ICSE: Years 1976 to 1984

Structured analysis for Requirements Definition (abstract only)
Douglas T. Ross, Kenneth E. Schoman, Jr.
Page: 1

PSL/PSA a computer-aided technique for structured documentation and analysis of information processing systems
Daniel Teichroew, Ernest Allen Hershey, III
Page: 2

Automated software engineering through structured data management
C. A. Irvine, J. W. Brackett
Page: 3

The design of the PSI program synthesis system
Cordell Green
Pages: 4 - 18

Observations on the interaction between coding and efficiency knowledge in the PSI Program Synthesis System
David R. Barstow, Elaine Kant
Pages: 19 - 31

Is `sometime’ sometimes better than `always’?: Intermittent assertions in proving program correctness
Zohar Manna, Richard Waldinger
Pages: 32 - 39

Verification of fairness in an implementation of monitors
Richard Alan Karp, David C. Luckham
Pages: 40 - 46

Signaling in monitors
John H. Howard
Pages: 47 - 52

Verification of a monitor specification
Ashok R. Saxena, Thomas H. Bredt
Pages: 53 - 59

The Software Development System
C. G. Davis, C. R. Vick
Page: 60
Software requirements: Are they really a problem?
T. E. Bell, T. A. Thayer
Pages: 61 - 68

A requirements engineering methodology for real-time processing requirements
Mack Alford
Page: 69

An extendable approach to computer-aided software requirements engineering
Thomas E. Bell, David C. Bixler, Margaret E. Dyer
Page: 70

Specifications a key to effective software development
P. C. Belford, A. F. Bond, D. G. Henderson, L. S. Sellers
Pages: 71 - 79

Process Design engineering a Methodology for Real-time Software development
S. N. Gaulding, J. D. Lawson
Pages: 80 - 85

Process design system an integrated set of software development tools
R. G. Koppang
Pages: 86 - 90

A methodology for decomposing system requirements into data processing requirements
Kenneth G. Salter
Pages: 91 - 101

Adaptive testing
Dennis W. Cooper
Pages: 102 - 105

A laboratory for the development and evaluation of BMD software quality enhancement techniques
Jeoffrey P. Benson, Richard A. Melton
Pages: 106 - 109

Research towards a technology to support the specification of data processing system performance requirements
Edward E. Balkovich, George P. Engelberg
Essential elements of software engineering education
Peter Freeman, Anthony I. Wasserman, Richard E. Fairley
Pages: 116 - 122

Design and specification of the minimal subset of an operating system family (Abstract only)
D. L. Parnas, G. Handzel, H. Würges
Page: 123

Design and verification of real-time systems
Jorge V. Phillips, Thomas H. Bredt
Pages: 124 - 131

RIG, rochester’s intelligent gateway: System overview
E. Ball, J. Feldman, J. Low, R. Rashid, P. Rovner
Page: 132

The structure and characteristics of distributed systems
C. V. Ravi
Pages: 133 - 137

A critical overview of computer performance evaluation
J. C. Browne
Pages: 138 - 145

An experiment in program restructuring for performance enhancement
Domenico Ferrari, Edwin Lau
Pages: 146 - 150

A data structure and drive mechanism for a table-driven simulation system employing multilevel structural representations of digital systems
N. Billawala, S. A. Szygenda, E. W. Thompson
Pages: 151 - 157

Automatic program analysis and evaluation
Marvin V. Zeikowitz
Pages: 158 - 163

An introduction to the Programmer’s Workbench
T. A. Dolotta, J. R. Mashey
Pages: 164 - 168
Using a command language as a high-level programming language
J. R. Mashey
Pages: 169 - 176

Documentation tools and techniques
J. R. Mashey, D. W. Smith
Pages: 177 - 181

The LEAP load and test driver
T. A. Dolotta, J. S. Licwinko, R. E. Menninger, W. D. Roome
Pages: 182 - 186

A Modification Request Control System
D. B. Knudsen, A. Barofsky, L. R. Satz
Pages: 187 - 192

A user's viewpoint on the Programmer's Workbench
M. H. Bianchi, J. L. Wood
Pages: 193 - 199

An illustration of current ideas on the derivation of correctness proofs and correct programs
David Gries
Page: 200

The role of software in successful computer applications
W. Wayne Black
Pages: 201 - 205

The preliminary design as a key to successful software development
Jack M. Dreyfus, Peter J. Karacsony
Pages: 206 - 213

Towards an engineering approach to software design
John R. White, Taylor L. Booth
Pages: 214 - 222

Programming as an evolutionary process
F. L. Bauer
Pages: 223 - 234

A language extension for controlling access to shared data
Anita K. Jones, Barbara H. Liskov
Extending Concurrent Pascal to allow dynamic resource management
A. Silberschatz, R. B. Kieburtz, A. Bernstein
Page: 236

New languages from old: The extension of programming languages by embedding, with a case study
Michael B. Feldman
Pages: 237 - 242

Aspects of a trigger subsystem in an integrated database system
Kapali P. Eswaran
Pages: 243 - 250

A model of roll-back recovery with multiple checkpoints
Erol Gelenbe
Pages: 251 - 255

On the modelling, analysis and design of protocols - a special class of software structures
Mohamed G. Gouda, Eric G. Manning
Pages: 256 - 262

On reliable topologies for computer networks
Izhak Rubin
Pages: 263 - 267

Structural models for software reliability prediction
Martin L. Shooman
Pages: 268 - 280

Bayesian models of design based on intuition
K. M. Chandy
Pages: 281 - 285

On the effectiveness of set associative page mapping and its application to main memory management
Alan Jay Smith
Pages: 286 - 292

Formal specification and automatic programming
Mehdi Jazayeri
Pages: 293 - 296
Introducing a Software Design Language
Yaohan Chu
Pages: 297 - 304

A management approach to the development of computer-based systems
R. Turn, M. R. Davis, R. N. Reinstedt
Pages: 305 - 311

An analysis of the resources used in the SAFEGUARD system software development
W. E. Stephenson
Pages: 312 - 321

Research paradigms in computer science
Peter Wegner
Pages: 322 - 330

Maxims for malfeasant designers, or how to design languages to make programming as difficult as possible
Richard L. Wexelblat
Pages: 331 - 336

On the Transformational Implementation approach to programming
Robert Balzer, Neil Goldman, David Wile
Pages: 337 - 344

Decision table programming and reliability
Art Lew, Doris Tamanaha
Pages: 345 - 349

Program evolution and its impact on software engineering
M. M. Lehman, F. N. Parr
Pages: 350 - 357

A defense view of software engineering
William A. Whitaker
Pages: 358 - 362

Software engineering in the development of the TRIDENT Fire Control System
R. M. Pollock, W. L. McCoy
Pages: 363 - 369
Structured Programming: From theory to practice
John C. Carrow
Pages: 370 - 372

Dynamic software engineering: An evolutionary approach to automated software development and management
Joseph S. Greene, Jr.
Pages: 373 - 378

Software research in the Department of Defense
William E. Carlson
Pages: 379 - 383

Sacrificing the calf of flexibility on the altar of reliability
Peter J. Denning
Pages: 384 - 386

Software engineering and structured programming
Maurice V. Wilkes
Page: 387

Software development
H. D. Mills
Page: 388

Induction as the basis for program verification
Charles Reynolds, Raymond Yeh
Page: 389

An introduction to the construction and verification of Alphard programs
Wm. A. Wulf, Ralph L. London, Mary Shaw
Page: 390

Control structure abstractions of the backtracking programming technique
Susan L. Gerhart, Lawrence Yelowitz
Page: 391

A generalized assertion language
Tsun S. Chow
Pages: 392 - 399

Some classes of naturally provable programs
S. K. Basu, J. Misra
A complexity measure
Thomas J. McCabe
Page: 407

The design of a template structure for a generalized data structure definition facility
Billy G. Claybrook
Pages: 408 - 413

The design of data type specifications
John V. Guttag, Ellis Horowitz, David R. Musser
Pages: 414 - 420

Software development and proofs of multi-level security
Peter G. Neumann, Richard J. Feiertag, Karl N. Levitt, Lawrence Robinson
Pages: 421 - 428

An approach to error-resistant software design
S. S. Yau, R. C. Cheung, D. C. Cochrane
Pages: 429 - 436

Response to undesired events in software systems
D. L. Parnas, H. Würges
Pages: 437 - 446

Recovery blocks in action: A system supporting high reliability
T. Anderson, R. Kerr
Pages: 447 - 457

The design, analysis, and verification of the SIFT fault tolerant system
John H. Wensley, Milton W. Green, Karl N. Levitt, Robert E. Shostak
Pages: 458 - 469

How to design a system in which modules can be changed on the fly
R. S. Fabry
Pages: 470 - 476

Test procedures: A new approach to software verification
David J. Panzl
Pages: 477 - 485
A software reliability assessment based on a structural and behavioral analysis of programs
Simone Pimont, Jean-Claude Rault
Pages: 486 - 491

The dimensions of maintenance
E. Burton Swanson
Pages: 492 - 497

A framework for data base semantic integrity
Michael Hammer, Dennis McLeod
Pages: 498 - 504

A Synthetic English query language for a relational associative processor
L. Kerschberg, E. A. Ozkarahan, J. E.S. Pacheco
Pages: 505 - 519

The translation and compatibility of SEQUEL and Query by Example
Dennis McLeod
Pages: 520 - 526

A comparison of the use of links and secondary indices in a relational data base system
Michael Stonebraker
Pages: 527 - 531

A software engineering experience in the management, design and implementation of a data secure system
David K. Hsiao
Pages: 532 - 538

Application of clustering to estimate missing data and improve data integrity
R. C. T. Lee, J. R. Slagle, C. T. Mong
Pages: 539 - 544

Some considerations in database application programming
Isao Miyamoto
Pages: 545 - 555

Improved updating in relational data base systems by deuter-sphere algorithms
Peter Kümmel
Pages: 556 - 561
A principle for resilient sharing of distributed resources
Peter A. Alsberg, John D. Day
Pages: 562 - 570

Design evaluation of the compiler generating system MUG1
Reinhard Wilhelm, Knut Ripken, Joachim Ciesinger, Harald Ganzinger, Walter Lahner, Rolf Nollmann
Pages: 571 - 576

An environmental simulator for the FDNY computer aided dispatch system
John Mohan
Pages: 577 - 584

Data flow, abstraction levels and specifications for communications switching systems
K. S. Shankar, C. S. Chandersekaran
Pages: 585 - 591

Quantitative evaluation of software quality
B. W. Boehm, J. R. Brown, M. Lipow
Pages: 592 - 605

Aspects of design, test and validation of the software for a computerized reactor protection system
U. Voges
Pages: 606 - 610

Program analysis - A method for the verification of software for the control of a nuclear reactor
W. Ehrenberger, G. Rauch, K. Okroy
Pages: 611 - 616

A system for automatic software evaluation
B. C. Hodges, J. P. Ryan
Pages: 617 - 623

The detection of anomalous interprocedural data flow
Lloyd D. Fosdick, Leon J. Osterweil
Pages: 624 - 628

ATLAS-An Automated Software Testing System
W. H. Jessop, J. R. Kane, S. Roy, J. M. Scanlon
Pages: 629 - 635
On the automated generation of program test data (Abstract only)
C. V. Ramamoorthy, S. F. Ho, W. T. Chen
Page: 636

Software engineering: A keynote address
C. A. R. Hoare
Pages: 1 - 4

Control structure aptness: A case study using top-down parsing
Gary Lindstrom
Pages: 5 - 12

Behavior modelling during software design
William E. Riddle, Jack C. Wileden, John H. Sayler, Alan R. Segal, Allan M. Stavely
Pages: 13 - 22

A type-checking program linkage system for pascal
R. B. Kieburitz, W. Barabash, C. R. Hill
Pages: 23 - 28

Measuring reliability of computation center software
Patricia A. Hamilton, John D. Musa
Pages: 29 - 36

How to measure software reliability, and how not to
B. Littlewood
Pages: 37 - 45

Toward an effective software reliability evaluation
Isao Miyamoto
Pages: 46 - 55

A proposed curriculum for software engineering education
Peter Freeman, Anthony I. Wasserman
Pages: 56 - 62

A model of software engineering
Lawrence J. Peters, Leonard L. Tripp
Pages: 63 - 70

Tutorial session 3B: Prospects for program verification
James C. King
Page: 71

Information systems: Modelling, sequencing and transformations
M M Jackson
Pages: 72 - 81

The interaction between the preliminary designs and the technical requirements for the DoD common high order language
David A. Fisher
Pages: 82 - 83
How software is really engineered?
Leon G. Stucki
Page: 84

Kongsberg's road to an industrial software methodology
Jan T Pedersen, John K Buckle
Pages: 85 - 93

Introduction to the WELLMADE design methodology
Donald L. Boyd, Antonio Pizzarello
Pages: 94 - 100

Requirements expression and verification aid
Sharon A. Stephens, Leonard L. Tripp
Pages: 101 - 108

DAS - an automated system to support design analysis
R. R. Willis
Pages: 109 - 115

Analyzing medium-scale software development
Victor R. Basili, Marvin V. Zelkowitz
Pages: 116 - 123

Software engineering in avionics applications
A. Dniestrowski, J. M. Guillaume, R. Mortier
Pages: 124 - 131

The design of a reliable applications system
Edward Gainer, Edward Kimball, Alan Kortesoja, Alfred Maley
Pages: 132 - 135

A graph theoretic approach to the verification of program structures
John R. Brown, Kurt F. Fischer
Pages: 136 - 141

