gives fundamentals of C++ presenting the basic syntax, fundamental types and functions basics. The chapter is illustrated with several comparative codes in FORTRAN and C++, which will help the indeterminate to transit smoothly to C++. Very helpful are sketches which present variables as boxes. The importance of this presentation becomes more then obvious when talking of pointers, arrays and references. Chapter 3 is an equivalent but, naturally, shorter introduction for C users.

The third chapter *Classes* introduces the basics of classes, which provide mechanisms to introduce user-defined types and to define operations for manipulating objects of those types. To illustrate the usefulness of classes, the authors develop two simple classes (`Point` and `Line`) as well as a dynamically variable array of floating point numbers. After the array class has been developed, the class template and the function template are applied to make the class reusable for other object types. The chapter includes also a section on exception handling, a relatively new feature to C++. The chapter ends with a description of classes nesting and with a valuable section on C++ program organization.

The *Functions* chapter digs deeper into functions, starting with a section on declaration and definition. This chapter contains a very helpful section on function arguments, dealing with the problem of side-effects. The difference between call by value and call by reference is thoroughly analyzed. The chapter closes with sections on function overloading and function templates.

The following chapter (*Functions and Classes*) focuses on how member function declarations affect an object's behavior as viewed from the client's perspective. It begins by revisiting the function overloading in the context of member functions. This is followed by a very interesting discussion on object copying and conversions. Then operator functions, assignment, destruction and friend functions are dealt with.

C++ provides mechanisms for memory management, for controlling the use of memory during program execution. These avoid the problem of fixed-size structures which may cause a program failure when memory has been exhausted. The chapter *Object Lifetime and Memory Management* mainly analyzes the difference between static, automatic and dynamic objects. This discussion is illustrated by a very instructive code which displays the scope of a variable. The chapter ends with a section on dangling references and garbage.

The eighth chapter gives a critical comparison of three different approaches to an array handling: (1) classical FORTRAN approach, (2) comprising class approach without information hiding, and (3) real object-oriented approach which includes information hiding.

Part II introduces grouping of classes into categories for the purpose of expressing various kinds of commonality. It starts with the chapter entitled *Expressing Common Behavior*, in which the authors bring in virtual member functions and the concept of an interface category, a group of classes with a common virtual member function declaration. According to the authors, virtual functions are central to object orientation in C++ providing a mechanism for capturing "is-usable-as" commonality. The concept of virtual functions is illustrated through an example code of computer control of electronic instruments.

The code for computer control of electronic instrument is further developed in Chapter 10 which deals with expressing common data representation and function implementations to a-void replication of code. Extension by public inheritance introduced in this chapter is simple, powerful and flexible, but sometimes yields surprising behavior and can violate the information hiding principle. Member function forwarding is more robust and couples program components less then public inheritance but has other drawbacks. Private inheritance introduced at the end of this chapter provides most of the advantages of both approaches.

C++ provides templates to abstract the commonality inherent in concepts like arrays or lists. The role of templates and the interplay of templates interfaces and inheritance is the topic of Chapter 11. The next chapter deals with types, specifications of both the content of the object and what can be done with the object.

Part III uses substantive examples to reinforce the techniques introduced in Part II and to introduce new techniques. In Chapter 13 (*Arrays*) a system of array classes is introduced which should circumvent the inadequacy of built-in arrays for scientific and engineering programming. This is followed by a chapter on pointer classes that automate memory management tasks.

Chapter 15 moves the focus toward examples that are more specific to scientific and engineering programming. The authors compare the organization of the C++ code to that of the well-known LAPACK (Linear Algebra PACKage) library written in FORTRAN. Chapter 16 builds a set of classes that model abstract algebra and uses them to implement classes for dimensional analysis. Chapter 17 illustrates techniques for manipulating functions as objects. These function objects associate state and behavior with function but otherwise act as functions.

The majority of well-tested and documented numerical algorithm libraries are written in FORTRAN and C. Fortunately, C++ can use many of these libraries and this is the topic of the chapter *Using Legacy Libraries* which discusses issues in calling C and FORTRAN functions from C++. The book is closed with a chapter on the data modeling problem.

I find this book unique in many aspects. Its orientation toward real scientific problems narrows the potential audience, although the first part of the book may be read and understood by any non-engineer. Since FORTRAN is the programming language most widely used by scientists and engineers, I believe that the introduction for

FORTRAN users (Chapter 2) will draw their attention and encourage them to break the barriers of classical languages. Unique illustrations that symbolize variables and pointers are very helpful for the reader to understand the matter. But, the second and the third part of the book require a stronger engineering background, especially with respect to the code samples. Examples become more specific and more difficult to follow.

