The Proposed New *Computing Reviews* Classification Scheme

A Report of the Computing Reviews Category Revision Committee

This report presents a proposed new classification system for *Computing Reviews* (*CR*) which has been developed over the past two years by a committee of 10 consisting of:

Dorothy E. Denning, Purdue University
Jack Minker, University of Maryland
Alice Parker, University of Southern California
Anthony Ralston (Chairman), SUNY at Buffalo
Edwin D. Reilly, Jr., SUNY at Albany
Arnold L. Rosenberg, IBM Corporation
Claude Walston, ITT Corporation
Theodore C. Willoughby, University of Evansville
Jean E. Sammet (Editor-in-Chief, *Computing Reviews*), IBM Corporation
Arthur R. Blum (Executive Editor, *Computing Reviews*) ACM Headquarters

Unless the feedback from this report causes us to make major changes in this scheme, every effort will be made to implement the new system on 1 January 1982.

The current scheme used to classify reviews in *CR* has been in effect, with only quite minor changes, since it was installed in 1964 by Aaron Finerman, then Editor-in-Chief of *CR*. The 1964 scheme was a significant modification of the original *CR* classification scheme installed by John Carr (the first Editor-in-Chief of *CR*) in 1960. It hardly needs saying, given the changes in our discipline in the past 17 years, that the current classification scheme no longer reflects the computer field very well. (On the other hand, the fact that it is usable at all is a tribute to the soundness of the 1964 version.)

But how much does this matter? Reasons can be adduced for and against changing an imperfect scheme to one which (at least) better reflects the current state of the discipline:

For the Change

1. For those *CR* readers who look for reviews in their specific areas of interest, the accuracy of the classification scheme may have a significant effect on the usefulness of *CR*.
2. For those individuals or libraries who use the *CR* classification scheme for their own filing or bibliographic purposes, accuracy is presumably also important.
3. For those who want to do retrospective searches, accuracy is important.

Against the Change

4. For those *CR* readers who browse (or read) from cover to cover, the classification scheme probably does not make much difference.
5. For anyone used to the current scheme or who uses the current scheme (as in 2 above), a major change in the classification scheme could cause significant upset.

Even if, as is likely, you believe that the first three are better arguments than the last two, there is the further point that a modest, cosmetic change might suffice for the first three arguments without having any real impact on the other two. At least, we would argue, despite the dated character of the current classification scheme, it is not totally obvious that major surgery is needed.

History of the Project to Design a New Classification Scheme

With the above arguments in mind, early in 1979 Jean Sammet, then newly appointed Editor-in-Chief of *CR*, appointed a committee to

(a) consider whether a major restructuring of the current classification scheme was desirable (as opposed to tinkering to remove some of the more egregious anachronisms) and

(b) if the conclusion from (a) were positive, to devise a new classification scheme.

Although the members of the committee generally agreed that major surgery was not only desirable but

Editor contact: Anthony Ralston, SUNY Buffalo, Computer Science Dept., 4226 Ridge Lea Road, Amherst, NY 14226.

necessary, the committee decided that a firm decision to go ahead and design a completely new scheme should not be made without input from the readers of *CR*. Consequently, in the July 1979 issue of *CR* a questionnaire was inserted to enable *CR* readers to express their views on the desirability of major change and, if desirable, on the form it should take.[1] The volume of the response to this questionnaire was predictably overwhelming but, of the 123 returns, a huge majority (72 for, 48 neutral, 3 opposed) was not opposed to a complete overhaul of the scheme. With this "mandate," the Committee proceeded to develop the new *CR* classification scheme.

It should be emphasized that the reason for presenting this new scheme in *Communications* at this time is because *it is not final*; it is not locked in concrete but is subject to change and enhancement based on comments from readers of this report. Some indication of the kind of comments we are most interested in receiving is contained in the last section of this introduction. Although our work has been reviewed by a considerable number of people, it is inevitably the work of a relatively small group with, at least in some areas, limited perspectives. Therefore, we solicit and encourage comment, which should be sent by 31 August 1981 to Anthony Ralston, Department of Computer Science, 4226 Ridge Lea Road, Amherst, NY 14226.

Philosophy of the New System

The philosophy which has guided the committee in the development of the new classification system is as follows:

1. The heart of the new classification system, like the old, should be a tree since this is clearly the preferred format for any hierarchical structure which must be linearized for publication purposes.
2. The current *CR* classification tree has only three levels (below the implied root "Computer Science and Technology"). Although by any metric the discipline has grown enormously since 1964, we decided to continue to restrict the new classification tree to three levels in order that the tree for the new scheme should be able to reflect accurately the essential structure of the discipline for some years (at least 10).
3. Since a three-level tree will, in most areas, not allow sufficient discrimination (i.e., the potential would exist for many reviews under a single leaf of the tree) nor would it provide an adequate method for coping with new developments, we have associated with (almost all of) the leaves of the tree a list of *subject descriptors* which refine the subject area denoted by the leaf. It is intended that the lists of subject descriptors be changed periodically (e.g., yearly) to reflect changes in the discipline.
4. Since the development of a new classification scheme is a necessarily iterative process with successive revisions based on comments addressed to the current version, the question arises as to where to start the iteration. One possibility would be with the previous (i.e., the current) system. But the Committee unanimously agreed that this scheme is so badly out of date that it would not be a useful zeroth iteration. Instead we decided to use the *Taxonomy of Computer Science and Engineering*[2] as our starting point because it purports to be a classification of the discipline as of 1980 (also, and if the truth be told, because both the chairman of the Committee and the Editor-in-Chief of *CR* were heavily involved with the development of the *Taxonomy*). We note, however, that the classification scheme presented here, while it bears noticeable resemblance to the *Taxonomy*, is distinctly different from it: sharply so in some areas, less so in others.
5. Other desiderata in the development of the new scheme were to:

(a) make the new scheme like the old where this is plausible.
(b) make the need of retraining of the *CR* editorial staff and *CR* editors and reviewers as painless as possible.
(c) provide additional mechanisms to enable easier use of *CR* without at the same time overly complicating the classification system.
(d) try to provide satisfactory solutions to the useful changes suggested in responses to the May 1976 and July 1979 requests for comments in *CR*.

A Description of the New Classification System

A. *The Tree and Subject Descriptors*

The new classification scheme has two main parts as implied in the previous section:

(1) A *tree* consisting of eleven first level nodes:

Hardware
Computer Systems Organization
Software
Data
Theory of Computation
Mathematics of Computing
Information Systems
Computing Methodologies
Computing Applications
Computing Milieux
General Literature

and one or two more levels under each of these. The set of children of all first and second level nodes begins with a node *General* and ends with a node *Miscellaneous*. To avoid confusion between the new classification scheme and the old, first level nodes in the new scheme will have letter designations (A through K) with numerals used for the second and third levels. Thus, for example,

E.1.3 Complexity Classes.