FAST: A second generation program analysis system
J. C. Browne, David B. Johnson
Pages: 142 - 148

A model for program complexity analysis
Carma L. McClure
Pages: 149 - 157

Inference rules for program annotation
Nachum Dershowitz, Zohar Manna
Pages: 158 - 167

Design and verification of communication procedures: A bottom-up approach
P. Azema, J. M. Ayache, B. Berthomieu
Pages: 168 - 174

The synthesis of structure-changing programs
Zohar Manna, Richard Waldinger
Pages: 175 - 187

User-perceived quality of interactive systems
W. Dzida, S. Herda, W. D. Itzfeldt
Pages: 188 - 195

Using a behavioral theory of program comprehension in software engineering
Ruven Brooks
Pages: 196 - 201

Analysis of error remediation expenditures during validation
Mary Anne Herndon, Ann P. Keenan
Pages: 202 - 206

Design and principles of a fault tolerant system
MM. L. Boi, P. Michel
Pages: 207 - 214

Use of state diagrams to engineer communications software
Paul Chung, Barry Gaiman
Pages: 215 - 221

An engineering methodology for presenting software functional architecture
Leighton R Scott
Pages: 222 - 229

Concurrent software system design supported by SARA at the age of one
Ivan M. Campos, Gerald Estrin
Pages: 230 - 242

Research directions in software technology
Peter Wegner
Pages: 243 - 259

A perspective on software development
Bill Hetzel
Pages: 260 - 263

Designing software for ease of extension and contraction
David L. Parnas
Pages: 264 - 277

The Multics system programming process
T. H. Van Vleck, C. T. Clingen
Pages: 278 - 280

Measuring memory protection
Douglas Cook
Pages: 281 - 287

An extensible file system for hydra
Guy Almes, George Robertson  
Pages: 288 - 294

Dynamic restructuring in an experimental operating system  
Hannes Goullon, Rainer Isle, Klaus-Peter Löhr  
Pages: 295 - 304

Theoretical and empirical studies of program testing  
William E. Howden  
Pages: 305 - 311

Symbolic interpretation and tracing of PASCAL-programs  
Guido Persch, Georg Winterstein  
Pages: 312 - 319

Automatic revision of formal test procedures  
David J. Panzl  
Pages: 320 - 326

Some experience in building portable software  
Max Stern  
Pages: 327 - 332

Evolution in the design of abstract machines for software portability  
Daniel Thalmann  
Pages: 333 - 341

Toward the engineering of software (Panel Discussion): Problems of the 1980's  
Anthony I. Wasserman, Laszlo A. Belady, Philip Enslow, Jr., C. A. R. Hoare, William A. Wulf  
Page: 341

Software engineering in 1968  
B. Randell  
Pages: 1 - 10

Software engineering-as it is  
Barry W. Boehm  
Pages: 11 - 21

Module structure in an evolving family of real time systems  
D. M. Lasker  
Pages: 22 - 28

Software development control based on module interconnection  
Walter F. Tichy  
Pages: 29 - 41

A recovery mechanism for modular software  
Flaviu Cristian  
Pages: 42 - 50.A
Education, training and technology transfer in software engineering
(Panl Discussion)
L. A. Belady, C. Floyd, I. Stinson, T. Straeter, A. Wasserman
Page: 51

Software engineering tools: Past, present and future
Charles R. Vick
Page: 52

A system for program refinement
Thomas E. Cheatham, Jr., Judy A. Townley, Glenn H. Holloway
Pages: 53 - 62

Pragmatic techniques for program analysis and verification
Erhard Ploedereder
Pages: 63 - 72

A new technique for improving the quality of computer programs
R. S. Scowen
Pages: 73 - 78

Computer-aided micro-analysis of programs
Jacques Cohen
Pages: 79 - 84

Central flow control software development: A case study of the
effectiveness of software engineering techniques
Peter C. Belford, Richard A. Berg, Thomas L. Hannan
Pages: 85 - 93

Pilot: A software engineering case study
Thomas R. Horsley, William C. Lynch
Pages: 94 - 99

PEARL at the age of three: Case study of development and application of a
common high order real time programming language
T. Martin
Pages: 100 - 109

Applicative programming
R. M. Burstall
Page: 110

The computer assisted software engineering (CASE) system
William Scott Amey
Pages: 111 - 115

Computer aided design of software systems
R. R. Willis, E. P. Jensen
Pages: 116 - 125

CORE - a method for controlled requirement specification
G. P. Mullery
Pages: 126 - 135
CADES - software engineering in practice
R. W. McGuffin, A. E. Elliston, B. R. Tranter, P. N. Westmacott
Pages: 136 - 144

An experiment in software engineering: The Architecture Research Facility as a case study
Honey S. Elovitz
Pages: 145 - 152

The project library - a tool for software development
Ernst Denert
Pages: 153 - 163

On correct program development
Andrzej Blikle
Pages: 164 - 173

The impact of mesa on system design
Hugh C. Lauer, Edwin H. Satterthwaite
Pages: 174 - 182

Invariants for specifications
J. Kramer, R. J. Cunningham
Pages: 183 - 193

A constructive approach to reliable synchronization code
Mark S. Laventhal
Pages: 194 - 202

Specification and implementation of parallel activities on abstract objects
S. Keramidis, L. Mackert
Pages: 203 - 211

Path expressions in pascal
Roy H. Campbell, Robert B. Kolstad
Pages: 212 - 219

An approach towards reliable software
J. C. Rault
Pages: 220 - 230

A systematic approach to the development and validation of critical software for nuclear power plants
C. V. Ramamoorthy, F. B. Bastani, J. M. Favaro, Y. R. Mok, C. W. Nam, K. Suzuki
Pages: 231 - 240

An interactive FORTRAN structuring aid
Julian E. Gomez
Pages: 241 - 244
EPOS - a specification and design technique for computer controlled real-time automation systems
Joachim Biewald, Peter Goehner, Rudolf Lauber, Helmut Schelling
Pages: 245 - 250

A flexible environment for program development based on a symbolic interpreter
Pages: 251 - 263

Transformation techniques; constructive or destructive? (Panel Session)
M. A. Jackson
Page: 264

Anomaly detection in concurrent programs
G. Bristow, C. Drey, B. Edwards, W. Riddle
Pages: 265 - 273

Hierarchical coroutines a mechanism for improved program structure
Leonard I. Vanek, Rudolf Marty
Pages: 274 - 285

A system to improve incorrect programs
Harald Wertz
Pages: 286 - 293

The use of a Module Interconnection Language in the SARA system design methodology
Maria Heloisa Penedo, Daniel M. Berry
Pages: 294 - 307

Language features for description of cooperating processes
J. P. Banatre, M. Banatre
Pages: 308 - 314

TASK forces: Distributed software for solving problems of substantial size
Anita K. Jones, Karsten Schwans
Pages: 315 - 330

Micro chaos in the 1980's - can we prevent it?
Gerald Estrin, Ernico Anceau, Gina Granata, Kenneth Hales, Jan Witt
Page: 331

Microprocessors: A platform for true program portability with examples from MicroCobol
A. d'Agapyeff
Pages: 332 - 339

The role of software engineering in the microcomputer revolution: An overview
W. M. Mckeeeman
Page: 340
Prediction and management of program quality
H. Remus, S. Zilles
Pages: 341 - 350

Some results from an empirical study of computer software
Alan R. Feuer, Edward B. Fowlkes
Pages: 351 - 355

Third time charm: Stronger prediction of programmer performance by software complexity metrics
Bill Curtis, Sylvia B. Sheppard, Phil Milliman
Pages: 356 - 360

Software engineering: Management, personnel and methodology
Thomas H. Bruggere
Pages: 361 - 368

An inter-organisational comparison of programming productivity
D. R. Jeffery, M. J. Lawrence
Pages: 369 - 377

An analysis of software project failure
Joichi Abe, Ken Sakamura, Hideo Aiso
Pages: 378 - 385

An environment for producing well-engineered microcomputer software
R. Sterling Eanes, Carl K. Hitchon, Richard M. Thall, John W. Brackett
Pages: 386 - 398

Cross software development for microprocessors using a translator writing system
Massimo Ancona, Gabriella Dodero, Ercole Luigi Durante
Pages: 399 - 402

APL2M-a general computer-aided software engineering environment for mini and microcomputers
Jean Rohmer
Pages: 403 - 411

Software engineering...an economic perspective
J. H. Morrissey, L. S.-Y. Wu
Pages: 412 - 422

Synchronization and recovery in database systems
Rudolf Bayer
Page: 423

Is software development manageable?
Page: 424

Test plan generation using formal grammars
Finite semantics: A technique for program testing
J. Hořejš
Pages: 433 - 440

Software tools - are we on the right track?
William E. Riddle
Page: 441

My hopes of computing science (EWD709)
Edsger W. Dijkstra
Pages: 442 - 448

Look ahead at software engineering
Wladyslaw M. Turski
Pages: 449 - 456

Lilith: A personal computer for the software engineer
N. Wirth
Pages: 2 - 15

Implementing a software management discipline
R. Loesh, B. Larman, P. Molko, D. Reifer
Page: 17

Experience with application of modern software management controls
Donald L. Paster
Pages: 18 - 26

Early experiences with Euclid
David B. Wortman, James R. Cordy
Pages: 27 - 32

Conspectus of software engineering environments
Hans-Ludwig Hausen, Monika Müllerburg
Pages: 34 - 43

An Incremental Programming Environment
Peter H. Feiler, Raul Medina-Mora
Pages: 44 - 53

A database model for effective configuration management in the programming environment
Karen E. Huff
Pages: 54 - 61

A robust B-tree implementation
J. P. Black, D. J. Taylor, D. E. Morgan
Pages: 63 - 70

An index organization for applications with highly skewed access patterns
Christer Hultén
The evolution of programs: Program abstraction and instantiation
Nachum Dershowitz
Pages: 79 - 88

A comparison of lifecycle models
Pentti Kerola, Peter Freeman
Pages: 90 - 99

Program complexity measure for software development management
Takeshi Sunohara, Akira Takano, Kenji Uehara, Tsutomu Ohkawa
Pages: 100 - 106

A meta-model for software development resource expenditures
John W. Bailey, Victor R. Basili
Pages: 107 - 116

Use of cluster analysis to evaluate software engineering methodologies
Eric Chen, Marvin V. Zelkowitz
Pages: 117 - 123

An algorithm to support code-skeleton generation for concurrent systems
Maria Heloisa Penedo, Daniel M Berry, Gerald Estrin
Pages: 125 - 135

Experience with a modular typed language: PROTEL
P. M. Cashin, M. L. Joliat, R. F. Kamel, D. M. Lasker
Pages: 136 - 143

Software architecture based on communicating residential environments
Erik Sandewall, Claes Strömberg, Henrik Sörensen
Pages: 144 - 152

Interactive software development tool: ISDT
M. Azuma, M. Takahashi, S. Kamiya, K. Minomura
Pages: 153 - 162

On the time overhead of counters and traversal markers
Ira R. Forman
Pages: 164 - 169

Using attributed grammars to test designs and implementations
A. G. Duncan, J. S. Hutchison
Pages: 170 - 178

A report on random testing
Joe W. Duran, Simeon Ntafos
Pages: 179 - 183

Sufficient test sets for path analysis testing strategies
Steven J. Zeil, Lee J. White
Pages: 184 - 191
A procedure for designing abstract interfaces for device interface modules
Kathryn Heninger Britton, R. Alan Parker, David L. Parnas
Pages: 195 - 204

The effects of symbology and spatial arrangement on the comprehension of software specifications
Sylvia B. Sheppard, Elizabeth Kruesi, Bill Curtis
Pages: 207 - 214

The effect of modularization and comments on program comprehension
S. N. Woodfield, H. E. Dunsmore, V. Y. Shen
Pages: 215 - 223

Methods for improving controlled experimentation in software engineering
Thomas Moher, G. Michael Schneider
Pages: 224 - 233

Completeness criteria for testing elementary program functions
William E. Howden
Pages: 235 - 243

A partition analysis method to increase program reliability
Debra J. Richardson, Lori A. Clarke
Pages: 244 - 253

An automated program testing methodology and its implementation
Dorothy M. Andrews, Jeoffrey P. Benson
Pages: 254 - 261

Resource = abstract data type + synchronization - A methodology for message oriented programming -
P. R.F. Cunha, T. S.E. Maibaum
Pages: 263 - 272

The finalization operation for abstract types
Richard L. Schwartz, P. M. Melliar-Smith
Pages: 273 - 282

A requirements and design aid for relational data bases
Max L. Wilson
Pages: 283 - 293

Executable requirements for embedded systems
Pamela Zave, Raymond T. Yeh
Pages: 295 - 304

An innovative approach to system requirements analysis by using structural modeling method
Norihsa Komoda, Koichi Haruna, Hiroyuki Kaji, Hiroshi Shinozawa
Pages: 305 - 313

Evaluation of a software requirements document by analysis of change data
Victor R. Basili, David M. Weiss
Development of computer programs by problem analysis Diagram(PAD)
Y. Futamura, T. Kawai, H. Horikoshi, M. Tsutsumi
Pages: 325 - 332

Prototyping as a tool in the specification of user requirements
Hassan Gomaa, Douglas B.H. Scott
Pages: 333 - 342

HFP: A hierarchical and functional programming based on attribute grammar
Takuya Katayama
Pages: 343 - 352

Active data structures
Gregory R. Andrews, David P. Dobkin
Pages: 354 - 362

The two-step commitment protocol: Modeling, specification and proof methodology
J. L. Baer, G. Gardarin, C. Girault, G. Roucairol
Pages: 363 - 373