I recommend this book to anyone involved in scientific and engineering programming, especially to FORTRAN users who are in doubt whether they should move to C++. As the authors themselves reveal, this book is not intended to be an introduction; they recommend either Lippman's *C++ Primer* or Horstmann's *Mastering C++* to be used as a complement to their book.

Julijan Šribar
Faculty of Electrical Engineering
and Computing
University of Zagreb
Zagreb, Croatia

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides

Design Patterns. Elements of Reusable Object-Oriented Software

Addison–Wesley Publishing Company, Reading, Massachusetts, 1994, pp. xii, 395, ISBN 0–201–63361–2.

One of the reasons for introducing object-oriented (OO) technology is the possibility to reuse more easily the same code for different tasks. Many different problems can thus be solved in an elegant way without having to rediscover the same design solutions. The term design patterns denotes a group of such solutions where each pattern describes a problem frequently occurring within an environment, along with the core of its solution: a design pattern systematically names, explains, and evaluates an important and recurring design in OO systems. The use of design patterns requires neither unusual language features nor amazing programming tricks. A pattern can be considered to consist of four essential elements: name, problem, solution, and consequences. The name is a handle which can be used to describe a design problem, the problem describes when to use the pattern, the solution describes the elements that make up the design, and the consequences are the results of applying the pattern.

Design patterns help to identify less obvious abstractions and the objects inclosing them. They support creation of object interfaces by identifying their key elements and the kinds of data that get sent across an interface.

As it is already known, the two most common techniques for reusing functionality in OO systems are class inheritance and object composition. With class inheritance the internals of parent classes are often visible to subclasses. As an alternative, in object composition no internal details of objects are visible. Some of the design patterns show the benefits of using object composition.

As the authors point out, the book *Design Patterns. Elements of Reusable Object-Oriented Software* is not an introduction to OO technology; it is rather a guide to OO programming style refinement. The book assumes a basic knowledge of OO programming and builds upon it. The book is structured into two main parts. The first part, consisting of Chapters 1 and 2, describes what design patterns are, and includes a case study that demonstrates how design patterns apply in practice. Specifically, Chapter 1 discusses the base theory of design patterns, explains their classification, and describes solving problems by using them. Design patterns for certain problems, as well as a simple selection thereof are also provided. Chapter 2 shows the use of design patterns in a case study of a document editor design.

The second part of the book encompasses Chapters 3, 4, and 5, and makes its main part. It elaborates on the previous classification along with a catalog of concrete design patterns, each pattern including example source code demonstrating how it may be implemented in standard OO languages like C++ or Smalltalk. It should be noted that all of these patterns are available on the Internet. Chapter 3 discusses *creational* patterns which abstract the instantiation process and help make a system independent of object creation, composition, and representation. The focus of Chapter 4 is on *structural* patterns which show how classes and objects are to be composed in order to form larger structures. Finally, Chapter 5 deals with *behavioral* patterns related to algorithms and the assignment of responsibilities between objects.

The ending Chapter 6 concludes the book with the author's explanations of catalogization of design patterns. A glossary, a guide to the notation, and some basic classes are given in the appendix.

The book can be recommended as a guide for practicing programmers who wish to learn how to include design patterns into their OO software development process, and how to use them for a more efficient problem solving, respectively. I myself will keep the book within reach because it offers plenty of usable recipes for common OO programming problems.

Hrvoje Zoković
Faculty of Electrical Engineering and Computing
University of Zagreb
Zagreb, Croatia

J. F. Koegel Buford, Contr. Ed.

Multimedia Systems

ACM Press-SIGGRAPH Series, New York/ Addison–Wesley Publishing Company, Reading, Massachusetts, 1994, pp. xii, 450, ISBN 0–201–53258–1

S. J. Gibbs, D. C. Tsichritzis

Multimedia Programming

ACM Press, New York/Addison–Wesley Publishing Company, Wokingham, England, 1995, pp. xvi, 323, ISBN 0–201–42282–4.