(2) A set of *subject descriptors* associated with most leaves of the tree (but seldom with the *General* and *Miscellaneous* leaves). These subject descriptors are in-

[1] A previous request to the readers of *CR* in May 1976 asked only how the scheme might be changed, not whether it should be.

[2] *Taxonomy of Computer Science and Engineering*, compiled by the AFIPS Taxonomy Committee, AFIPS Press, 1980.

tended to subdivide the subject area denoted by the leaves into subareas which

—will serve to group reviews usefully but

—may not have such permanent life in the discipline that they would always be wanted in the classification scheme; later developments could cause a subject descriptor to be deleted or subdivided.

While we expect the set of subject descriptors associated with each leaf to change slowly, we do expect them to change. Since the subject descriptor idea is a new approach for *CR*, an estimate of the rate at which the set of subject descriptors will change is pure speculation. Our speculation is that the change from one year to the next will be less than 3 percent.

B. *The General Term List*

One approach to grouping reviews in *CR* was not by subject descriptors but rather according to a set of *general terms* (e.g., *reliability*) which are common to more than one (typically many) areas of the discipline. On reflection the Committee decided that to group reviews in this way would be less useful to most users of *CR* than a grouping by subject descriptor. Nevertheless, the general term idea seemed to us to have merit as a mechanism for classifying material and enabling users of *CR* to search for reviews in a variety of areas. Therefore, part of the new classification scheme is a *General Term List* (see Table 1) which will be used as follows:

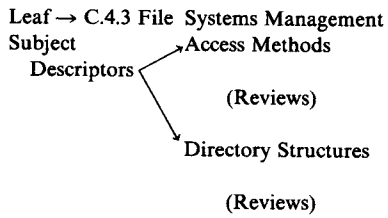
—many *CR* reviews will have a term from this list attached to them and

—the monthly index in *CR* (see below) will contain pointers to reviews corresponding to each general term.

Only experience with this idea can prove whether or not it will really prove useful. But since users of a review journal have a variety of needs which lead them to a variety of ways to look for reviews, we believe that this additional review-locating mechanism will find favor with many *CR* readers.

C. *The Organization of Reviews in* CR

As is currently the case, reviews in *CR* will be ordered according to the order of the nodes in the classification tree (top to bottom, left to right). Under each leaf reviews will be ordered by subject descriptor. Thus, for example,



Each review itself would have a format much like the following:

Author — Review Number
Title — General Term
Journal or Publisher Reference

(Body of review)

Table 1. General Term List

Algorithms	Human Factors	Reliability
Design	Legal Aspects	Security
Documentation	Management	Standardization
Economics	Measurement	Theory
Experimentation	Performance	Verification

The only change from current practice in the above is the addition of the General Term where one is appropriate.

D. *The Monthly Index*

This would consist of

—a listing of the first two nodes of the tree with their respective page numbers as at present, and

—a list of the general terms with, for each term, the review numbers of each review labeled with that term.

E. *Other Periodic Indexes*

As at present, the entire tree *without* subject descriptors would be published every two or three months. Once or twice a year a complete index would be published; this would consist of a single alphabetic listing as follows:

—Node names

—Subject descriptors

—General terms

—Other terms which are not yet subject descriptors but which are candidates to be at the next revision because of more than occasional designation by authors or reviewers as key terms

This index would be cross-referenced, term-by-term to a tree node number or the general term list, as appropriate.

F. *Mapping the Old System to the New One*

In order to facilitate getting used to the new system and to aid retrospective searches involving both the old and new systems, a mapping from the old tree to the new will be provided in, at least, the first issue in which the new classification scheme is used.

Schedule

For logistic reasons which are not worth detailing here, it is necessary that any new classification scheme be initiated in a January issue of *CR*. Our hope is to implement the final version of the scheme presented in this article on *1 January 1982.* To accomplish this will require

- that various administrative tasks at ACM Headquarters can be handled over the next few months to allow implementation at the beginning of next year and
- that the reaction to this draft does not necessitate major changes.

We cannot, of course, predict whether or not the second requirement will be satisfied. But, in order for us to make

a judgment on this, all responses to this article must be received no later than *31 August 1981.*

Request for Comments from Readers

The remainder of this report consists of a listing of the current version of the classification scheme, first with just the nodes given and then with all nodes and subject descriptors given. The kind of comments we should like about this scheme are:

- general comments on the philosophy.
- specific criticisms of the tree structure itself.
- suggestions for deletions and/or additions to the subject descriptors for any node.

We would appreciate it if comments are made in the context of an understanding that the new classification scheme is and must be a compromise among many points of view and perspectives on what the computing field really is.

One specific item on which the Committee would particularly like response concerns Node D. On the Committee itself and among the reviewers of our work there was considerable disagreement on the *need for a Data node. On the one hand,* this separate node recognizes the distinctness of considerations about data from those about algorithms which are spread through a variety of other nodes. The existence of this node is also a convenient place to group a set of topics which do not fit very easily into other places. *On the other hand,* the topics included under this node are really quite disparate. And each *could* be put somewhere else, for example much of D.1 under G.2, DATABASE MANAGEMENT, D.2 under C.3.3, Language Constructs, D.3 under C.4.6 Protection and D.4 under G.1.1 Systems and Information Theory.

The final disposition of this matter will likely depend upon the number and weight of the comments received from readers of Communications.

The proposed new *CR* Classification Tree without subject descriptors appears on pages 423–425; pages 426–433 include the subject descriptors.

Acknowledgments. In the development of this classification the Committee has had the help of many colleagues who served as consultants on particular nodes or as reviewers of our efforts or otherwise made suggestions to us. Their names are listed below.

Paul W. Abrahams
Robert M. Aiken
James P. Anderson
Norman I. Badler
Laslzo A. Belady
C. Gordon Bell
Robert P. Bigelow
Harold Borko
Frederick P. Brooks
George G. Dodd
Hubert E. Dunsmore
Philip H. Enslow, Jr.
Aaron Finerman
M. David Freedman
Bernard A. Galler
Bruce Gilchrist
C. C. Gotlieb
John Granacki
John Grason
Stanley Habib
John P. Hayes
Richard D. Jenks
Butler W. Lampson
Joel Moses
Nils J. Nilsson
John C. Reynolds
John R. Rice
Azriel Rosenfeld
Daniel J. Rosenkrantz
Gerard A. Salton
Lawrence F. Shampine
Stuart C. Shapiro
Ben Shneiderman
Val Silbey
Robert F. Sproull
Sargur N. Srihari
Donald F. Stanat
Harold S. Stone
Michael Stonebraker
Andrew S. Tanenbaum
Steven L. Tanimoto
James W. Thatcher
Godfried T. Toussaint
B. L. Trippett
Andries Van Dam
David S. Wise
David C. Wood

And particular thanks must go to Robert L. Ashenhurst, Editor-in-Chief of *Communications,* whose interest in this project and advice has been of great benefit to the Committee.