Specifying and proving properties of sentinel processes
Krithivasan Ramamritham, Robert M. Keller
Pages: 374 - 382

User Software Engineering and the design of interactive systems
Anthony I. Wasserman
Pages: 387 - 393

A software methodology for building interactive tools
Robert W. Lingard
Pages: 394 - 399

A user interface for online assistance
Nathan Relles, Lynne A. Price
Pages: 400 - 408

BIGMAC II: A FORTRAN language augmentation tool
Eugene W. Myers, Jr., Leon J. Osterweil
Pages: 410 - 421

Beyond ALBE/P: Language neutral form
J. W. Lewis
Pages: 422 - 429

Program refinement by transformation
Thomas E. Cheatham, Jr., Glenn H. Holloway, Judy A. Townley
Pages: 430 - 437

Program slicing
Mark Weiser
Pages: 439 - 449
Application downloading
Robert Balzer, Alvin Cooperband, Martin Feather, Philip London, David Wile
Pages: 450 - 459

An interactive tool for program manipulation
Anne Adam, Paul Gloess, Jean-Pierre Laurent
Pages: 460 - 468

Overstructured management of software engineering
Gerald M. Weinberg
Pages: 2 - 8

The dimensions of healthy maintenance
Robert S. Arnold, Donald A. Parker
Pages: 10 - 27

MAP: A tool for understanding software
Sally Warren
Pages: 28 - 37

Design considerations in language processing tools for Ada
W. Babich, L. Weissman, M. Wolfe
Pages: 40 - 47

A closer look at iteration: The self stabilizing capability of loops
Ali Mili
Pages: 48 - 56

Design, implementation, and evaluation of a Revision Control System
Walter F. Tichy
Pages: 58 - 67

Configuration control for evolutive software products
Osamu Shigo, Yoshio Wada, Yuichi Terashima, Kanji Iwamoto, Takashi Nishimura
Pages: 68 - 75

From specifications to machine code: Program construction through formal reasoning
Friedrich L. Bauer
Pages: 84 - 91

Experiments with computer software complexity and reliability
D. Potier, J. L. Albin, R. Ferreol, A. Bilodeau
Pages: 94 - 103

A model for estimating program size and its evaluation
Minoru Itakura, Akio Takayanagi
Pages: 104 - 109

Experimental results on the paging behavior of numerical programs
W. Abu-Sufah, R. Lee, M. Malkawi, P. Yew
Verification system for formal requirements description
Kiyoshi Agusa, Atsushi Ohnishi, Yutaka Ohno
Pages: 120 - 126

Requirement specification description system in Japanese language-JISDOS-
Jun Murai, Nobuo Saito, Norihisa Doi, Masayuki Morohashi, Tetunosuke Fujisaki
Pages: 127 - 136

Static and dynamic data modeling for information system design
D. Marca, C. McGowan
Pages: 137 - 146

The TRW Software Productivity System
Barry W. Boehm, James F. Elwell, Arthur B. Pyster, E. Donald Stuckle,
Robert D. Williams
Pages: 148 - 156

The design of an integrated, interactive and incremental programming environment
Harald Wertz
Pages: 157 - 165

Toolpack - an experimental software development environment research project
Leon J. Osterweil
Pages: 166 - 175

An Insider's Survey on Software Development
Jean C. Zolnowski, Peter D. Ting
Pages: 178 - 187

An examination of evolution dynamics
M. J. Lawrence
Pages: 188 - 196

M.H. Halstead's Software Science - a critical examination
Peter G. Hamer, Gillian D. Frewin
Pages: 197 - 206

Functional specification of synchronized processes based on modal logic
Naoki Yonezaki, Takao Katayama
Pages: 208 - 217

Specification of abstract data types with partially defined operations
Toshiki Sakabe, Yasuyoshi Inagaki, Namio Honda
Pages: 218 - 224

Capturing more world knowledge in the requirements specification
Sol J. Greenspan, John Mylopoulos, Alex Borgida
Pages: 225 - 234
Consistency checking within embedded design languages
A. Rudmik, B. E. Casey, H. Cohen
Pages: 236 - 245

Experience with a Module Package in developing production quality PASCAL programs
Sally Warren, Bruce E. Martin, Charles Hoch
Pages: 246 - 253

A form-based approach to human engineering methodologies
H. C. Kuo, C. H. Li, J. Ramanathan
Pages: 254 - 263

Data flow analysis techniques for test data selection
Sandra Rapps, Elaine J. Weyuker
Pages: 272 - 278

No. 1A ESS Laboratory Support System - erasable flag facility
Donald V. Buyansky, James W. Schatz
Pages: 279 - 286

Software quality &equil; test accuracy × test coverage
M. Ohba
Pages: 287 - 293

Designing data entry programs using state diagram as a common model
Moses M. Ling
Pages: 296 - 308

Data-driven implementation of data flow diagrams
Robert G. Babb, II
Pages: 309 - 318

Automatic input of flow chart in document image
S. Ito
Pages: 319 - 328

An editor for documentation in &pgr;-system to support software development and maintenance
Yukikazu Nakamoto, Tadahiro Iwamoto, Masato Hori, Kenichi Hagihara, Nobuki Tokura
Pages: 330 - 339

XS-1: An integrated interactive system and its kernel
Pages: 340 - 349

DUAL: An interactive tool for developing documented programs by step-wise refinements.
Luigi Petrone, Antonio Di Leva, Franco Sirovich
Pages: 350 - 357

An automatic programming system to support an experimental science
D. Barstow, R. Duffey, S. Smoliar, S. Vestal  
Pages: 360 - 366

Representation of factual information by equations and their evaluation  
Peter Lucas, Tore Risch  
Pages: 367 - 376

A retrospective on the development of Star  
Eric Harslem, LeRoy E. Nelson  
Pages: 377 - 383

Global data flow analysis by decomposition into primes  
Ira R. Forman  
Pages: 386 - 392

A hierarchical structuring method for functional software systems  
Kokichi Futatsugi, Koji Okada  
Pages: 393 - 402

A new design language for communication protocols and a systematic design method of communication systems  
Norio Shiratori, Junichi Gohara, Shoichi Noguchi  
Pages: 403 - 412

Visual aid for FORTRAN program debugging  
K. Takahashi, T. Asò, M. Kobayashi  
Pages: 414 - 415

The Inspection Method applied to small projects  
B. Runge  
Pages: 416 - 417

Software development with executable functional specifications  
Joseph E. Urban  
Pages: 418 - 419

The portable communication protocol program COMPAS for data terminal systems  
K. Miyazawa, M. Oda, S. Kato  
Pages: 420 - 421

Early experiences with a multi-display programming environment  
Yoshihisa Mano, Kazuhito Omaki, Koji Torii  
Pages: 422 - 423

A model for description of communication protocol  
Kazuhiko Chiba, Kazunori Konishi, Akira Kurematsu  
Pages: 424 - 425

Session 1A (Panel Session): Panel— the national software engineering projects  
J. Marciniak, K. Fuchi, P. Hughes, H. Hünke, E. Lieblein, J. C. Rault  
Page: 1
Session 2A: Presentation—software development paradigm
R. Balzer, F. L. Bauer, T. E. Cheatham
Page: 3

Session 2B: Error processing
R. Kemmerer
Page: 5

Error recovery in systems of communicating processes.
A. Ciuffoletti
Pages: 6 - 17

Exception handling: Formal specification and systematic program construction
M. Bidoit, B. Biebow, M-C. Gaudel, C. Gresse, G. Guiho
Pages: 18 - 29

Diagnostic system for distributed software: A relational database approach
Kiwon Chong, Pei Hsia
Pages: 30 - 40

Session 3A: Debate—software development paradigms
Page: 41

Session 3B: Distributed systems
S. Budkowski
Page: 43

Multifaceted distributed systems specification using processes and event synchronization
Gruia-Catalin Roman, Mark S. Day
Pages: 44 - 55

EDMAS: A locally distributed mail system
Guy Almes, Andrew Black, Carl Bunje, Douglas Wiebe
Pages: 56 - 66

Resource controller tasks in ADA: Their structure and semantics
Krithivasan Ramamritham
Pages: 67 - 76

Session 4A: Management issues
J. H. Frame
Page: 77

A spiral approach to software engineering project management education
Joseph C. Spicer
Pages: 78 - 85

A cooperative approach to software development by application engineers and software engineers
Keiji Uemura, Miki Ohori
Pages: 86 - 96

Fifteen years of psychology in software engineering: Individual differences and cognitive science
Bill Curtis
Pages: 97 - 106

Session 4B: Programming environments I
A. Wasserman
Page: 107

PDAS: An assistant for detailed design and implementation of programs
Tomoharu Mohri, Etsuo Ono, Sanya Uehara, Tetsuyasu Takao, Hideki Sato
Pages: 108 - 115

Steps to an advanced ada programming environment
Richard N. Taylor, Thomas A. Standish
Pages: 116 - 125

Maiday: An environment for guided programming with a definitional language
Jacques Guyard, Jean-Pierre Jacquot
Pages: 126 - 135

Session 5A: Cost and Productivity Measures
B. Boehm
Page: 137

Power: A tool for quantitative evaluation of software project effectiveness
Michael W. Evans, Louis M. Picinich
Pages: 138 - 142

Productivity factors and programming environments
J. Vosburgh, B. Curtis, R. Wolverton, B. Albert, H. Malec, S. Hoben, Y. Liu
Pages: 143 - 152

Programming cost estimate: Is it reasonable?
Robert E. Boydston
Pages: 153 - 159

Session 5B: Programming environments II
R. Fryer
Page: 161

Interactive system for structured program production
H. Maezawa, M. Kobayashi, K. Saito, Y. Futamura
Pages: 162 - 171

A large system evaluation of SREM
Paul A. Scheffer, William E. Rzepka, Albert H. Stone, III
Pages: 172 - 180
A tour through cedar
Warren Teitelman
Pages: 181 - 195

Session 6A: User interfaces
I. Kimura
Page: 197

MGEN - a generator for menu driven programs
Bertil Friman
Pages: 198 - 206

SYNCRO: A dataflow command shell for the lilith/modula computer
Tom DeMarco, Aurel Soceneantu
Pages: 207 - 213

Software engineering for user interfaces
Stephen W. Draper, Donald A. Norman
Pages: 214 - 220

Session 6B: Reliability and complexity measures I
A. Marmor-Squires
Page: 221

Comparison of concurrent software reliability models
Ramadan Moawad
Pages: 222 - 229

A logarithmic poisson execution time model for software reliability measurement
J. D. Musa, K. Okumoto
Pages: 230 - 238

A program complexity metric based on data flow information in control graphs
Kuo-Chung Tai
Pages: 239 - 248

Session 7A: Testing
C. V. Ramamoorthy
Page: 249

An evaluation of required element testing strategies
Simeon C. Ntafos
Pages: 250 - 256

Perturbation testing for computation errors
Steven J. Zeil
Pages: 257 - 265

Assessing a class of software tools
M. A. Hennell, D. Hedley, I. J. Riddell
Pages: 266 - 277
An algebra for data flow anomaly detection
Ira R. Forman
Pages: 278 - 286

Session 7B: Reliability and complexity measures II
P. de Feo
Page: 287

Characteristic program complexity measures
James L. Elshoff
Pages: 288 - 293

A discriminant metric for module cohesion
Thomas J. Emerson
Pages: 294 - 303

Integrated program measurement and documentation tools
Anne Schroeder
Pages: 304 - 313

Quantifying software designs
John Beane, Nancy Giddings, Jon Silverman
Pages: 314 - 322

Session 8A: Editors
I. Miyamoto
Page: 323

Pecan: Program development systems that support multiple views
Steven P. Reiss
Pages: 324 - 333

Table: Object oriented editing of complex structures
Ted J. Biggerstaff, D. Mack Endres, Ira R. Forman
Pages: 334 - 345

Practical applications of a syntax directed program manipulation environment
V Donzeau-Gouge, B. Lang, B. Mélèse
Pages: 346 - 354

Session 8B: Panel—software engineering and social responsibility
S. Gerhart
Page: 355

Session 9A: Knowledge-based software engineering
D. R. Barstow
Page: 357

Knowledge-based communication processes in software engineering
Gerhard Fischer, Matthias Schneider
Pages: 358 - 368
PROUST: Knowledge-based program understanding
W. Lewis Johnson, Elliot Soloway
Pages: 369 - 380

A knowledge base for supporting an intelligent program editor
Daniel G. Shapiro, Jeffrey S. Dean, Brian P. McCune
Pages: 381 - 386

Session 9B: Design methods
R. T. Yeh
Page: 387

The data transform programming method: An example for file processing problems
Pages: 388 - 397

Communication system design using ADA
Arthur G. Duncan, John S. Hutchison, John W. Bailey, Thomas M. Chapman, Andrew Fregly, Elizabeth Kruesi, Thomas McDonald, Dennis Merrill, Sylvia B. Sheppard
Pages: 398 - 407

The modular structure of complex systems
Pages: 408 - 417

Session 10A: Formal specifications
M. Nivat
Page: 419

Abstract models of dialogue concepts
R. Studer
Pages: 420 - 429

Formal specification and development of an ada compiler - a vdm case study
Geert B. Clemmensen, Ole N. Oest
Pages: 430 - 440

Application of axiomatic methods to a specification analyser
Susan L. Gerhart
Pages: 441 - 451

Session 10B: Programming methodologies
M. Zelkowitz
Page: 453

A new program structure to improve accuracy and readability of pascal software
W. J. Rogers
Pages: 454 - 462
Functional programming and logical programming for the telegram analysis problem
Koji Torii, Yuji Sugiyama, Yoshitomi Morisawa, Tadao Kasami
Pages: 463 - 472