One of the hottest keywords being used these days in computing and communications is surely multimedia. This intrinsically heterogeneous technology built upon various and frequently very different underlying technologies including system architecture, (digital) sound and video, data storage and transmission, digital signal processing, user interfaces and the like is finding likewise various and frequently different applications ranging from medical, educational, and business ones up to entertainment, to name only some of them. Thus multimedia has definitely entered everyday life becoming quite a commodity and fully deserving to be denoted as "pervasive and persuasive".

Since multimedia occurs to be the blend of areas widely thought of as being definitely distinct among them, a comprehensive elucidation of all of its aspects is highly welcome. The two books to be reviewed in the following are already acclaimed as a valuable introduction to multimedia. They embrace both a thorough and multifaceted overview of multimedia systems (Koegel Bufford's book) as well as an exposition of models and tools necessary for multimedia software production (Gibbs–Tsichritzis' one). Although overlapping in some topics (e.g. explanation of some multimedia hardware fundamentals as well as some characteristic applications), the books are certainly complementary.

Multimedia Systems is an edited book, the editor contributing himself with rough a third of the chapters. The book is organized "based on the widespread recognition that personal communications and computing are converging" and is composed of eighteen chapters structured into five sections. The first or them, consisting of three chapters, is an overview of multimedia systems from various perspectives. In the introductory chapter *Uses of Multimedia Information* Koegel Buford offers the reader the various domains of multimedia systems applications and gives a historical overview as well as envisioned directions of application areas. Chapter 2: *The Convergence of Computers, Communications, and Entertainment Products* by J. S. Porter provides a contemporary account of the major technology areas converging into multimedia systems. It surveys the historical developments recently achieved, unveils contemporary technology trends, and elaborates on multimedia systems from the standpoint of home/office appliances. Ends the overview section Koegel Buford's *Architectures and Issues for Distributed Multimedia Systems* in which a theoretical framework for analysing distributed multimedia systems is introduced. The basic notions of synchronization, orchestration and quality of service (QoS) are given.

Several models for distributed multimedia systems are corroborated.

The second section in its four chapters brings background information on the fundamental concepts of media and time. J. Strawn in *Digital Audio Representation and Processing* addresses the fundamentals of digital audio. Physical aspects of sound, its digital representation and algorithms for data compression, musical sound and related processing techniques, MIDI protocol for music synthesizers, and a short survey of speech recognition and generation are provided. A. C. Luther in two subsequent chapters explains the elements of video. *Video Technology* covers the principles of analog video as the basis for digital video used in multimedia. After introducing a simplified model of an analog video system, its single items are covered. In *Digital Video and Image Compression* the use of compression techniques is motivated and the basics of data compression presented here including the elements for evaluating a compression system, single video compression techniques, standardization of algorithms, and the relevant standards JPEG, ITU-T H.261, MPEG, and DVI. Finally T. D. C. Little's *Time-Based Media Representation and Delivery* elaborates on temporal relationships, an issue typical of multimedia objects. The single addressed topics are synchronisation and orchestration, models of time and multimedia requirements, and support for system timing enforcement in delivering information.

Five chapters of *Multimedia Information Systems* contemplate topics ranging from real-time kernel support to multimedia authoring. In Chapter 8: *Operating System Support for Continuous Media Applications* H. Tokuda elaborates on real-time support from the operating system necessary for continuous media applications extending from multimedia mail, hypermedia, conferencing systems, remote teleprocessing and virtual reality. He also discusses the issue of QoS. The experience gained using the Real-Time Mach microkernel for digital video and audio applications is described as a case study. Chapter 9: *Middleware System Services Architecture* is an overview of the Multimedia System Services architecture jointly presented by Hewlett-Packard, IBM and SunSoft, which defines a standard set of services that can be used by multimedia application developers in a variety of computing environments. Goals and architecture views are introduced, with an emphasis on classes and objects. Follows Koegel Buford with two chapters. *Multimedia Devices, Presentation Services, and the User Interface* discusses architecture and toolkit support for multimedia presentation services. Here the issue is an integration of multimedia and the window system. Client control, device control, temporal coordination and composition, toolkits, and hyperapplications are the single topics covered in the chapter. *Multimedia File Systems and Information Models* surveys current research results in multimedia information systems. Particular file systems and document models are discussed with the emphasis on continuous media support, multimedia and hypermedia information and content-based retrieval of unstructured data. The exposition is corroborated with descriptions of sample systems. Chapter 12: *Multimedia Presentation and Authoring* by D. S. Backer reviews multimedia authoring and presentation systems, software programs supporting creation and delivery for an end user of "interactive experience" ranging from e.g. training courses to virtual reality presentations. Historical evolution, current state of the industry, design paradigms and user interfaces, barriers to widespread use, and research trends are surveyed.