The Proposed New *CR* Classification Tree Without *Subject Descriptors*

In order for the reader to be able to grasp the essential structure and coverage of the proposed new classification scheme we present first the complete tree but with no subject descriptors attached to the leaves. Node designations in parentheses are cross-references to other nodes which cover similar or related material.

- A. HARDWARE
  - A.0 General
  - A.1 Control Structures and Microprogramming (C.3.2)
    - A.1.0 General
    - A.1.1 Control Design Styles
    - A.1.2 Control Structure Performance Analysis and Design Aids
    - A.1.3 Control Structure Reliability, Testing and Fault-Tolerance
    - A.1.4 Microprogram Design Aids (C.2.2, C.2.4, C.3.2, C.3.4)
    - A.1.5 Microcode Applications
    - A.1.6 Miscellaneous
  - A.2 Arithmetic and Logic Structures
    - A.2.0 General
    - A.2.1 Design Styles (B.1.1-2)
    - A.2.2 Performance Analysis and Design Aids
    - A.2.3 Reliability, Testing and Fault-Tolerance
    - A.2.4 Miscellaneous
  - A.3 Memory Structures
    - A.3.1 General
    - A.3.2 Design Styles (C.4.2)
    - A.3.3 Performance Analysis and Design Aids (B.4)
    - A.3.4 Reliability, Testing and Fault-Tolerance
    - A.3.5 Miscellaneous
  - A.4 Input/Output and Data Communications
    - A.4.0 General
    - A.4.1 Data Communications Devices
    - A.4.2 Input/Output Devices
    - A.4.3 Interconnections (subsystems)
    - A.4.4 Performance Analysis and Design Aids
    - A.4.5 Reliability, Testing and Fault-Tolerance
    - A.4.6 Miscellaneous
  - A.5 Register-Transfer-Level Implementation
    - A.5.0 General
    - A.5.1 Design
    - A.5.2 Design Aids
    - A.5.3 Reliability and Testing
    - A.5.4 Miscellaneous
  - A.6 Logic Design
    - A.6.0 General
    - A.6.1 Design Styles
    - A.6.2 Reliability and Testing
    - A.6.3 Design Aids
    - A.6.4 Miscellaneous
  - A.7 Integrated Circuits
    - A.7.0 General
    - A.7.1 Types and Design Styles
    - A.7.2 Design Aids
    - A.7.3 Reliability and Testing
    - A.7.4 Miscellaneous
  - A.8 Miscellaneous
- B. COMPUTER SYSTEMS ORGANIZATION
  - B.0 General
  - B.1 Processor Architectures
    - B.1.0 General
    - B.1.1 Single Data Stream Architectures
    - B.1.2 Multiple Data Stream Architectures (Multiprocessors)
    - B.1.3 Other Architecture Styles
    - B.1.4 Miscellaneous
  - B.2 Computer-Communication Networks
    - B.2.0 General
    - B.2.1 Network Architecture and Design
    - B.2.2 Network Protocols
    - B.2.3 Network Operations
    - B.2.4 Distributed Systems
    - B.2.5 Local Networks
    - B.2.6 Miscellaneous
  - B.3 Special-Purpose and Application-Based Systems
  - B.4 Performance of Systems
  - B.5 Miscellaneous
- C. SOFTWARE
  - C.0 General
  - C.1 Programming Techniques (D)
    - C.1.0 General
    - C.1.1 Applicative (Functional) Programming
    - C.1.2 Automatic Programming (H.2.2)
    - C.1.3 Concurrent Programming
    - C.1.4 Sequential Programming
    - C.1.5 Miscellaneous
  - C.2 Software Engineering (J.6.3)
    - C.2.0 General (J.5.1)
    - C.2.1 Requirements/Specifications
    - C.2.2 Tools and Techniques
    - C.2.3 Coding
    - C.2.4 Program Verification
    - C.2.5 Testing and Debugging
    - C.2.6 Distribution and Maintenance
    - C.2.7 Metrics
    - C.2.8 Management (J.6.3)
    - C.2.9 Miscellaneous

- C.3 Programming Languages
  - C.3.0 General
  - C.3.1 Formal Definitions and Theory (E.3.2-3, E.4.2-3)
  - C.3.2 Language Classifications
  - C.3.3 Language Constructs (D.2)
  - C.3.4 Processors
  - C.3.5 Miscellaneous
- C.4 Operating Systems (B)
  - C.4.0 General
  - C.4.1 Process Management
  - C.4.2 Storage Management
  - C.4.3 File Systems Management
  - C.4.4 Communications Management (B.2)
  - C.4.5 Reliability
  - C.4.6 Protection
  - C.4.7 Organization and Design
  - C.4.8 Performance (B.5, H.6)
  - C.4.9 Systems Programs and Utilities
  - C.4.10 Miscellaneous
- C.5 Miscellaneous

D. DATA
- D.0 General
- D.1 Data Structures
- D.2 Data Storage Representations
- D.3 Data Encryption
- D.4 Coding and Information Theory (G.1.1)
- D.5 Miscellaneous

E. THEORY OF COMPUTATION
- E.0 General
- E.1 Computation by Abstract Devices
  - E.1.0 General
  - E.1.1 Models of Computation (E.3.1)
  - E.1.2 Modes of Computation
  - E.1.3 Complexity Classes (E.2)
  - E.1.4 Miscellaneous
- E.2 Analysis of Algorithms and Problems (A.6-7, E.1.3)
  - E.2.0 General
  - E.2.1 Numerical Algorithms and Problems (F.1, F.4, H.1)
  - E.2.2 Nonnumerical Algorithms and Problems (D.2-4, F.2, G.2-3)
  - E.2.3 Tradeoffs among Complexity Measures
  - E.2.4 Miscellaneous
- E.3 Logic and Meaning
  - E.3.0 General
  - E.3.1 Mathematical Logic (E.1.1, H.2.2-3)
  - E.3.2 Specifying and Verifying and Reasoning about Programs (C.2.1, C.2.4, C.3.1, D.1)
  - E.3.3 Semantics of Programming Languages (C.3.1)
  - E.3.4 Studies of Program Constructs (C.3.2-3)
  - E.3.5 Miscellaneous
- E.4 Automata and Formal Languages
  - E.4.0 General
  - E.4.1 Automata (E.1.1)
  - E.4.2 Grammars and Other Rewriting Systems (C.3.1)
  - E.4.3 Formal Languages (C.3.1)
  - E.4.4 Miscellaneous
- E.5 Miscellaneous