Prototyping vs. specifying: A multi-project experiment
Barry W. Boehm, Terence E. Gray, Thomas Seewaldt
Pages: 473 - 484

Session 11A: Verification
C. Ghezzi
Page: 485

Structure charts and program correctness proofs
Shang-Cheng Chyou
Pages: 486 - 498

A system for classifying program verification methods: Assigning meanings to program verification methods
Ali Mili, Jules Desharnais
Pages: 499 - 509

Specification and verification of distributed systems using prolog interpreted petri nets.
Pierre Azema, Guy Juanole, Eric Sanchis, Michel Montbernard
Pages: 510 - 518

Session 11B: Development models and methods
R. Bourgonjon
Page: 519

A method of large-scale software development
M. Hozumi, Y. Sekine, H. Ejima
Pages: 520 - 527

On what exactly is going on when software is developed step-by-step
T. S. E. Maibaum, W. M. Turski
Pages: 528 - 533

Three paradigms for developing information systems
Bruce I. Blum
Pages: 534 - 543

ICSE: Years 1985 to 1993

PRISM—productivity improvement for software engineers and managers
Doug Rosenberg
Pages: 2 - 7

SODOS—a software documentation support environment: its use
Ellis Horowitz, Ronald Williamson
Pages: 8 - 14
Operations for programming in the all
Nazim H. Madhavji
Pages: 15 - 25

``DASOM''—a software engineering tool for communication applications increasing productivity and software quality
Eirik A. M. Vefsnmo
Pages: 26 - 33

Two-way translators based on attribute grammar inversion
Daniel M. Yellin, Eva-Maria M. Mueckstein
Pages: 36 - 42

Database theory for supporting specification-based database system development
David Stemple, Tim Sheard
Pages: 43 - 49

Extending the implementation scheme of functional programming system FP for supporting the formal software development methodology
Qiming Chen
Pages: 50 - 54

Software templates
Dennis M. Volpano, Richard B. Kieburtz
Pages: 55 - 60

Experiments with Prolog design descriptions and tools in CAEDE: an iconic design environment for multitasking, embedded systems
R. J. A. Buhr, C. M. Woodside, G. M. Karam, K. Van Der Loo, D. G. Lewis
Pages: 62 - 67

Continuous execution: the VisiProg environment
Peter Henderson, Mark Weiser
Pages: 68 - 74

Interface control and incremental development in the PIC environment
Alexander L. Wolf, Lori A. Clarke, Jack C. Wileden
Pages: 75 - 82

SPADES—a specification and design system and its graphical interface
J. Ludewig, M. Glinz, H. Huser, G. Matheis, H. Matheis, M. F. Schmidt
Pages: 83 - 89

Ensuring integrity by adding obligations to privileges
Naftaly H. Minsky, Abe D. Lockman
Pages: 92 - 102

The stream machine: a data flow architecture for real-time applications
Paul Barth, Scott Guthery, David Barstow
Pages: 103 - 110

A collection of software tools for analyzing designs of concurrent software systems
Allan M. Stavely, David C. Becker, Susan P. Orr, Gregory B. Titus
Pages: 111 - 118

Discrete event simulation as a means of validating JSD design specifications
Colin Potts, Andy Bartlett, Brian Cherrie, Roy MacLean
Pages: 119 - 125

New software engineering programs—worldwide (panel session)
Laszlo A. Belady, Nico Haberman, Lim Swee Say, John Elmore, W. M. Murray, R. W. Witty, K. Kishida, Fuad Gattaz Sobrinho
Pages: 128 - 130

Active design reviews: principles and practices
David L. Parnas, David M. Weiss
Pages: 132 - 136

Software development process audits—a general procedure
Stewart G. Crawford, M. Hosein Fallah
Pages: 137 - 141

Software quality assessment technology
Toshihiko Sunazuka, Motoei Azuma, Noriko Yamagishi
Pages: 142 - 148

PMDB—a project master database for software engineering environments
Maria H. Penedo, E. Don Stuckle
Pages: 150 - 157

The software knowledge base
Bertrand Meyer
Pages: 158 - 165

The persistent store as an enabling technology for integrated project support environments
Ronald Morrison, Alan Dearle, Peter J. Bailey, Alfred L. Brown, Malcolm P. Atkinson
Pages: 166 - 172

A comparison of cost estimation tools (panel session)
B. Kitchenbaum, Howard A. Rubin, A. Jensen, L. Putnam, P. Rook
Pages: 174 - 180

Building awareness of system testing issues
N. H. Petschenik
Pages: 182 - 188

Software technology maturation
Samuel T. Redwine, Jr., William E. Riddle
Pages: 189 - 200

Design using software engineering principles: overview of an educational program
Michael Schaul
On executable models for rule-based prototyping  
Stanley Lee  
Pages: 210 - 215

Distributed software prototyping with ADS  
James W. Hooper, John T. Ellis, Tony A. Johnson  
Pages: 216 - 223

A technique for prototyping directly from a specification  
R. D. Tavendale  
Pages: 224 - 229

Rapid prototyping of control systems using high level Petri nets  
Giorgio Bruno, Giuseppe Marchetto  
Pages: 230 - 235

Experiments determining best paths for testing computer program predicates  
Lee J. White, Prabhat N. Sahay  
Pages: 238 - 243

A comparison of data flow path selection criteria  
Lori A. Clarke, Andy Podgurski, Debra J. Richardson, Steven J. Zeil  
Pages: 244 - 251

Test case generation using Prolog  
Herbert Pesch, Peter Schnupp, Hans Schaller, Anton Paul Spirk  
Pages: 252 - 258

The causes and effects of infeasible paths in computer programs  
D. Hedley, M. A. Hennell  
Pages: 259 - 266

Programmer performance and the effects of the workplace  
Tom DeMarco, Tim Lister  
Pages: 268 - 272

Software engineering provisioning process  
John H. Manley  
Pages: 273 - 284

A comparison of Lisp, Prolog, and Ada programming productivity in AI area  
F. Hattori, K. Kushima, T. Wasano  
Pages: 285 - 291

COCOMO evaluation and tailoring  
Yukio Miyazaki, Kuniaki Mori  
Pages: 292 - 299

The software process and software environments (panel session)  
Jack C. Wiolated, Mark Dowson  
Pages: 302 - 304
Analyzing the test process using structural coverage
James Ramsey, Victor R. Basili
Pages: 306 - 312

An integrated system for program testing using weak mutation and data flow analysis
M. R. Girgis, M. R. Woodward
Pages: 313 - 319

An interactive debugger for a concurrent language
N. De Francesco, D. Latella, G. Vaglini
Pages: 320 - 325

Program productivity hot topics and hot air (panel session)
T. C. Jones, T. DeMarco
Page: 327

An empirical study of a model for program error prediction
Muneo Takahashi, Yuji Kamayachi
Pages: 330 - 336

An experimental evaluation of simple methods for seeding program errors
John C. Knight, Paul E. Ammann
Pages: 337 - 342

Completely monotone regression estimates of software failure rates
Douglas R. Miller, Ariela Sofer
Pages: 343 - 348

Automatic tuning of multi-task programs for real-time embedded systems
Toru Shimizu, Ken Sakamura
Pages: 350 - 357

PROMPTER: a knowledge based support tool for code understanding
Koichi Fukunaga
Pages: 358 - 363

The Analyst—a workstation for analysis and design
Mark Stephens, Ken Whitehead
Pages: 364 - 369

Criteria for software modularization
David N. Card, Gerald T. Page, Frank E. McGarry
Pages: 372 - 377

A validation of software metrics using many metrics and two resources
Dennis Kafura, James Canning
Pages: 378 - 385

Calculation and use of an environment's characteristic software metric set
Victor R. Basili, Richard W. Selby, Jr.
Pages: 386 - 391
The role of logic and AI in the software enterprise (panel session):
panel description
Robert Balzer
Pages: 394 - 394

Another view of software (panel session)
Alan J. Perlis
Pages: 395 - 396

Intelligent support for the engineering of software (panel paper)
Gerald Jay Sussman
Pages: 397 - 399

The role of logic in software enterprise (panel paper)
W. M. Turski
Page: 400

Software processes are software too
L. Osterweil
Pages: 2 - 13

Process models, process programs, programming support
M. M. Lehman
Pages: 14 - 16

On the use of formal methods in software development
D. Bjorner
Pages: 17 - 29

On the role of an environment
V. Stenning
Pages: 30 - 35

Iteration in the software process; review of the 3rd International Software Process Workshop
M. Dowson
Pages: 36 - 41

An experiment in technology transfer: PAISLey specification of requirements for an undersea lightwave cable system
E.F. Berliner, P. Zave
Pages: 42 - 50

Parameterized programming in OBJ2
K. Futatsugi, J. Goguen, J. Meseguer, K. Okada
Pages: 51 - 60

Software interconnection models
D. E. Perry
Pages: 61 - 69

Modeling software failures and reliability growth during system testing
W. K. Ehrlich, T. J. Emerson
Software metrics using deviation value
Y. Miyazaki, N. Murakami
Pages: 83 - 91

On building software process models under the lamppost
W. Curtis, H. Krasner, V. Shen, N. Iscoe
Pages: 96 - 103

Specification of time dependencies and synthesis of concurrent processes
P. Ladkin
Pages: 106 - 115

SEGRAS—a formal and semigraphical language combining Petri nets and abstract data types for the specification of distributed systems
B. Krämer
Pages: 116 - 125

Specifying software/hardware interactions in distributed systems
G-C. Roman
Pages: 126 - 139

Version control in the Inscape environment
D. E. Perry
Pages: 142 - 149

Version control in families of large programs
J. F. H. Winkler
Pages: 150 - 161

Database system support for software engineering
P. A. Berstein
Pages: 166 - 178

An architecture for intelligent assistance in software development
G. E. Kaiser, P. H. Feiler
Pages: 180 - 188

Interform: a CAD system for program development
L. D. I. Serbanati
Pages: 190 - 197

Artificial intelligence and software engineering
D. Barstow
Pages: 200 - 211

Human-computer communication meets software engineering
C. Rathke
Pages: 216 - 224

A conceptual programming environment
S. P. Reiss
Pages: 225 - 235
A comparison of four design methods for real-time systems
J. C. Kelly
Pages: 238 - 252

Knowledge-based software design using design schemas
M. D. Lubars, M. T. Harandi
Pages: 253 - 262

Use of software engineering tools in Japan
T. Nomura
Pages: 263 - 269

A system for parallel programming
K. Schwan, R. Ramnath, S. Vasudevan, D. Ogle
Pages: 270 - 282

Lessons of current environments
L. Osterweil
Pages: 286 - 286

ISTAR and the contractual approach
M. Dowson
Pages: 287 - 288

Tool interfaces in integrated project support environments
A. Hall
Pages: 289 - 290

Ten years experiences of a software/hardware producing environment
Y. Matsumoto
Pages: 291 - 291

Evaluating software development environments
N. Weiderman
Pages: 292 - 293

Software process management: lessons learned from history
B. W. Boehm
Pages: 296 - 298

Production of large computer programs
H. D. Benington
Pages: 299 - 310

Pitfalls and safeguards in real-time digital systems with emphasis on
programming
W. A. Hosier
Pages: 311 - 327

Managing the development of large software systems: concepts and
techniques
W. W. Royce
Pages: 328 - 338
Prospects for future environments: introduction to panel session
C. J. Tully
Pages: 340 - 341

Improving the software process
W. E. Riddle
Pages: 344 - 344

Tailoring the software process to project goals and environments
V. R. Basili, H. D. Rombach
Pages: 345 - 357

A knowledge structure for reusing abstract data types
D. W. Embley, S. N. Woodfield
Pages: 360 - 368

From design to redesign
G. Fischer, A. C. Lemke, C. Rathke
Pages: 369 - 376

PARIS: a system for reusing partially interpreted schemas
S. Katz, C. A. Richter, K.-S. The
Pages: 377 - 385

Software evolution management: an integrated discipline for managing software
D. P. Schwartz
Pages: 388 - 398

A method for asynchronous parallelization
P. R. Itzigehl
Pages: 4 - 9

On the influence of scale in a distributed system
M. Satyanarayanan
Pages: 10 - 18

Programming at the processor-memory-switch level
M. R. Barbacci, C. B. Weinstock, J. M. Wing
Pages: 19 - 28

Theory-W software project management: a case study
B. Boehm, R. Ross
Pages: 30 - 40

Software development productivity tools and metrics
A. S. Duncan
Pages: 41 - 48

Integrated support for project management
D. Dixon
Pages: 49 - 58
Models of software development environments
D. E. Perry, G. E. Kaiser
Pages: 60 - 68

SDA: A novel approach to software environment design and construction
K. Kishida, T. Katayama, M. Matsuo, I. Miyamoto, K. Ochimizu, Nobuo Saito, Johner H. Sayler, K. Torii, L. G. Williams
Pages: 69 - 79

Rigi-A system for programming-in-the-large
H. A. Müller, K. Klashinsky
Pages: 80 - 86

Design and test of distributed applications
J. Fagerström
Pages: 88 - 92

Software engineering for distributed applications: the design project
M. Dr. Muhlhäuser
Pages: 93 - 101

Using raddle to design distributed systems
M. Evangelist, V. Y. Shen, I. R. Forman, M. Graf
Pages: 102 - 111

Heuristic software partitioning algorithms for distributed real-time applications
H. Xin, Z. Hong, C. Xiyao
Pages: 116 - 121