A similar approach for multimedia communications systems, from B-ISDN architecture to groupwork, is used in Section Four. *Multimedia Services over the Public Network: Requirements, Architectures, and Protocols* by P. Sen describes a typical medical application which is analyzed with respect to network requirements and is subsequently used to illustrate a general service architecture for supporting requrements of typical multimedia applications. This architecture is three-layered with the end-user application laying upon advanced call services, which in turn lays upon network access services. A short review of relevant protocol stacks is provided, too. Koegel Buford and R. Brennan address the architectures and abstractions characteristic of files for multimedia interchange in the following Chapter 14. Standard formats are overviewed and compared with respect to composition models used. Special attention is given to real-time features of these formats. In Chapter 15 teleconferencing as a communication paradigm is overviewed by Koegel Buford and

W. L. Hill. Basic requirements are discussed from the standpoint of multimedia practice corroborated with short descriptions of shared application architectures of several operating prototypes. And in Chapter 16: *Multimedia Groupware: Computer and Video Fusion Approach to Open Shared Workspace* H. Ishii and N. Miyake elaborate on the concept of groupware, a shared workspace that supports dynamic collaboration in a workgroup over space and time. The authors discuss some classification issues and detail a prototype multimedia distributed system experimentally used in calligraphy teaching.

Future Directions is the last section comprising two HDTV and knowledge-based multimedia systems. C. A. Poynton in High *Definition Television and Desktop Computing* gives a short tutorial introduction into HDTV issues including standardization ones. Basic definitions of TV characteristic notions are given focusing on viewing problems. J. G. Neal and S. C. Shapiro in *Knowledge-Based Multimedia Computing* consider the possible role of artificial intelligence in multimedia systems through the presentation of an integrated work environment with an intelligent HCI providing communication and presentation in coordinated multiple media/modalities. Such an approach is meant to simplify user interaction and is corroborated with the description of a workstation-based prototype.

Multimedia Systems is certainly a worthy reference book for everyone wishing to make himself acquainted with the plethora of terms and influences as well as technologies and applications of multimedia. The chapters are tutorial in nature and self-contained. As a collection, and supplemented by a substantial index, they offer the general reader a good interdisciplinary survey of the area. Since the authors are established experts in particular subfields of multimedia, this remains the greatest value of the book.

The other book studying aspects of multimedia is *Multimedia Programming* authored by S. J. Gibbs and D. C. Tsichritzis. Along with the fact of elaborating a slightly different subject, and surely a more restricted one, it is better focused with respect to the exposition of the subject matter. Concentrating on multimedia programming issues it endeavors "to formulate an approach to multimedia programming based on well-defined concepts and techniques that tries to anticipate the needs of future applications;... the approach ... emphasizes software engineering idiosyncrasies of particular platforms, particular devices and data formats." The emphasis is on applying the object-oriented philosophy. In fact three promising technologies for multimedia programming are identified: *objects* providing for appropriate encapsulation of various digital media (images, audio, video, graphics, animation, music) along with the specification of respective operations, *environments* giving support in acquiring, processing and presenting digital media by providing high-level interfaces for tasks as device control, buffering, process scheduling, synchronization and data conversion, and *frameworks* as generic software and generic interfaces from which specific environments can be realized.

The overall scope of the book being "[the exploring of] objects, environments, and frameworks for multimedia programming", it is structured into three parts. The first of them is meant as an introduction and consists of three chapters. In *An Introduction to Multimedia* the authors provide the definition of multimedia and supply a short overview of features of multimedia hardware, networking, and software. The next chapter *Media Types* delves on peculiar types of data characteristic of multimedia. Main media types, classified into nontemporal, temporal and other, are thoroughly surveyed, and are subsequently used as elements to be combined and manipulated as a whole, resulting with multimedia types which are in turn supported by multimedia environments. These latter are defined to be a mix of hardware and software implementing the type since they specify and realize the programming interface of the type. Several widely used multimedia environments are described in Chapter 3 including the CD family, DVI, Quick Time, MPC/MME and Director.