F. MATHEMATICS OF COMPUTING
- F.0 General
- F.1 Numerical Analysis
  - F.1.0 General
  - F.1.1 Interpolation
  - F.1.2 Approximation
  - F.1.3 Numerical Linear Algebra
  - F.1.4 Quadrature and Numerical Differentiation
  - F.1.5 Roots of Nonlinear Equations
  - F.1.6 Optimization
  - F.1.7 Ordinary Differential Equations
  - F.1.8 Partial Differential Equations
  - F.1.9 Integral Equations
  - F.1.10 Miscellaneous
- F.2 Discrete Mathematics
  - F.2.0 General
  - F.2.1 Combinatorics (E.2.2)
  - F.2.2 Graph Theory (E.2.2)
  - F.2.3 Miscellaneous
- F.3 Statistics
- F.4 Mathematical Software
- F.5 Miscellaneous

G. INFORMATION SYSTEMS
- G.0 General
- G.1 Models and Principles
  - G.1.0 General
  - G.1.1 Systems and Information Theory (D.4)
  - G.1.2 Man/Machine Systems
  - G.1.3 Miscellaneous
- G.2 Database Management
  - G.2.0 General
  - G.2.1 Logical Design
  - G.2.2 Physical Design
  - G.2.3 Languages
  - G.2.4 Systems
  - G.2.5 Heterogeneous Databases
  - G.2.6 Database Machines
  - G.2.7 Database Administration
  - G.2.8 Miscellaneous
- G.3 Information Storage and Retrieval
  - G.3.0 General
  - G.3.1 Content Analysis and Indexing
  - G.3.2 Information Storage
  - G.3.3 Information Search and Retrieval
  - G.3.4 Systems and Software
  - G.3.5 On-Line Information Services
  - G.3.6 Library Automation
  - G.3.7 Miscellaneous
- G.4 Office Automation (H.7)
  - G.4.0 General
  - G.4.1 Equipment
  - G.4.2 Electronic Mail
  - G.4.3 Word Processing
  - G.4.4 Miscellaneous
- G.5 Miscellaneous

H. COMPUTING METHODOLOGIES
- H.0 General
- H.1 Algebraic Manipulation
  - H.1.0 General
  - H.1.1 Expressions and Their Representation (D.1-2)
  - H.1.2 Algorithms (E.2.1-2)
  - H.1.3 Languages and Systems (C.3.2-3, E.2.2)
  - H.1.4 Applications
  - H.1.5 Miscellaneous
- H.2 Artificial Intelligence
  - H.2.0 General
  - H.2.1 Applications and Expert Systems (G.4, I)
  - H.2.2 Automatic Programming (C.1.2, E.3.2)
  - H.2.3 Deduction and Theorem Proving
  - H.2.4 Knowledge Representation Formalisms and Methods
  - H.2.5 Languages and Software (C.3.2)
  - H.2.6 Learning (J.3.2)
  - H.2.7 Natural Language Processing
  - H.2.8 Problem Solving, Control Methods and Search (E.2.2)
  - H.2.9 Robotics
  - H.2.10 Vision and Scene Analysis (H.4.8, H.5)
  - H.2.11 Miscellaneous
- H.3 Computer Graphics
  - H.3.0 General
  - H.3.1 Hardware Architecture (A.4.2)
  - H.3.2 Graphics Systems (B.2.1, B.2.4, B.3)
  - H.3.3 Picture and Image Generation
  - H.3.4 Graphics Utilities
  - H.3.5 Object Modeling
  - H.3.6 Methodology and Techniques
  - H.3.7 Three-Dimensional Graphics and Realism
  - H.3.8 Miscellaneous
- H.4 Image Processing
  - H.4.0 General
  - H.4.1 Digitization
  - H.4.2 Compression (coding) (D.4)
  - H.4.3 Enhancement
  - H.4.4 Restoration
  - H.4.5 Reconstruction
  - H.4.6 Segmentation
  - H.4.7 Feature Measurement
  - H.4.8 Scene Analysis
  - H.4.9 Applications
  - H.4.10 Miscellaneous
- H.5 Pattern Recognition
  - H.5.0 General
  - H.5.1 Models
  - H.5.2 Design Methodology
  - H.5.3 Clustering
  - H.5.4 Applications
  - H.5.5 Implementation (B.3)
  - H.5.6 Miscellaneous
- H.6 Simulation and Modeling
  - H.6.0 General
  - H.6.1 Simulation Theory
  - H.6.2 Simulation Languages
  - H.6.3 Applications
  - H.6.4 Model Validation and Analysis
  - H.6.5 Miscellaneous
- H.7 Text Processing (G.4)
  - H.7.0 General
  - H.7.1 Text Editing
  - H.7.2 Document Preparation
  - H.7.3 Index Generation
  - H.7.4 Miscellaneous
- H.8 Miscellaneous

I. COMPUTER APPLICATIONS
- I.0 General
- I.1 Administrative Data Processing
- I.2 Physical Sciences and Engineering
- I.3 Life and Medical Sciences
- I.4 Social and Behavioral Sciences
- I.5 Arts and Humanities
- I.6 Computer-Aided Systems
- I.7 Computers in Other Systems
- I.8 Miscellaneous

J. COMPUTING MILIEUX
- J.0 General
- J.1 The Computer Industry
- J.2 History of Computing
- J.3 Computers and Education
  - J.3.0 General
  - J.3.1 Computer Uses in Education
  - J.3.2 Computer and Information Science Education
  - J.3.3 Miscellaneous
- J.4 Computers and Society
  - J.4.0 General
  - J.4.1 Public Policy Issues
  - J.4.2 Social Issues
  - J.4.3 Miscellaneous
- J.5 Legal Aspects of Computing
  - J.5.0 General
  - J.5.1 Software Protection
  - J.5.2 Governmental Issues
  - J.5.3 Miscellaneous
- J.6 Management of Computing and Information Systems
  - J.6.0 General
  - J.6.1 Project and People Management
  - J.6.2 Installation Management
  - J.6.3 Software Management
  - J.6.4 Miscellaneous
- J.7 The Computing Profession
  - J.7.0 General
  - J.7.1 Occupations
  - J.7.2 Organizations
  - J.7.3 Testing, Certification, and Licensing
  - J.7.4 Miscellaneous
- J.8 Miscellaneous

K. GENERAL LITERATURE
- K.0 General
- K.1 Introductory and Survey
- K.2 Reference (e.g., dictionaries, encyclopedias, glossaries)
- K.3 Miscellaneous

The Proposed New *CR* Classification Tree

What follows is a complete listing of the proposed new classification tree with all subject descriptors except the names of languages or systems which are implicit subject descriptors under appropriate nodes (e.g., Pascal under C.3.2 Language Classification or Unix under C.4.0 Operating Systems–General).