Requirements analysis for real time automation projects
M. Hagemann
Pages: 122 - 129

Tools for real time system design
R. A. Orr, M. T. Norris, R. Tinker, C. D. V. Rouch
Pages: 130 - 137

Assessing the quality of abstract data types written in ADA
D. W. Embley, S. N. Woodfield
Pages: 144 - 153

Modeling mutation on a vector processor
A. P. Mathur, E. W. Krauser
Pages: 154 - 161

A review of automated debugging systems: knowledge, strategies and techniques
M. Ducassé, A.-M. Emde
Pages: 162 - 171

Software process modeling: a behavioral approach
L. G. Williams
Pages: 174 - 186
Validating the TAME resource data model
D. R. Jeffery, V. R. Basili
Pages: 187 - 201

Modeling software tools with ICON
O. R. Fonorow
Pages: 202 - 220

Static Analysis-based program evolution support in the common lisp framework
K. Narayanswamy
Pages: 222 - 230

Modeling of data-processing software for generating and reusing their programs
N. Mano
Pages: 231 - 240

Interpretation in a tool-fragment environment
S. J. Zeil, E. C. Epp
Pages: 241 - 248

Implications of hardware advances for software development
A. I. Wasserman
Pages: 250 - 251

Demeter: a case study of software growth through parameterized classes
K. J. Lieberherr, A. J. Riel
Pages: 254 - 264

A programming environment supporting reuse of object-oriented software
H. Tarumi, K. Agusa, Y. Ohno
Pages: 265 - 273

Extension and software development
D. Notkin, W. G. Griswold
Pages: 274 - 283

An operational requirement description model for open systems
M. Hallman
Pages: 286 - 295

Language and visualization support for large-scale concurrency
G.-C. Roman
Pages: 296 - 308

High level specification of concurrency control in distributed database systems
L. Chiu, M. T. Liu
Pages: 309 - 317

m-EVES: A tool for verifying software
D. Craigen, S. Kromodimoeljo, I. Meisels, A. Neilson, B. Pase, M. Saaltink
Pages: 324 - 333

Algebraic specification of Macintosh's Quickdraw using OBJ2
A. T. Nakagawa, K. Futatsugi, S. Tomura
Pages: 334 - 343

An early report on encompass
R. B. Terwilliger, R. H. Campbell
Pages: 344 - 354

Enhancing program readability and comprehensibility with tools for program visualization
R. Baecker
Pages: 356 - 366

Design principles behind Chiron: a UIMS for software environments
M. Young, R. N. Taylor, D. B. Troup, C. D. Kelly
Pages: 367 - 376

Safety verification in Murphy using fault tree analysis
S. S. Cha, N. G. Leveson, T. J. Shimeall
Pages: 377 - 386

Statemate: a working environment for the development of complex reactive systems
D. Harel, H. Lachover, A. Naamad, A. Pnueli, M. Politi, R. Sherman, a. Shtul-Trauring
Pages: 396 - 406

KDA - A tool for automatic design evaluation and refinement using the blackboard model of control
H. Sharp
Pages: 407 - 416

Recording the reasons for design decisions
C. Potts, G. Bruns
Pages: 418 - 427

Deriving specifications from requirements
W. L. Johnson
Pages: 428 - 438

Automatic programming for streams II: transformational implementation
D. Barstow
Pages: 439 - 447

The space station information system and software support environment
C. W. Pittman
Pages: 455 - 458

The inscape environment
Dewayne E. Perry
Tool integration in the Pact environment
Ian Thomas
Pages: 13 - 22

The IDeA design environment
Mitchell D. Lubars
Pages: 23 - 32

Declarative visualization in the shared dataspace paradigm
Gruia-Catalin Roman, Kenneth C. Cox
Pages: 34 - 43

Task interaction graphs for concurrency analysis
Douglas L. Long, Lori A. Clarke
Pages: 44 - 52

Rethinking the taxonomy of fault detection techniques
Michael Young, Richard N. Taylor
Pages: 53 - 62

Software development process from natural language specification
Motoshi Saeki, Hisayuki Horai, Hajime Enomoto
Pages: 64 - 73

An algebraic data type specification language and its rapid prototyping environment
Luc Jadoul, Luc Duponcheel, Willy Van Puymbroeck
Pages: 74 - 84

Performance engineering as a part of the development life cycle for large-scale software systems
Gregory Fox
Pages: 85 - 94

A twenty year retrospective of the NATO software engineering conferences (panel session)
James Tomayko
Page: 95

A twenty year retrospective of the NATO software engineering conferences (panel session): twenty-year retrospective: the NATO software engineering conferences
James E. Tomayko
Page: 96

Twenty-year retrospective: the NATO software engineering conferences
James E. Tomayko
Page: 96

A twenty year retrospective of the NATO software engineering conferences (panel session): thoughts on software engineering
Bernard A. Galler
A twenty year retrospective of the NATO software engineering conferences (panel session): my thoughts on software engineering in the late 1960s
David Gries
Page: 98

A twenty year retrospective of the NATO software engineering conferences (panel session): remembrances of a graduate student
Mary Shaw
Pages: 99 - 100

A twenty year retrospective of the NATO software engineering conferences (panel session): the NATO conferences from the perspective of an active software engineer
Doug Ross
Pages: 101 - 102

A knowledge-base environment for the development of software parts composition systems
Bradley P. Allen, S. Daniel Lee
Pages: 104 - 112

SRE: a knowledge-based environment for large-scale software re-engineering activities
Wojtek Kozaczynski, Jim Q. Ning
Pages: 113 - 122

Tools to support formal methods
S. Patel, R. A. Orr, M. T. Norris, D. W. Bustard
Pages: 123 - 132

Software engineering for business applications: twenty years of software engineering (panel session): looking forward, looking back
Tom DeMarco
Page: 134

Software engineering for business applications: software engineering for business DP (panel session): looking back and looking forward
Michael Jackson
Page: 135

Software survivor
Nicholas Zvegintzov
Page: 136

ObNet: an object-oriented approach for supporting large, long-lived, highly configurable systems
T. Gallo, G. Serrano, F. Tisato
Pages: 138 - 144

A formal adaption method for process descriptions
Katsuro Inoue, Takeshi Ogihara, Tohru Kikuno, Koji Torii
Pages: 145 - 153
Object management in a CASE environment
Evan W. Adams, Masahiro Honda, Terrence C. Miller
Pages: 154 - 163

Stepwise refinement process with modularity
Ataru T. Nakagawa, Kokichi Futatsugi
Pages: 166 - 177

Modular specification of incremental program transformation systems
Alan Carle, Lori Pollock
Pages: 178 - 187

Validating system requirements by functional decomposition and dynamic analysis
Kaizhi Yue
Pages: 188 - 196

The relationship between slices and module cohesion
Linda M. Ott, Jeffrey J. Thuss
Pages: 198 - 204

A model for the reuse of software design information
Kevin W. Jameson
Pages: 205 - 216

A generic model for representing design methods
Colin Potts
Pages: 217 - 226

Software engineering research agendas (panel session): a view from the trenches
Peter Freeman, Ilene Birkwood, Jack Munson, Bob Martin, Al Pietrasanta, Vic Basili, Susan Gerhart
Page: 228

An error complexity model for software reliability measurement
Yutaka Nakagawa, Shuetsu Hanata
Pages: 230 - 236

Does imperfect debugging affect software reliability growth?
M. Ohba, X.-M. Chou
Pages: 237 - 244

The dimensionality of program complexity
J. C. Munson, T. M. Khoshgoftaar
Pages: 245 - 253

Software engineering in the year 2001 (panel session)
Robert Balzer, Michael Jackson, Alan Kay, Michael Sintzoff
Page: 256

Software engineering in the year 2001: software development in the year 2000 (panel session)
Software engineering in the year 2001: the scientific engineering of software (panel session)
Michael Sintzoff
Page: 258

Work structures and shifts: an empirical analysis of software specification teamwork
Salah Bendifallah, Walt Scacchi
Pages: 260 - 270

Software development: state of the art vs. state of the practice
Tom DeMarco, Tim Lister
Pages: 271 - 275

The state of software engineering practice
Watts S. Humphrey, David H. Kitson, Tim C. Kasse
Pages: 277 - 285

Structuring criteria for real time system design
Hassan Gomaa
Pages: 290 - 301

Completeness, robustness, and safety in real-time software requirements specification
M. S. Jaffe, N. G. Leveson
Pages: 302 - 311

Timing analysis of cyclic concurrent programs
C. Samuel Hsieh
Pages: 312 - 318

A process-oriented approach to configuration management
Yves Bernard, Pierre Lavency
Pages: 320 - 330

Software process modeling: principles of entity process models
Watts S. Humphrey, Marc I. Kellner
Pages: 331 - 342

A hierarchical and functional software process description and its enactment
Takuya Katayama
Pages: 343 - 352

Completely validated software (panel session)
Susan Gerhart
Page: 353

Complete validated software: validation through testing (panel session)
John C. Cherniavsky
Page: 354
Completely validated software: test adequacy and program mutation (panel session)
R. A. DeMillo
Pages: 355 - 356

Completely validated software: error-based validation completeness (panel session)
W. E. Howden
Page: 357

Completely validated software: mathematics-based software engineering for completely validated software (panel session)
Richard C. Linger
Page: 358

Completely validated software: completely validated software (panel session): a position paper
Richard A. Kemmerer
Pages: 359 - 360

Completely validated software: in defense of coverage criteria (panel session)
Elaine J. Weyuker
Page: 361

NLH/E: a natural language help system
Walter F. Tichy, Rolf L. Adams, Lars Holter
Pages: 364 - 374

Technology dependence in function point analysis: a case study and critical review
J. M. Verner, G. Tate, B. Jackson, R. G. Hayward
Pages: 375 - 382

Configuration management in BiiN SMS
R. W. Schwanke, E. S. Cohen, R. Gluecker, W. M. Hasling, D. A. Soni, M. E. Wagner
Pages: 383 - 393

Modeling the software process (panel session)
Gregory A. Hansen
Page: 395

Modeling the software process: modeling the software engineering process (panel session)
Gregory A. Hansen
Page: 396

Modeling the software process: what we really need are process model generators (panel session)
Barry Boehm
Pages: 397 - 397
Modeling the software process: three problems overcome with behavioral models of the software development process (panel session)
Bill Curtis
Pages: 398 - 399

Modeling the software process: software process modeling experience (panel session): panel session position paper
Marc I. Kellner
Pages: 400 - 401

Modeling the software process: `not waving but drowning’ (panel session): representation schemes for modelling software development
Anthony Finkelstein
Pages: 402 - 404

TRW's Ada process model for incremental development of large software systems
W. Royce
Pages: 2 - 11

Software process a àgrave; la algebra: OBJ for OBJ
A. T. Nakagawa, K. Futatsugi
Pages: 12 - 23

Evaluation criteria for functional specifications
S. Cardenas, M. V. Zelkowitz
Pages: 26 - 33

An experiment in formal software development: using the B theorem prover on a VDM case study
C. Lafontaine, Y. Ledru, P.-Y. Schobbens
Pages: 34 - 42

Recent advances in software measurement (abstract and references for talk)
V. R. Basili
Pages: 44 - 49

Multi-level specification and verification of real-time software
A. Gabrielian, M. K. Franklin
Pages: 52 - 62

Object-oriented specification of reactive systems
H.-M. Järvinen, R. Kurki-Suonio, M. Sakkinen, K. Systä
Pages: 63 - 71

Specification level interoperability
J. C. Wileden, Alexander L. Wolf, W. R. Rosenblatt, P. L. Tarr
Pages: 74 - 85

Design decisions for the incremental Adage framework
J.-L. Giavitto, G. Rosuel, A. Devarenne, A. Mauboussin
Pages: 86 - 95
Practice of quality modelling and measurement on software life-cycle
M. Hirayama, H. Sato, J. Tsuda
Pages: 98 - 107

Application of software reliability modelling to product quality and test process
W. K. Ehrlich, J. P. Stampfel, J. R. Wu
Pages: 108 - 116

Panel on software re-engineering
G. M. E. Lafue
Page: 118

The `Catch 22' of re-engineering
W. Kozaczynski
Page: 119

Human-oriented conceptual abstractions in the re-engineering of software
T. Biggerstaff
Page: 120

Software re-engineering position statement
E. Bush
Page: 121

CASE & reengineering: from archaeology to software perestroika
E. J. Chikofsky
Page: 122

Reengineering: can a program put intelligence in stupid programs?
M. Maiocchi
Pages: 123 - 124

Experience report: use of video for program documentation
T. DeMarco, C. Geertgens
Pages: 126 - 128

Using object-oriented development for support prototyping
L. Zucconi, G. Mack, L. G. Williams
Pages: 129 - 132

Getting started on metrics-JPL productivity and quality
M. Bush
Pages: 133 - 142

Logical animation
M. C. Costa, R. J. Cunningham, J. Booth
Pages: 144 - 149

System support for modular order-sorted horn clause specifications
H. Ganzinger, R. Schäfers
Pages: 150 - 159

An environment for specifying an executing hierarchical Petri nets
H. Oswald, R. Esser, R. Mattmann
Pages: 164 - 172

Mixing abstract and concrete modules: specification, development and prototyping
C. Choppy, S. Kaplan
Pages: 173 - 184

SACEM software validation
G. Guiho, C. Hennebert
Pages: 186 - 191

Experience using the Graphite meta-tool
M. Dowson
Pages: 192 - 195

Improving software quality: the use of formal inspections at the JPL
M. Bush
Pages: 196 - 199

A rule-based approach to modular system design
F. Parisi-Presicce
Pages: 202 - 211