The second part, which also consists of three chapters, establishes the framework of the book. The first of them, the short *Object-Oriented Multimedia*, motivates the use of object-oriented technology for multimedia programming, gives a concise survey of its main properties and incorporates it in the multimedia framework. Chapter 5: *A Multimedia Framework* explores a possible framework for multimedia programming, focusing on its structure and organization. Classes and inheritance are used to develop the framework through specialization, thus extending the respective interfaces. The approach taken is top-down, considering media collectively. In Chapter 6 the described framework is employed to build two applications: a "virtual museum" — a virtual environment for exploring a multimedia information space, and "video widgets" — a technique for integrating digital video with user interfaces. These example applications are built from reusable components contained in an appropriate "component kit".

Finally in the third part some examples of multimedia systems are described in the two remaining chapters. *Some Research Problems in Multimedia Programming* tackles certain general problems confronting environment developers, namely composition, synchronisation, interaction and database integration, using framework notions of media objects and components. In the last chapter *Integrated Multimedia Systems* the application of multimedia in providing "virtual offices" is explored. The German Polikom program is shortly overviewed and a component application — the "communication wall" — proposed.

Three appendices ending the book embrace some additional material of interest to the domain of multimedia programming: *A Media Processing Platform* describing the University of Geneva equipment for experiments in application development, *A Media Component Kit* with components used to construct prototypes described in Chapter 6, and *Online Multimedia Information Sources* available through WWW from the University of Geneve server. The book is supplemented with a glossary of acronyms used throughout the book, an updated bibliography and an index.

The book is well-written, within a coherent theoretical framework, and provides a novel programmer's perspective on the domain of multimedia. It is also well-structured; I mostly appreciated the chapter organization with the particular summary introduction. Although the book is intended for (practitioning) programmers, because of its conceptual foundation and its good style and commendable structure, it can be used as a textbook, too.

Finally, it must be noted that the books are published as a joint effort of Addison-Wesley, the renown scientific publisher, and of ACM Press, the publishing branch of the Association for Computing Machinery. This cooperation is to be welcomed since it will be providing high-standard literature in the computing domain. Thus, it is to be hoped that such a collaboration will continue to the benefit of computing and information technology community.

Vlado Glavinić
Faculty of Electrical Engineering
and Computing
University of Zagreb
Zagreb, Croatia

Frank Nanfarra, Tony Uccello, Derek Murphy

The CNC Workbook: An Introduction to Computer Numerical Control

Addison–Wesley Publishing Company, Reading, Massachusetts, 1995, pp. viii, 310, ISBN 0–201–65600–0.

The objective of this workbook is to teach CNC programming. Students and other potential users interested in this subject do not require only instructions on specific machining terms and appropriate procedures — they need an efficient teaching method which introduces learning through practical work. In conventional circumstances such requirement would be very difficult to fulfill. For this purpose the use of manufacturing equipment is extremely inconvenient and expensive.

To teach CNC programming the workbook offers a simulation software which can be run on conventional PCs. So, the workbook teaches about CNC programming and software, preparing students to create CNC programs and test them in a simulated manufacturing environment.

The workbook features eight chapters. It starts with the discussion of the CNC basic concepts, including theoretical and applied aspects. Chapter 1 covers numerical control evolution, basic computer technology and a brief survey of NC applications. Variety of different CNC controllers with many additional options requires a special effort for introducing and maintaining controller standards. Standard programming functions are then used in part program preparation. Part program truly describes all manufacturing process activities. Chapter 2 presents axes and motion nomenclature with fundamentals of CNC milling and CNC turning. Chapter 3 introduces the Programming concepts which comprise CNC code, block format and all relevant geometrical and technological parameters. Chapter 4 describes the simulation software, software installation, user interface, the way of graphical interpretation and interactive programming possibilities. Chapter 5 deals with CNC milling, i.e. programming of the specific milling process activities like canned cycles and tool compensation facilities. Chapter 6 is devoted to CNC turning application illustrating specific rough facing and finishing cycles, threading and other multiple repetitive cycles. Chapter 7 is an introduction to Computer-Aided Design and Computer-Aided Manufacturing. It covers the basic structure of the CAD/CAM system with description of AutoCAD and MASTERCAM software products. Chapter 8 completes the book with some additional exercises, formulas and data tables concerning material and processing technology. This workbook can be considered valuable for a novice who wants to get started in the field of numerical control. The combination of text and software provides a reader with the opportunity to learn and practice at his own pace. In this respect, this book is fully recommended.

Joško Radej
Faculty of Electrical Engineering
and Computing
University of Zagreb
Zagreb, Croatia