A. HARDWARE

A.0 General

Design management

A.1 Control Structures and Microprogramming (C.3.2)

A.1.0 General

A.1.1 Control Design Styles

Hardwired
Microprogrammed logic arrays
Writable control store

A.1.2 Control Structure Performance Analysis and Design Aids

Automatic synthesis
Formal models
Simulation

A.1.3 Control Structure Reliability, Testing and Fault-Tolerance

Diagnostics
Error-checking
Redundant design
Test generation

A.1.4 Microprogram Design Aids (C.2.2, C.2.4, C.3.2, C.3.4)

Firmware engineering
Machine-independent microcode generation
Languages and compilers
Optimization
Verification

A.1.5 Microcode Applications

Direct data manipulation
Firmware support of operating systems/instruction sets
Instruction set interpretation
Peripheral control
Special-purpose

A.1.6 Miscellaneous

A.2 Arithmetic and Logic Structures

A.2.0 General

A.2.1 Design Styles (B.1.1-2)

Calculator
Parallel
Pipeline

A.2.2 Performance Analysis and Design Aids

Simulation
Verification
Worst-case analysis

A.2.3 Reliability, Testing and Fault-Tolerance

Diagnostics
Error-checking
Redundant design
Test generation

A.2.4 Miscellaneous

A.3 Memory Structures

A.3.1 General

A.3.2 Design Styles (C.4.2)

Associative memories
Cache memories
Interleaved memories
Mass storage
Primary memory
Sequential-access memory
Shared memory
Virtual memory

A.3.3 Performance Analysis and Design Aids (B.4)

Formal models
Simulation
Worst-case analysis

A.3.4 Reliability, Testing and Fault-Tolerance

Diagnostics
Error-checking
Redundant design
Test generation

A.3.5 Miscellaneous

A.4 Input/Output and Data Communications

A.4.0 General

A.4.1 Data Communications Devices

Processors
Receivers (e.g., voice, data, image)
Transmitters

A.4.2 Input/Output Devices

Channels and controllers
Data terminals and printers
Image display
Voice

A.4.3 Interconnections (subsystems)

Asynchronous/synchronous operation
Fiber optics
Interfaces
Physical structures (e.g., backplanes, cables, chip carriers)
Topology (e.g., buses, point-to-point)

A.4.4 Performance Analysis and Design Aids

Formal models
Simulation
Verification
Worst-case analysis

A.4.5 Reliability, Testing and Fault-Tolerance

Built-in tests
Diagnostics
Error-checking
Hardware reliability
Redundant design
Test generation

A.4.6 Miscellaneous

A.5 Register-Transfer-Level Implementation

A.5.0 General

A.5.1 Design

Arithmetic and logic units
Control design
Data-path design
Memory design
Styles (e.g., parallel, pipelined, special-purpose)

A.5.2 Design Aids

Automatic synthesis
Hardware description languages
Optimization
Simulation
Verification

A.5.3 Reliability and Testing

Built-in tests
Error-checking
Redundant design
Test generation
Testability

A.5.4 Miscellaneous

A.6 Logic Design

A.6.0 General

A.6.1 Design Styles

Cellular arrays and automata
Combinational logic
Logic arrays
Memory control and access
Memory used as logic
Parallel circuits
Sequential circuits

A.6.2 Reliability and Testing

Built-in tests
Error-checking
Redundant design
Test generation
Testability

A.6.3 Design Aids

Arithmetic synthesis
Hardware description languages
Optimization
Simulation
Switching theory
Verification

A.6.4 Miscellaneous

A.7 Integrated Circuits

A.7.0 General

A.7.1 Types and Design Styles

Advanced technologies
Algorithms implemented in hardware
Gate arrays
Input/Output circuits
Memory technologies
Microprocessors and microcomputers
Standard cells
VLSI (very large scale integration)

A.7.2 Design Aids

Graphics
Layout
Placement and routing
Simulation
Verification

A.7.3 Reliability and Testing

Built-in tests
Error-checking
Redundant design
Testability
Test generation

A.7.4 Miscellaneous

A.8 Miscellaneous

B. COMPUTER SYSTEMS ORGANIZATION

B.0 General

Hardware/software interfaces
Instruction set design
System architectures
Systems specification methodology

B.1 Processor Architectures

B.1.0 General

Analog computers
Hybrid systems

B.1.1 Single Data Stream Architectures

Multiple-instruction-stream, single-data-stream processors (MISD)
Pipeline processors
Single-instruction-stream, single-data-stream processors (SISD)
Von Neumann architectures

B.1.2 Multiple Data Stream Architectures (Multiprocessors)

Array and vector processors
Associative processors
Interconnection architectures (e.g., common bus, multiport memory, crossbar switch)
Multiple-instruction-stream, multiple-data-stream processors (MIMD)
Parallel processors
Pipeline processors
Single-instruction-stream, multiple-data-stream processors (SIMD)

B.1.3 Other Architecture Styles

Adaptable architectures
Capability architectures
Data-flow architectures
High-level language architectures
Stack-oriented processors

B.1.4 Miscellaneous

B.2 Computer-Communication Networks

B.2.0 General

Security

B.2.1 Network Architecture and Design

Centralized networks
Distributed networks
Network communications
Network topology

B.2.2 Network Protocols

Protocol architecture
Protocol verification

B.2.3 Network Operations

Network management
Network monitoring
Public networks

B.2.4 Distributed Systems

Distributed applications
Distributed databases
Network operating systems

B.2.5 Local Networks

Access schemes
Buses
Rings

B.2.6 Miscellaneous

B.3 Special-Purpose and Application-Based Systems

Process control systems
Real-time systems

B.4 Performance of Systems

Design studies
Measurement techniques
Modeling techniques
Performance attributes
Reliability, availability, and serviceability

B.5 Miscellaneous

C. SOFTWARE

C.0 GENERAL

Software psychology

C.1 PROGRAMMING TECHNIQUES (D)

C.1.0 General

C.1.1 Applicative (Functional) Programming

C.1.2 Automatic Programming (H.2.2)