An object-oriented software application architecture
M. Ganti, P. Goyal, S. Podar
Pages: 212 - 220

On the assessment of safety-critical software systems
J.-C. Laprie
Page: 222

Evaluation of software safety
N. B. Leveson
Pages: 223 - 224

Some experiences of critical software development
E. Pilaud
Pages: 225 - 226

Assessing failure probabilities in safety-critical systems containing software
M. Thomas
Page: 227

ISPW5: experience with software process models
D. E. Perry
Pages: 230 - 232

FM 89: assessment of formal methods for trustworthy computer systems
D. Craigen
Pages: 233 - 235

Building an evolution transformation library
W. L. Johnson, M. Feather
Pages: 238 - 248

LaSSIE—a knowledge-based software information system
P. T. Devanbu, R. J. Brachman, P. G. Selfridge, B. W. Ballard
Pages: 249 - 261

Position statement: ICSE-12 Workshop on Industrial Experience Using Formal Methods
D. Bjorner, L. Druffel
Pages: 264 - 266

Negotiation behavior during requirements specification
W. N. Robinson
Pages: 268 - 276

Prism = methodology + process-oriented environment
N. H. Madhavji, V. Gruhn, W. Deiters, W. Schäfer
Pages: 277 - 288

The evolution of technology transfer at MCC's Software Technology Program: from didactic to dialectic
J. D. Babcock, L. A. Belady, N. C. Gore
Pages: 290 - 299

Implementing faceted classification for software reuse (experience report)
R. Prieto-Diaz
Pages: 300 - 304

Towards systems engineering—a personal view of progress
D. E. Talbot
Pages: 306 - 307

Automatic maintenance of routine programming tasks based on a declarative description (experience report)
D. Bernstein, R. Farrow
Pages: 310 - 315

A decision-based configuration process model
T. Rose, M. Jarke
Pages: 316 - 325

Experience in using defined processes for technology transfer
K. F. Fischer
Page: 328

Position paper on technology transfer
R. Alonzo
Page: 329

CEFRIEL: an innovative approach to university-industry cooperation in information technologies
M. Decina
Technology transfer as collaboration: the receptor group
P. J. Fowler
Pages: 332 - 333

Semi formal process model for technology transfer
G. Glynn
Pages: 334 - 335

Technology transfer aspects of environment construction
K. Kishida
Page: 336

A methodology for prototyping-in-the-large
James Purtilo, Aaron Larson, Jeff Clark
Pages: 2 - 12

Aspect: an economical bug-detector
Daniel Jackson
Pages: 13 - 22

Using weaves for software construction and analysis
Michael M. Gorlick, Rami R. Razouk
Pages: 23 - 34

Formal methods: an international perspective
Susan Gerhart
Pages: 36 - 37

A comparison of U.S. and Japanese software process maturity
Watts S. Humphrey, David H. Kitson, Julia Gale
Pages: 38 - 49

Software reuse: is it delivering?
William B. Frakes, Ted J. Biggerstaff, Ruben Prieto-Diaz, Kazuo Matsumura, Wilhelm Schaefer
Pages: 52 - 59

Understanding natural programs using proper decomposition
John Hartman
Pages: 62 - 73

Toward new techniques to assess the software implementation process
John D. Litke
Pages: 74 - 82

An intelligent tool for re-engineering software modularity
Robert W. Schwanke
Pages: 83 - 92

Augmenting SADT to develop computer support for cooperative work
David A. Marca
Pages: 94 - 103
Designing software for use by humans, not machines
L. Ruston, M. J. Muller, K. D. Cebulka
Pages: 104 - 113

Extending the Potts and Bruns model for recording design rationale
Jintae Lee
Pages: 114 - 125

CASE at the start of the 1990's
Ronald J. Norman, Wayne Stevens, Elliot J. Chikofsky, John Jenkins, Burt L. Rubenstein, Gene Forte
Pages: 128 - 139

FM91: formal methods workshop
Dan Craigen, Peter Ryan
Pages: 140 - 142

Non-technological issues in software engineering
Marc I. Kellner, Bill Curtis, Tom DeMarco, Kouichi Kishida, Maurice Schlumberger, Colin Tully
Pages: 144 - 146

Techies as non-technological factors in software engineering?
Bill Curtis
Pages: 147 - 148

Non-technological issues in software engineering
Tom DeMarco
Pages: 149 - 150

Position statement
Kouichi Kishida
Page: 151

Software engineering management
Maurice Schlumberger
Pages: 152 - 153

A failure of management nerve and vision
Colin Tully
Pages: 154 - 155

Tolerating inconsistency
Robert Balzer
Pages: 158 - 165

The prism model of changes
Nazim H. Madhavji
Pages: 166 - 177

Progress toward automated software testing
Richard A. DeMillo
Pages: 180 - 183
Tool support for formal methods
Dan Craigen
Pages: 184 - 185

Software development for parallel systems
J. Dongarra
Page: 188

Hiding distribution in distributed systems
Michael Banâtre
Pages: 189 - 196

PRESTIGE: a CASE workbench for the JSD implementor
Andrew Bass, Maeve Boyle, Bryan Ratcliff
Pages: 198 - 207

User interface development and software environments: the Chiron-1 system
Rudolf K. Keller, Mary Cameron, Richard N. Taylor, Dennis B. Troup
Pages: 208 - 218

Experiences with an environment generation system
Steven S. Popovich, William M. Schell, Dewayne E. Perry
Pages: 219 - 224

Parameter value computation by least square method and evaluation of software availability and reliability at service-operation by the hyper-geometric distribution software reliability growth model (HGDM)
Raymond Jacoby, Yoshihiro Tohma
Pages: 226 - 237

Integrating prior knowledge with a software reliability growth model
Joseph M. Caruso, David W. Desormeau
Pages: 238 - 245

Defect type and its impact on the growth curve
Ram Chillarege, Wei-Lun Kao, Richard G. Condit
Pages: 246 - 255

ISPW6: support for the software process
Takuya Katayama
Pages: 258 - 262

International workshop on software quality improvement
T. Tamai
Page: 263

Ethical considerations in software engineering
Donald Gotterbarn
Pages: 266 - 274

Cost estimation of software intensive projects: a survey of current practices
Jairus Hihn, Hamid Habib-agahi
Metric-driven analysis and feedback systems for enabling empirically guided software development
Richard W. Selby, Adam A. Porter, Doug C. Schmidt, Jim Berney
Pages: 288 - 298

Process assessments in NASA
Marilyn W. Bush
Pages: 299 - 304

The information technology security evaluation criteria
Christian Jahl
Pages: 306 - 312

Computer emergency response—an international problem
R. Pethia
Page: 313

Cognitive tools for locating and comprehending software objects for reuse
Gerhard Fischer, Scott Henninger, David Redmiles
Pages: 318 - 328

Software reuse in an industrial setting: a case study
Michael F. Dunn, John C. Knight
Pages: 329 - 338

Domain modeling for software engineering
Neil Iscoe, Gerald B. Williams, Guillermo Arango
Pages: 340 - 343

Hardware/software codesign: a perspective
David W. Franke, Martin K. Purvis
Pages: 344 - 352

Directions in software engineering education
John Werth, Laurie Werth
Pages: 353 - 357

High-pressure steam engines and computer software
Nancy G. Leveson
Pages: 2 - 14

Trusted computing systems: the ProCoS experience
Dines Bjørner
Pages: 15 - 34

Design of dependable real-time systems
Andrew Lister
Pages: 35 - 36

A measure for composite module cohesion
Sukesh Patel, William Chu, Rich Baxter
Pages: 38 - 48
Productivity analysis of software development with an integrated CASE tool
Michio Tsuda, Yosuke Morioka, Masato Takadachi, Mayumi Takahashi
Pages: 49 - 58

Estimating software fault content before coding
Pages: 59 - 65

Incremental testing of object-oriented class structures
Mary Jean Harrold, John D. McGregor, Kevin J. Fitzpatrick
Pages: 68 - 80

Testing for linear errors in nonlinear computer programs
Faten H. Afifi, Lee J. White, Steven J. Zeil
Pages: 81 - 91

Towards a method of programming with assertions
David S. Rosenblum
Pages: 92 - 104

Specification-based test oracles for reactive systems
Debra J. Richardson, Stephanie Leif Aha, T. Owen O'Malley
Pages: 105 - 118

Connecting software components with declarative glue
Brian W. Beach
Pages: 120 - 137

Visualizing and querying software structures
Mariano Consens, Alberto Mendelzon, Arthur Ryman
Pages: 138 - 156

Software evolution through iterative prototyping
Neil Goldman, K. Narayanaswamy
Pages: 158 - 172

Object-oriented analysis for evolving systems
Mitchell Lubars, Greg Meredith, Colin Potts, Charles Richter
Pages: 173 - 185

Formal specification of asynchronous distributed real-time systems by APTL
Farn Wang, Al Mok, E. Allen Emerson
Pages: 188 - 198

Validating real-time systems by history-checking TRIO specifications
Miguel Felder, Angelo Morzenti
Pages: 199 - 211

Graphical specifications for concurrent software systems
L. K. Dillon, G. Kutty, L. E. Moser, P. M. Melliar-Smith, Y. S. Ramakrishna
Pages: 214 - 224

Seeking concurrency in rule-based programming
Gruia-Catalin Roman, Rose F. Gamble, William E. Ball
Pages: 225 - 234

The property vector specification of a multiset iterator
Trevor W. Pearce, David A. Lamb
Pages: 235 - 245

A toolbox for the verification of LOTOS programs
Jean-Claude Fernandez, Hubert Garavel, Laurent Mounier, Anne Rasse, Carlos Rodriguez, Joseph Sifakis
Pages: 246 - 259

Concepts and implementation of a rule-based process engine
Burkhard Peuschel, Wilhelm Schäfer
Pages: 262 - 279

Process programming by hindsight
Pankaj K. Garg, Sanjay Bhansali
Pages: 280 - 293

Call path profiling
Robert J. Hall
Pages: 296 - 306

GENOA: a customizable language- and front-end independent code analyzer
Premkumar T. Devanbu
Pages: 307 - 317

Experience report on software reuse project: its structure, activities, and statistical results
Sadahiro Isoda
Pages: 320 - 326

A general economics model of software reuse
J. E. Gaffney, Jr., R. D. Cruickshank
Pages: 327 - 337

Software reuse economics: cost-benefit analysis on a large-scale Ada project
Johan Margono, Thomas E. Rhoads
Pages: 338 - 348

Behavior sampling: a technique for automated retrieval of reusable components
Andy Podgurski, Lynn Pierce
Pages: 349 - 361

Programming-in-the-large: past, present, and future
Walter F. Tichy
The software engineering laboratory: an operational software experience factory
Victor Basili, Gianluigi Caldiera, Frank McGarry, Rose Pajerski, Gerald Page, Sharon Waligora
Pages: 370 - 381

Recent advances in software estimation techniques
Richard E. Fairley
Pages: 382 - 391

The use of program dependence graphs in software engineering
Susan Horwitz, Thomas Reps
Pages: 392 - 411

Program visualization: the art of mapping programs to pictures
Gruia-Catalin Roman, Kenneth C. Cox
Pages: 412 - 420

Cleanroom software engineering for zero-defect software
Richard C. Linger
Pages: 2 - 13

Simulating the behaviour of software modules by trace rewriting
Yabo Wang, David Lorge Parnas
Pages: 14 - 23

Observations on industrial practice using formal methods
Susan Gerhart, Dan Craigen, Ted Ralston
Pages: 24 - 33

Rule-based approach to computing module cohesion
Arun Lakhotia
Pages: 35 - 44

Experimental evaluation of a fuzzy-set based measure of software correctness using program mutation
Farokh B. Bastani, Giuseppe DiMarco, Alberto Pasquini
Pages: 45 - 54

Modeling and managing risk early in software development
Lionel C. Briand, William M. Thomas, Christopher J. Hetmanski
Pages: 55 - 65

David H. Kitson, Stephen M. Masters
Pages: 68 - 77

A comprehensive process model for studying software process papers
Rudolf K. Keller, Richard Lajoie, Nazim H. Madhavji, Tilmann F. Bruckhaus, Kamel Toubache, Won-Kook Hong, Khaled El Emam
Pages: 78 - 88
Post-process feedback with and without attribute focusing: a comparative evaluation
Inderpal Bhandari, Norman Roth
Pages: 89 - 98

An experimental evaluation of selective mutation
A. Jefferson Offutt, Gregg Rothermel, Christian Zapf
Pages: 100 - 107

Dynamic mutation testing in integrated regression analysis
Janusz Laski, Wojciech Szermer, Piotr Luczycki
Pages: 108 - 117

Exploring dataflow testing of arrays
Dick Hamlet, Bruce Gifford, Borislav Nikolik
Pages: 118 - 129

A bi-level language for software process modeling
Gail E. Kaiser, Steven S. Popovich, Israel Z. Ben-Shaul
Pages: 132 - 143

Computational reflection in software process modeling: the SLANG approach
Sergio Bandinelli, Alfonso Fuggetta
Pages: 144 - 154

Prototyping a process monitoring experiment
Mark G. Bradac, Dewayne E. Perry, Lawrence G. Votta
Pages: 155 - 165

Requirements and design change in large-scale software development: analysis from the viewpoint of process backtracking
Tetsuo Tamai, Akito Itou
Pages: 167 - 176

Approach and case study of requirement analysis where end users take an active role
J. M. Drake, W. W. Xie, W. T. Tsai, I. A. Zualkernan
Pages: 177 - 186