C.1.3 Concurrent Programming

C.1.4 Sequential Programming

C.1.5 Miscellaneous

C.2 SOFTWARE ENGINEERING (J.6.3)

C.2.0 General (J.5.1)

Protection mechanisms
Standards

C.2.1 Requirements/Specifications

Languages
Methodologies
Tools

C.2.2 Tools and Techniques

Decision tables
Flow charts
Modules and interfaces
Programmer workbench
Software libraries
Structured programming
Top-down programming

C.2.3 Coding

Language selection
Pretty printers
Program editors
Reentrant code
Standards

C.2.4 Program Verification

Assertion checkers
Correctness proofs
Reliability
Validation

C.2.5 Testing and Debugging

Debugging aids
Diagnostics
Dumps
Error handling and recovery
Monitors
Quality assurance
Symbolic execution
Test data generators
Tracing

C.2.6 Distribution and Maintenance

Corrections
Documentation
Enhancement
Extensibility
Portability
Restructuring
Version control

C.2.7 Metrics

Complexity measures
Software science

C.2.8 Management (J.6.3)

Copyrights
Cost estimation
Life cycle
Productivity
Programming teams
Software configuration management
Software quality assurance (SQA)

C.2.9 Miscellaneous

C.3 PROGRAMMING LANGUAGES

C.3.0 General

Standards

C.3.1 Formal Definitions and Theory (E.3.2-3, E.4.2-3)

Semantics
Syntax

C.3.2 Language Classifications

Applicative languages
Data-flow languages
Extensible languages
Macro and assembly languages
Microprogramming languages
Nonprocedural languages
Very high-level languages

C.3.3 Language Constructs (D.2)

Abstract data types
Concurrent programming structures
Control structures
Coroutines
Data types and structures
Input/Output
Procedures, functions and subroutines

C.3.4 Processors

Code generation
Compilers
Interpreters
Parsing
Preprocessors
Programming and run-time environments
Optimization
Translator writing systems and compiler generators

C.3.5 Miscellaneous

C.4 OPERATING SYSTEMS (B)

C.4.0 General

C.4.1 Process Management

Concurrency
Deadlocks
Multiprocessing/multiprogramming
Mutual exclusion
Scheduling
Synchronization

C.4.2 Storage Management

Allocation/deallocation strategies
Distributed memories
Main memory
Secondary storage devices
Segmentation
Storage hierarchies
Swapping
Virtual memory

C.4.3 File Systems Management

Access methods
Directory structures
Distributed file systems
File organization
Maintenance

C.4.4 Communications Management (B.2)

Buffering
Input/Output
Message sending
Network communication
Terminal management

C.4.5 Reliability

Atomicity
Backup procedures
Checkpoint/restart
Transactions
Verification

C.4.6 Protection

Authentication
Cryptographic controls
Information flow controls
Security kernels
Verification

C.4.7 Organization and Design

Batch processing systems
Distributed systems
Hierarchical systems
Interactive systems
Real-time systems

C.4.8 Performance (B.5, H.6)

Measurements
Modeling and prediction
Monitors
Operational analysis
Queueing theory
Simulation
Stochastic analysis

C.4.9 Systems Programs and Utilities

Command and control languages
Linkers
Loaders
Servers

C.4.10 Miscellaneous

C.5 MISCELLANEOUS

D. DATA

D.0 GENERAL

D.1 DATA STRUCTURES

Arrays
Graphs
Lists
Trees

D.2 DATA STORAGE REPRESENTATIONS

Composite structures
Contiguous representations
Hash-table representations
Linked representations
Primitive data items

D.3 DATA ENCRYPTION

Data encryption standard (DES)
Public key cryptosystems

D.4 CODING AND INFORMATION THEORY (G.1.1)

Data compaction and compression
Formal models of communication
Nonsecret encoding schemes

D.5 MISCELLANEOUS

E. THEORY OF COMPUTATION

E.0 GENERAL

E.1 COMPUTATION BY ABSTRACT DEVICES

E.1.0 General

E.1.1 Models of Computation (E.3.1)

Bounded-action devices (e.g., Turing machines, random access machines)
Computability
Relations among models
Self-modifying machines
Unbounded-action devices (e.g., circuits, networks of machines)

E.1.2 Modes of Computation

Alternation and nondeterminism
Parallelism
Probabilistic computation
Relations among modes
Relativized computation

E.1.3 Complexity Classes (E.2)

Complexity hierarchies
Machine-independent complexity
Reducibility and completeness
Relations among complexity classes
Relations among complexity measures

E.1.4 Miscellaneous

E.2 ANALYSIS OF ALGORITHMS AND PROBLEMS (A.6-7, E.1.3)

E.2.0 General

E.2.1 Numerical Algorithms and Problems (F.1, F.4, H.1)

Computation of transforms (e.g., Fast Fourier transform)
Computations in finite fields
Computations on matrices
Computations on polynomials
Number-theoretic computations (e.g., factoring, primality testing)

E.2.2 Nonnumerical Algorithms and Problems (D.2-4, F.2, G.2-3)

Complexity of proof procedures
Computations on discrete structures
Geometrical problems
Pattern matching
Routing and layout
Sequencing and scheduling
Sorting and searching

E.2.3 Tradeoffs among Complexity Measures

E.2.4 Miscellaneous

E.3 LOGIC AND MEANING

E.3.0 General

E.3.1 Mathematical Logic (E.1.1, H.2.2-3)

Computability theory
Computational logic
Lambda calculus and related systems
Logic programming
Mechanical theorem proving
Recursive function theory

E.3.2 Specifying and Verifying and Reasoning about Programs (C.2.1, C.2.4, C.3.1, D.1)

Assertions
Invariants
Logics of programs
Mechanical verification
Pre- and post-conditions
Specification techniques

E.3.3 Semantics of Programming Languages (C.3.1)

Algebraic approaches to semantics
Denotational semantics
Operational semantics

E.3.4 Studies of Program Constructs (C.3.2-3)

Control primitives
Functional constructs
Program and recursion schemes
Type structure

E.3.5 Miscellaneous

E.4 AUTOMATA AND FORMAL LANGUAGES

E.4.0 General

E.4.1 Automata (E.1.1)

Automata types (e.g., Turing machines, cellular automata)
Decision problems
Modes of computation (e.g., alternating, nondeterministic, probabilistic)
Resource-bounded automata

E.4.2 Grammars and Other Rewriting Systems (C.3.1)

Decision problems
Grammar types (e.g., context-free, context-sensitive)
Parallel rewriting systems (e.g., developmental systems, L-systems)
Parsing
Thue systems

E.4.3 Formal Languages (C.3.1)

Algebraic language theory
Classes defined by grammars or automata (e.g., regular sets, recursive sets)
Classes defined by resource-bounded automata
Decision problems
Operations on languages

E.4.4 Miscellaneous

E.5 MISCELLANEOUS

F. MATHEMATICS OF COMPUTING

F.0 GENERAL

F.1 NUMERICAL ANALYSIS

F.1.0 General

Computer arithmetic
Condition (and ill-condition)
Error analysis
Numerical algorithms
Parallel algorithms
Stability (and instability)

F.1.1 Interpolation

Difference formulas
Extrapolation
Interpolation formulas
Smoothing
Spline and piecewise polynomial interpolation

F.1.2 Approximation

Chebyshev approximation and theory
Elementary function approximation
Least squares approximation
Linear approximation
Minimax approximation and algorithms
Nonlinear approximation
Rational approximation
Spline and piecewise polynomial approximation

F.1.3 Numerical Linear Algebra

Conditioning
Determinants
Eigenvalues
Error analysis
Linear systems (direct and iterative methods)
Matrix inversion
Pseudoinverses
Sparse and very large systems

F.1.4 Quadrature and Numerical Differentiation

Adaptive quadrature
Equal interval integration
Error analysis
Finite difference methods
Gaussian quadrature
Iterated methods
Multiple quadrature

F.1.5 Roots of Nonlinear Equations

Convergence
Error analysis
Iterative methods
Polynomials, methods for
Systems of equations

F.1.6 Optimization

Constrained optimization
Gradient methods
Integer programming
Least squares methods
Linear programming
Nonlinear programming

F.1.7 Ordinary Differential Equations

Boundary value problems
Convergence and stability
Error analysis
Initial value problems
Multistep methods
Single step methods
Stiff equations

F.1.8 Partial Differential Equations

Difference methods
Elliptic equations
Finite element methods
Hyperbolic equations
Method of lines
Parabolic equations

F.1.9 Integral Equations

Fredholm equations
Integro-differential equations
Volterra equations

F.1.10 Miscellaneous

F.2 DISCRETE MATHEMATICS

F.2.0 General

F.2.1 Combinatorics (E.2.2)

Combinatorial algorithms
Counting problems
Generating functions
Permutations and combinations
Recurrences and difference equations

F.2.2 Graph Theory (E.2.2)

Graph algorithms
Network problems
Path and circuit problems
Trees

F.2.3 Miscellaneous

F.3 STATISTICS

Random number generation
Statistical computing

F.4 MATHEMATICAL SOFTWARE

Algorithm analysis
Certification and testing
Efficiency
Portability
Reliability and robustness
Verification

F.5 MISCELLANEOUS

G. INFORMATION SYSTEMS

G.0 GENERAL

G.1 MODELS AND PRINCIPLES

G.1.0 General

G.1.1 Systems and Information Theory (D. 4)

General systems theory
Information theory
Value of information

G.1.2 Man/Machine Systems

Human factors
Human information processing

G.1.3 Miscellaneous

Decision support systems
Information support systems

G.2 DATABASE MANAGEMENT

G.2.0 General

G.2.1 Logical Design

Data models
Normal forms
Schema and subschema

G.2.2 Physical Design

Access methods
Deadlock avoidance
Recovery and restart

G.2.3 Languages

Data description languages (DDL)
Data manipulation languages (DML)
Query languages
Report writers

G.2.4 Systems

Distributed systems
Query processing
Transaction processing

G.2.5 Heterogeneous Databases

Data translation
Program translation

G.2.6 Database Machines

G.2.7 Database Administration

Data dictionary/directory
Logging and recovery

G.2.8 Miscellaneous

G.3 Information Storage and Retrieval

G.3.0 General

G.3.1 Content Analysis and Indexing

Abstracting methods
Dictionaries
Indexing methods
Linguistic processing
Thesauruses

G.3.2 Information Storage

Record classification
File organization

G.3.3 Information Search and Retrieval

Query formulation
Retrieval models
Search process
Selection process

G.3.4 Systems and Software

Current awareness systems (selective dissemination of information–SDI)
Information networks
Question-answering (fact retrieval) systems

G.3.5 On-Line Information Services

Data bank sharing

G.3.6 Library Automation

G.3.7 Miscellaneous

G.4 Office Automation (H.7)

G.4.0 General

Teleconferencing

G.4.1 Equipment

G.4.2 Electronic Mail

G.4.3 Word Processing

Text editing

G.4.4 Miscellaneous

G.5 Miscellaneous

H. COMPUTING METHODOLOGIES

H.0 General

H.1 Algebraic Manipulation

H.1.0 General

H.1.1 Expressions and Their Representation (D.1–2)

Representations (general and polynomial)
Simplification of expressions

H.1.2 Algorithms (E.2.1–2)

Algebraic algorithms
Analysis of algorithms
Nonalgebraic algorithms

H.1.3 Languages and Systems (C.3.2–3, E.2.2)

Evaluation strategies
Nonprocedural languages
Special-purpose algebraic systems
Special-purpose hardware
Substitution mechanisms

H.1.4 Applications

H.1.5 Miscellaneous

H.2 Artifical Intelligence

H.2.0 General

H.2.1 Applications and Expert Systems (G.4, I)

Cartography
Games
Industrial automation
Law
Medicine and science
Office automation

H.2.2 Automatic Programming (C.1.2, E.3.2)

Automatic analysis of algorithms
Program modification
Program synthesis
Program transformation
Program verification

H.2.3 Deduction and Theorem Proving

Answer/reason extraction
Deduction (e.g., natural, rule-based)
Logic programming
Mathematical induction
Metatheory
Nonmonotonic reasoning and belief revision
Resolution

H.2.4 Knowledge Representation Formalisms and Methods

Frames and scripts
Predicate logic
Relation systems
Representation languages
Representations (procedural and rule-based)
Semantic networks

H.2.5 Languages and Software (C.3.2)

H.2.6 Learning (J.3.2)

Analogies
Concept learning
Induction
Knowledge acquisition
Language acquisition
Parameter learning

H.2.7 Natural Language Processing

Language generation
Language models
Language parsing and understanding
Machine translation
Speech recognition and understanding
Text analysis

H.2.8 Problem Solving, Control Methods and Search (E.2.2)

Backtracking
Dynamic programming
Graph and tree search strategies
Heuristic methods
Plan execution, formation, generation

H.2.9 Robotics

Manipulators
Propelling mechanisms
Sensors

H.2.10 Vision and Scene Analysis (H.4.8, H.5)