Expressing the relationships between multiple views in requirements specification
Bashar Nuseibeh, Jeff Kramer, Anthony Finkelstein
Pages: 187 - 196

Domain modeling—overview & ongoing research at EDS
Neil Iscoe
Pages: 198 - 200

Model based process assessments
Clement L. McGowan, Shawn A. Bohner
Pages: 202 - 211

Software improvements in an international company
Harvey Wohlwend, Susan Rosenbaum
Prototyping in industrial software projects—bridging the gap between theory and practice
Horst Lichter, Matthias Schneider-Hufschmidt, Heinz Züllighoven
Pages: 212 - 220

A process for consolidating and reusing design knowledge
Guillermo Arango, Eric Schoen, Robert Pettengill
Pages: 221 - 229

The Graft-Host method for design change
Guillermo Arango, Eric Schoen, Robert Pettengill, Josiah Hoskins
Pages: 231 - 242

Developing initial OOA models
Mitchell Lubars, Colin Potts, Charles Richter
Pages: 243 - 254

Predicate-based test generation for computer programs
Kuo-Chung Tai
Pages: 255 - 264

Modeling software for accurate data flow representation
Hasan Ural, Bo Yang
Pages: 267 - 276

Coverage measurement experience during function test
Paul Piwowarski, Mitsuru Ohba, Joe Caruso
Pages: 277 - 286

``...And nothing else changes'': the frame problem in procedure specifications
Alex Borgida, John Mylopoulos, Raymond Reiter
Pages: 303 - 314

Documentation for safety critical software
P.-J. Courtois, D. L. Parnas
Pages: 315 - 323

A high level language for specifying graph based languages and their programming environments
M. F. Kleyn, J. C. Browne
Pages: 324 - 335

An examination of the current state of IPSE technology
Alan W. Brown
Pages: 338 - 347

Use of an environment classification model
Marvin V. Zelkowitz
Pages: 348 - 357
Programming heterogeneous transactions for software development environments
Peri Tarr, Stanley M. Sutton, Jr.
Pages: 358 - 369

Generalized behavior-based retrieval
Robert J. Hall
Pages: 371 - 380

Automating the detection of reusable parts in existing software
Michael F. Dunn, John C. Knight
Pages: 381 - 390

Building, modifying and using component generators
Stephen B. Ornburn, Richard J. LeBlanc, Jr.
Pages: 391 - 402

Test templates: a specification-based testing framework
P. A. Stocks, D. A. Carrington
Pages: 405 - 414

An analytical comparison of the fault-detecting ability of data flow testing techniques
Phyllis G. Frankl, Elaine J. Weyuker
Pages: 415 - 424

Reliable software and communication: software quality, reliability, and safety
S. R. Dalal, J. R. Horgan, J. R. Kettenring
Pages: 425 - 435

Software components in a data structure precompiler
Marty Sirkin, Don Batory, Vivek Singhal
Pages: 437 - 446

Adding implicit invocation to traditional programming languages
David Garlan, Curtis Scott
Pages: 447 - 455

Applying algorithm animation techniques for program tracing, debugging, and understanding
Sougata Mukherjea, John T. Stasko
Pages: 456 - 465

Software process maturity: measuring its impact on productivity and quality
Howard A. Rubin
Pages: 468 - 476

Inter-item correlations among function points
Barbara Kitchenham, Kari Känsälä
Pages: 477 - 480

The concept assignment problem in program understanding
Ted J. Biggerstaff, Bharat G. Mitbander, Dallas Webster
Pages: 482 - 498

RECAST: reverse engineering from COBOL to SSADM specification
Helen M. Edwards, Malcolm Munro
Pages: 499 - 508

Program and interface slicing for reverse engineering
Jon Beck, David Eichmann
Pages: 509 - 518

**Proceedings of an ACM Conference on Language Design for Reliable Software**

Process structuring, synchronization, and recovery using atomic actions
D. B. Lomet

Early experience with Mesa
Charles M. Geschke, James H. Morris, Jr., Edwin H. Satterthwaite
Revised version in CACM 1977

Abstraction mechanisms in CLU
Barbara Liskov, Alan Snyder, Russell Atkinson, Craig Schaffert
Revised version in CACM 1977

Towards a discipline of real-time programming
N. Wirth
Revised version in CACM 1977

**Proceedings of the software quality assurance workshop on Functional and performance issues, 1978**

Software Configuration Management
Edward H. Bersoff, Vilas D. Henderson, Stan G. Siegel
republished in TSE 1984

A framework for the measurement of software quality
Joseph P. Cavano, James A. McCall

**Proceedings of the 2nd International Workshop on Software configuration management, October 24-27, 1989 [mostly position papers]**

Change oriented versioning in a software engineering database
A. Lie, R. Conradi, T. M. Didriksen, E.-A. Karlsson

Cross references are features
R. W. Schwanke, M. A. Platoff

Revision control in an integrated software development environment
B. Westfechtek
20

Configuration management using SySL
R. Thomson, I. Sommerville
3

Tool configuration assistant
S. Dart
1

Maintaining the evolution of software objects in an integrated environment
A. Gustavsson
3

Configuration management in the PACT software engineering environment
I. Simmonds
16

The change request process
M. Lacroix, P. Lavency
4

Experiences in configuration management for modula-2
M. Jordan
1

Object-oriented configuration control
V. Ambriola, L. Bendix
8

Modeling configuration as transactions
G. W. Kaiser
4

Orthogonal version management
C. Reichenberger
24

Aladdin/lamp: Configuration management tools for distributed computer control systems
A. Pakstas
2

Incremental variant control
U. Schroeder
3

Conflict management in a source version management system
B. Mack-Crane, A. Pal
2

Commercial realtime software needs different configuration management
W. M. Gentleman, A. MacKay, D. A. Stewart
19

Replacing version-control with job-control
G. M. Clemm
15

Object derivation and validation from a data base definition
A. Dinsart, B. Hucq, N. Piperakis, P. J. Thollembeck
0

Proceedings of the 3rd international workshop on Software configuration management, 1991

Concepts in configuration management systems
Susan Dart

Structure-oriented merging of revisions of software documents
Bernhard Westfechtel
52

Software architecture analysis: a case study
Ronald Lange, Robert W. Schwanke
21

Dimensions of consistency in source versions and system compositions
Dewayne E. Perry
5

Epochs, configuration schema, and version cursors in the KBSA framework CCM model
John Kimball, Aaron Larson
5

VMCM, a PCTE based version and configuration management system
K. Berrada, F. Lopez, R. Minot
6

Managing multi-variant software configuration
Peter J. Nicklin
11

Duplicates': a convention for defining configurations in PCTE-based environments
Ian Simmonds
0

Supporting reuse and configuration: a port based SCM model
D. Aquilino, P. Asirelli, P. Inverardi, P. Malara
2

Conditions to assure semantically consistent software merges in linear time
Keith Gallagher
7
A logic-based approach to system modelling
Anthony Rich, Marvin Solomon

An event-based model of software configuration management
David S. Rosenblum, Balachander Krishnamurthy

Cooperating transactions against the EPOS database
Reidar Conradi, Carl Chr. Malm

Flexible support for cooperation in software development
Shem M. J. Lacroix, D. Roelants, J. E. Waroquier

A process-oriented version and configuration management model for communications software
Shem J. Ochuodho, Alan W. Brown

Sorceress: a database approach to software configuration management
Chris Sheedy

An object-oriented model of software configuration management
Hal Render, Roy Campbell

History management system
Scott A. Kramer

Delta storage for arbitrary non-text files
Christoph Reichenberger

Heterogeneous configuration management with DSEE
David Lubkin

Version management in distributed network environment
Bogdan Korel, Horst Wedde, Srinivas Magaraj, Kalique Nawaz, Venugopal Dayana

Design by decomposition of multiparty interactions in Raddle87
I. R. Forman

Rooting UNITY
R. Gerth, A. Pnueli
Specifying concurrent systems with Dgr-grammars
S. M. Kaplan, S. K. Goering, R. H. Campbell
0

**Proceedings of the 5th international workshop on Software specification and design, 1989**

Configuration support for system description, construction and evolution
J. Kramer, J. Magee, M. Sloman
93
Larger scale systems require higher-level abstractions
M. Shaw
90

Domain analysis: from art form to engineering discipline
G. Arango
150

Multiparty specification
A. Finkelstein, H. Fuks
75

Integrating multiple specifications using domain goals
W. N. Robinson
75

Action system approach to the specification and design of distributed systems
R. Kurki-Suonio, H.-M. Järvinen
25

Implementing a shared dataspace language on a message-based multiprocessor
G.-C. Roman, K. C. Cox
5

A specification language for parallel architectures and algorithms
T. Bräunl
0

On the algebraic representation and treatment of processing structures
H. von Issendorff
1

Reflective computation in object-oriented concurrent systems and its applications
T. Watanabe, A. Yonezawa
0

A general way to put time in Petri nets
C. Ghezzi, D. Mandrioli, S. Morasca, M. Pezze
45
Toward reliable reactive systems
L.-M. Gilham, A. Goldberg, T. C. Wang
9

An operational semantics of real time design language RT-CDL
L. Y. H. Liu, R. K. Shyamasundar
3

Timing extensions to structured analysis for real time systems
L. Peters
5

Reasoning of real-time distributed programming languages
R. K. Shyamasundar, J. Hooman, R. Gerth
0

Analysis capabilities for requirements specified in statecharts
B. E. Melhart, N. G. Leveson, M. S. Jaffe
4

Interface specification with temporal logic
M. Kooij
2

Viewpoint analysis: a case study
J. C. S. P. Leite
16

Specifying reusable components using Z: realistic sets and dictionaries
R. L. London, K. R. Milsted
19

A look through PRISMA: towards pluralistic knowledge-based environments
for software specification acquisition
C. Niskier, T. Maibaum, D. Schwabe
26

A net method for specification of reusable software
H. S. Dhama, D. Shtern
2

Software development using models
R. S. D’Ippolito, C. P. Plinta
7

Discovering, visualizing, and controlling software structure
R. W. Schwanke, R. Z. Altucher, M. A. Platoff
46

A logic of action for supporting goal-oriented elaborations of
requirements
Detecting interference when merging specification evolutions
M. S. Feather
33

A proposed perspective shift: viewing specification design as a planning problem
J. S. Anderson, S. Fickas
37

The use of lexical affinities in requirements extraction
Y. S. Maarek, D. M. Berry
30

Analogical approach to specification derivation
K. Miriyala, M. T. Harandi
8

The requirements apprentice: an initial scenario
H. B. Reubenstein, R. C. Waters
34

Human factors: its place in system development methods
R. Summersgill, D. P. Browne
4

Specifying coordinators: guidelines for groupware developers
D. A. Marca
Pages: 235 - 237
3

Software design issues: a very large information systems perspective
G. B. Williams, C. Mui, W. Alagappan, B. B. Johnson
0

Formally specifying electronic instruments
N. Delisle, d. Garlan
19

LCD-reification: a formal method for developing Prolog programs
F. Lin, F. E. Hunt
0

Developing formally verified Ada programs
N. Ramsey
12

An example of formal specification as an aid to design and development
R. B. Terwilliger, M. J. Maybee, L. J. Osterweil
0
Formal specification of fault-tolerance and its relation to computer security
D. G. Weber
21

Systematic design of algebraic specifications
S. Antoy
8

Data-flow design as a visual programming language
S. Eisenbach, L. McLoughlin, C. Sadler
4

A formal framework for the design of development environments
M. M. Pett, M. D. Rice, S. B. Seidman
2

Functional programming: executable specifications and program transformations
S. Thompson
7

Representing first order logic-based specifications in Petri-net-like graphs
K. Yue
2

Proceedings of the 6th international workshop on Software specification and design, 1991

Goal-directed concept acquisition in requirements elicitation
Anne Dardenne, Stephen Fickas, Axel van Lamsweerde

Heterogeneous design idioms for software architecture
Mary Shaw
41

FUNSOFT nets: a Petri-net based software process modeling language
Wolfgang Emmerich, Volker Gruhn
48

Declarative specification and declarative programming
Manfred Broy
Pages: 2 - 11
0

Specification criticism via goal-directed envisionment
Keith Downing, Stephen Fickas
2

Experiences using statecharts for a system requirements specification
N. G. Leveson, M. Heimdahl, H. Hildreth, J. Reese, R. Ortega
19
PoliS: a programming model for multiple tuple spaces
Paolo Ciancarini
23

Abstraction and composition in ∆-specifications of concurrent systems
Joseph P. Loyall, Simon M. Kaplan, Steven K. Goering
1

Prototyping and formal analysis of concurrent and distributed systems
Bernd Krämer
4

Techniques for the design of communicating processes
Sylvie Damy, Guy-René Perrin
1

Issues in the specification and design of parallel programs
Robert G. Babb
5

Real-time specification and modeling with joint actions
Reino Kurki-Suonio, Kari Systä, Jüri Vain
15

Dealing with different time scales in formal specifications
37

An integration environment to put formal specifications into practical use in real-time systems
Shinichi Honiden, Akihiko Ohsuga, Naoshi Uchihiara
3

A compositional proof system for real-time systems based on explicit clock temporal logic
J. Hooman, R. Kuiper, P. Zhou
0

A formal specification of a visual language editor
Jeannette M. Wing, Amy Moormann Zaremski
5

Developing reactive systems in a VDM framework
Y. Ledru
4

Telecommunication service description using state transition rules
Y. Hirakawa, T. Takenaka
36

Equational reasoning about LOTOS specifications: a rewriting approach
Rocco De Nicola, Paola Inverardi, Monica Nesi
On the specification of software components
Michael Goedicke, Harald Schumann, Joachim Cramer
5