Architecture and control structures
Intensity, color, photometry and thresholding
Modeling and recovery of physical attributes
Motion
Perceptual reasoning
Representations, data structures and transforms
Shape
Texture

H.2.11 Miscellaneous

H.3 COMPUTER GRAPHICS
- H.3.0 General
- H.3.1 Hardware Architecture (A.4.2)
  - *Input devices*
  - *Raster display devices*
  - *Storage devices*
  - *Vector display devices*
- H.3.2 Graphics Systems (B.2.1, B.2.4, B.3)
  - *Distributed/network graphics*
  - *Remote systems*
  - *Stand-alone systems*
- H.3.3 Picture and Image Generation
  - *Image representation*
  - *Viewing algorithms*
- H.3.4 Graphics Utilities
  - *Application packages*
  - *Graphics packages*
  - *Picture description languages*
  - *Software support*
- H.3.5 Object Modeling
  - *Curve and surface representations*
  - *Hierarchy and geometric transformations*
  - *Modeling packages*
- H.3.6 Methodology and Techniques
  - *Device independence*
  - *Economics*
  - *Interaction techniques*
- H.3.7 Three-Dimensional Graphics and Realism
  - *Animation*
  - *Color, shading and texture*
  - *Hidden line/surface elimination*
- H.3.8 Miscellaneous

H.4 IMAGE PROCESSING
- H.4.0 General
  - *Image displays*
  - *Image processing software*
- H.4.1 Digitization
  - *Quantization*
  - *Sampling*
  - *Scanning*
- H.4.2 Compression (coding) (D.4)
  - *Approximate methods*
  - *Exact coding*
- H.4.3 Enhancement
  - *Filtering*
  - *Geometric correction*
  - *Grayscale manipulation*
  - *Registration*
  - *Sharpening and deblurring*
  - *Smoothing*
- H.4.4 Restoration
  - *Inverse filtering*
  - *Kalman filtering*
  - *Pseudoinverse restoration*
  - *Wiener filtering*
- H.4.5 Reconstruction
  - *Series expansion methods*
  - *Summation methods*
  - *Transform methods*
- H.4.6 Segmentation
  - *Edge and feature detection*
  - *Pixel classification*
  - *Region growing, partitioning*
- H.4.7 Feature Measurement
  - *Invariants*
  - *Moments*
  - *Projections*
  - *Size and shape*
  - *Texture*
- H.4.8 Scene Analysis
  - *Depth cues*
  - *Photometry*
  - *Range data*
  - *Stereo*
  - *Time-varying imagery*
- H.4.9 Applications
- H.4.10 Miscellaneous

H.5 PATTERN RECOGNITION
- H.5.0 General
- H.5.1 Models
  - *Deterministic*
  - *Geometric*
  - *Fuzzy set*
  - *Statistical*
  - *Structural*
- H.5.2 Design Methodology
  - *Classifier design and evaluation*
  - *Feature evaluation and selection*
  - *Pattern analysis*
- H.5.3 Clustering
  - *Algorithms*
  - *Similarity measures*
- H.5.4 Applications
  - *Computer vision*
  - *Text processing*
  - *Waveform analysis*
- H.5.5 Implementation (B.3)
  - *Interactive systems*
  - *Special architectures*
- H.5.6 Miscellaneous

H.6 SIMULATION AND MODELING
- H.6.0 General
- H.6.1 Simulation Theory
  - *Model classification*
  - *Types of simulation (continuous and discrete)*
- H.6.2 Simulation Languages
- H.6.3 Applications
- H.6.4 Model Validation and Analysis
- H.6.5 Miscellaneous

H.7 TEXT PROCESSING (G.4)
- H.7.0 General
- H.7.1 Text Editing
  - *Languages*
  - *Spelling*
- H.7.2 Document Preparation
  - *Format and notation*
  - *Languages*
  - *Photocomposition*
- H.7.3 Index Generation
- H.7.4 Miscellaneous

H.8 MISCELLANEOUS

I. COMPUTER APPLICATIONS

I.0 General

I.1 Administrative Data Processing

Business
Education
Financial (e.g., EFTS)
Government
Law
Manufacturing
Marketing
Military

I.2 Physical Sciences and Engineering

Aerospace
Astronomy
Chemistry
Earth and atmospheric sciences
Electronics
Engineering
Mathematics and statistics
Physics

I.3 Life and Medical Sciences

Biology
Health
Medical information systems

I.4 Social and Behavioral Sciences

Economics
Psychology
Sociology

I.5 Arts and Humanities

Arts, fine and performing
Language translation
Linguistics
Literature
Music

I.6 Computer-Aided Systems

Computer-aided design (CAD)
Computer-aided manufacturing (CAM)

I.7 Computers in Other Systems

Command and control
Consumer products
Industrial control
Military
Process control
Real time

I.8 Miscellaneous

J. COMPUTING MILIEUX

J.0 General

J.1 The Computer Industry

Markets
Standards
Statistics
Suppliers

J.2 History of Computing

Hardware
People
Software
Systems

J.3 Computers and Education

J.3.0 General

J.3.1 Computer Uses in Education

Computer-assisted instruction (CAI)
Computer-managed instruction (CMI)

J.3.2 Computer and Information Science Education

Computer science education
Curriculum
Information systems education
Self-assessment

J.3.3 Miscellaneous

Accreditation
Computer literacy

J.4 Computers and Society

J.4.0 General

J.4.1 Public Policy Issues

Privacy
Regulation
Transborder data flow

J.4.2 Social Issues

Automation
Computer crime
Effects in organizations
Government use

J.4.3 Miscellaneous

J.5 Legal Aspects of Computing

J.5.0 General

J.5.1 Software Protection

Copyrights
Patents
Proprietary rights
Trade secrets

J.5.2 Governmental Issues

Regulation
Taxation

J.5.3 Miscellaneous

Contracts
Hardware patents

J.6 Management of Computing and Information Systems

J.6.0 General

J.6.1 Project and People Management

Life cycle
PERT/CPM
Staffing
Systems analysis and design
Systems development
Training

J.6.2 Installation Management

Benchmarks
Computer selection
Computing equipment management
Performance measurement
Pricing

J.6.3 Software Management

Software development
Software selection

J.6.4 Miscellaneous

Insurance
Security

J.7 The Computing Profession

J.7.0 General

J.7.1 Occupations

J.7.2 Organizations

J.7.3 Testing, Certification, and Licensing

J.7.4 Miscellaneous

Codes of good practice
Ethics

J.8 Miscellaneous

K. GENERAL LITERATURE

K.0 General

K.1 Introductory and Survey

K.2 Reference (e.g., dictionaries, encyclopedias, glossaries)

K.3 Miscellaneous