On the meaning of "but"
Pierre-Yves Schobbens
17

Authoring-in-the-large: software engineering techniques for hypertext application design
Franca Garzotto, Paolo Paolini, Daniel Schwabe
10

A system for multiparadigm development of software systems
Scott Meyers, Steven P. Reiss
18

Addressing requirements issues within a conceptual modeling environment
PP 4

A model for composite system design
E. Doerry, S. Fickas, R. Helm, M. Feather
PP 7

Reuse of analogous specifications during requirements analysis
N. A. M. Maiden, A. G. Sutcliffe
PP 2

Specifying groupware requirements from direct experience
David A. Marca
PP 2

Swarming over the software barrier
H. Conrad Cunningham
PP 1

Advantages and limits of formal approaches for ultra-high dependability
M. C. Gaudel
PP 11

Preconditions for understanding
David Garlan
PP 7

Overcoming shortcomings in schema-based software design systems
Hing-Yan Lee, Mehdi T. Harandi
PP 2

Designing software for customization and evolution
David Barstow, Guillermo Arango
Seven (plus or minus two) challenges for requirements research
Colin Potts

Proceedings of the 7th international workshop on Software specification and design, 1993

An active hypertext model for system requirements
Colin Potts, Kenji Takahashi

Fine-grain process modelling
Bashar Nuseibeh, Anthony Finkelstein, Jeff Kramer

Bridging the requirements gap: policies, goals and domains
A. G. Sutcliffe, N. A. M. Maiden

An incremental approach to reachability analysis of distributed programs
Kuo-Chung Tai, Pramod V. Koppol

A framework for program development based on schematic proof
David Basin, Alan Bundy, Ina Kraan, Sean Matthews

Taking more of the soft out of software engineering
Tom Maibaum

Design and validation of a message-passing system
Fuyau Lin

Specification and refinement of a message router
H. Conrad Cunningham, Yinxiu Cai

Modular reasoning about open systems: a case study of distributed commit
R. Das, A. Fekete

An empirical approach to software architectures
Dilip Soni, Robert L. Nord, Liang Hsu

A Petri-net-based approach to real-time program analysis
Ugo Buy, Robert Sloan
Towards extensible graphical formalisms
Carlo Ghezzi, Mauro Pezzè
4

Requirements elicitation and formalization through external design and object-oriented specification
Oscar Barros
3

Specifying reactive systems with attributed finite state machines
Shiyuan Ding, Takuya Katayama
4

Identical tasks and counter variables in an integer programming-based approach to verification
James C. Corbett
6

Merlot: a tool for analysis of real-time specifications
Carlo Bellettini, Miguel Felder, Mauro Pezzè
10

Specifying reactive systems by abstract events
E. Astesiano, G. Reggio
8

A pragmatic basis for the formal development of distributed systems
Kenneth R. Wood
1

Related specifications reuse
P. Wolff
1

Predicting operational behaviour of abstract programs during formal developments
Yamin AIT-AMEUR
0

Using design fragments to modularize the design of communications in concurrent systems
Joanne L. Boyd, Gerald M. Karam
1

A taxonomy of inheritance semantics
Alyce Faulstich-Brady
1

Reflective specification: applying a reflective language to formal specification
Motoshi Saeki, Takeshi Hiroi, Takanori Ugai
4
Specification approaches express different world hypotheses
David A. Marca, Clement L. McGowan
1

Conference proceedings on Formal methods in software development, 1990

The role of formal reusable frameworks
David Garlan
13

Formal specification for a clinical cyclotron control system
Jonathan Jacky
13

On the soundness of graphical representations of interconnected processes
in LOTOS
Tommaso Bolognesi
1

Experience with an Estelle development system
Anthony Chung, Deepinder Sidhu
1

Making languages more powerful by removing limitations
Jozef De Man
2

Toward special-purpose program verification
Paul Eggert
Pages: 25 - 29
0

Constructing programs in a calculus of lazy interpreters
Richard A. Frost
37 republished in The Computer Journal 1989

Applying VDM to large developments
Yves Ledru, Pierre-Yves Schobbens
3

The algebraic specification language LOTOS: an industrial experience
Luigi Logrippo, Tim Melanchuk, Robert J. Du Wors
11

Invariant properties in a dialog system
K. T. Narayana, Sanjeev Dharap
8

L.0: a parallel executable temporal logic language
Linda Ness
4

Formal manipulation of modular software systems
Robert L. Nord, Peter Lee, William L. Scherlis
Pages: 90 - 99

Formal methods in mathematics
Richard Platek
1

Formal methods in transformational derivation of programs
Uday S. Reddy
7

Formalised development of software by machine assisted transformation
Colin Runciman, Mike Firth
0

Structured analysis—formal design, using stream and object oriented formal specifications
Hans Toetenel, Jan van Katwijk, Nico Plat
2

Adding relational abstraction to programming languages
David S. Wile
6

Experience with the Larch Prover
Jeannette M. Wing, Chun Gong
6

Application of formal methods to system and software specification
William G. Wood
0

Verified program support environments
William D. Young
Pages: 147 - 149

Proceedings of the workshop on Rapid prototyping, 1982

Rapid prototyping workshop: overview
Stephen L. Squires, Marvin Zelkowitz, Martha Branstad
Pages: 2 - 2

Operational specification as the basis for rapid prototyping
Robert M. Balzer, Neil M. Goldman, David S. Wile
Pages: 3 - 16

Mappings for rapid prototyping
Martin S. Feather
Pages: 17 - 24
Using symbolic execution to characterize behavior
Donald Cohen, William Swartout, Robert Balzer
Pages: 25 - 32

Rapid prototyping, automatic programming, and experimental sciences
David Barstow
Pages: 33 - 34

Rapid prototyping of information management systems
Bruce I. Blum, Raymond C. Houghton
Pages: 35 - 38

Rapid prototyping using executable requirements specifications
Alan M. Davis
Pages: 39 - 44

Rapid prototyping by means of abstract module specifications written as trace axioms
John K. Dixon, John McLean, David L. Parnas
Pages: 45 - 49

A prototyping language for text-processing applications
Pages: 50 - 53

Prototyping in ADA: a case study
Arthur G. Duncan
Pages: 54 - 60

Implementation prototypes in the development of programming language features
Ray Ford, Chris Marlin
Pages: 61 - 66

Experience from computer supported prototyping for information flow in hospitals
Hans Gill, Roland Lindvall, Olle Rosin, Erik Sandewall, Henrik Sörensen, Ove Wigertz
Pages: 67 - 70

A study in prototyping
N. H. Gehani
Pages: 71 - 74
Rapid prototyping: in the OBJ executable specification language
Joseph Goguen, Jose Meseguer
Pages: 75 - 84

The use of quick prototypes in the secure military message systems project
C. Heitmeyer, C. Landwehr, M. Cornwell
Pages: 85 - 87

Scenario-based prototyping for requirements identification
James W. Hooper, Pei Hsia
Pages: 88 - 93

Prototyping: a more reasonable approach to system development
Hans E. Keus
Pages: 94 - 95

Rapid prototyping and requirements specification using PDS
A. Klausner, T. E. Konchan
Pages: 96 - 105

Research proposal to study: the role of executable metric models in the programming process
M. M. Lehman
Pages: 106 - 111

Specification prototyping
Glenn H. MacEwen
Pages: 112 - 119

ACT/1: a tool for information systems prototyping
R. E. A. Mason, T. T. Carey, A. Benjamin
Pages: 120 - 126

System sketching: the generation of rapid prototypes for transaction based systems
Gerard C. McCoyd, John R. Mitchell
Pages: 127 - 132

HIBOL: a language for fast prototyping in data processing environments
Roland T. Mittermeir
Pages: 133 - 140
Use of annotated schemes for developing prototype programs
J. Ramanathan, C. J. Shubra
Pages: 141 - 149

The disciplined use of simplifying assumptions
Charles Rich, Richard C. Waters
Pages: 150 - 154

Approaches to executable specifications
Stephen W. Smoliar
Pages: 155 - 159

Initial thoughts on rapid prototyping techniques
Tamara Taylor, Thomas A. Standish
Pages: 160 - 166

Models as executable designs
Allan M. Stavely
Pages: 167 - 168

Prototyping and small scale software projects
Eugene M. Strand, Warren T. Jones
Pages: 169 - 170

Rapid prototyping of interactive information systems
Anthony I. Wasserman, David T. Shewmake
Pages: 171 - 180

Scale models and rapid prototyping
Mark Weiser
Pages: 181 - 185

**Symposium on High-level Debugging, 1983**

   Debugging methodology: session summary
Wayne C. Gramlich
Pages: 1 - 3

Debugging `level': step-wise debugging
Dick Hamlet
Pages: 4 - 8

Interactive debug requirements
Rich Seidner, Nick Tindall
Pages: 9 - 22

Knowledge-based debugging: session summary
Mark A. Linton, Robert Balzer, R. J. Cunningham, Cordell Green, Robert Sedlmeyer, Elliott Solloway, William Swartout, Pamela Zave
Pages: 23 - 24

Knowledge-based fault localization in debugging: preliminary draft
Robert L. Sedlmeyer, William B. Thompson, Paul E. Johnson
Pages: 25 - 31

Requirements/design debugging: session summary
Peter Bates
Pages: 32 - 33

Generalized path expressions: a high level debugging mechanism
Bernd Bruegge, Peter Hibbard
Pages: 34 - 44

The application of error--sensitive testing strategies to debugging
Lori A. Clarke, Debra J. Richardson
Pages: 45 - 52

An approach to testing specifications
C. Jard, G. v. Bochmann
Pages: 53 - 59

Integrated environments
Insup Lee
Pages: 60 - 62

An experimental debugger in a limited programming environment
Zen Kishimoto
Pages: 63 - 66

A database model of debugging
Michael I. Powell, Mark A. Linton
Pages: 67 - 70

Interactive program execution in Lispeedit
Martin Mikelsons
Pages: 71 - 80

The role of debugging within software engineering environments
Monika A. F. Müllerburg
Pages: 81 - 90

An integrated Lisp programming environment
Harald Wertz
Pages: 91 - 95

Distributed debugging: session summary
Thomas Gross
Pages: 96 - 97

Development of a debugger for a concurrent language
F. Baiardi, N. De Francesco, E. Matteoli, S. Stefanini, G. Vaglini
An approach to high-level debugging of distributed systems: preliminary draft
Peter Bates, Jack C. Wileden
Pages: 107 - 111

Interactive debugging of concurrent programs
Jonice Cynthia Weber
Pages: 112 - 113

Implementation issues: session summary
Janice Cynthia Weber
Pages: 114 - 116

Implementation issues II
Michael S. Kenniston
Pages: 117 - 120

SIMOB: a portable toolbox for observation of SIMULA executions
Knut Barra, Hans Petter Dahle
Pages: 121 - 122

High level language debugging with a compiler
Jeanne Ferrante
Pages: 123 - 129

A systematic approach to advanced debugging: incremental compilation
Peter Fritzson
Pages: 130 - 139

Hardware assisted high level debugging: preliminary draft
W. Morven Gentleman, Henry Hoeksma
Pages: 140 - 144

A real-time microprocessor debugging technique
Charles R. Hill
Pages: 145 - 148

Implementation issues for a source level symbolic debugger
John D. Johnson, Gary W. Kenney
Pages: 149 - 151

High-level debugging assistance via optimizing compiler technology
Karl J. Ottenstein, Linda M. Ottenstein
Pages: 152 - 154

Static analysis of programs as an aid to debugging
Ron Tischler, Robin Schaufler, Charlotte Payne
Pages: 155 - 158

An interactive high-level debugger for control-flow optimized programs
Polle T. Zellweger
Pages: 159 - 172
First ACM SIGSOFT Symposium on the Foundations of Software Engineering, 1993

Algebra and models
C. A. R. Hoare
Pages: 1 - 8

Using style to understand descriptions of software architecture
Gregory Abowd, Robert Allen, David Garlan
Pages: 9 - 20

Mechanisms for generic process support
Robert Balzer, K. Narayanaswamy
Pages: 21 - 32

Fine-grained revision control for collaborative software development
Boris Magnusson, Ulf Asklund, Sten Minör
Pages: 33 - 41

Direct update of data flow representations for a meaning-preserving program restructuring tool
William G. Griswold
Pages: 42 - 55

PLEIADES: an object management system for software engineering environments
Peri Tarr, Lori A. Clarke
Pages: 56 - 70

Towards increased productivity of algorithm implementation
Jiazhen Cai, Robert Paige
Pages: 71 - 78

A practical approach to software engineering using Z and the refinement calculus
Kenneth R. Wood
Pages: 79 - 88

Deriving modular designs from formal specifications
David Carrington, David Duke, Ian Hayes, Jim Welsh
Pages: 89 - 98

Targeting safety-related errors during software requirements analysis
Robyn R. Lutz
Pages: 99 - 106

Does every inspection need a meeting?
Lawrence G. Votta, Jr.
Pages: 107 - 114

Enhancing compositional reachability analysis with context constraints
S. C. Cheung, J. Kramer
A framework for programming and formalizing concurrent objects
Jean Paul Bahsoun, Stephan Merz, Corinne Servieres
Pages: 126 - 137

A logical approach to data structures
Russell Turpin
Pages: 138 - 148

Test case generation by means of learning techniques
F. Bergadano
Pages: 149 - 162

Software measure specification
David A. Gustafson, Joo T. Tan, Perla Weaver
Pages: 163 - 168

Partition testing, stratified sampling, and cluster analysis
Andy Podgurski, Charles Yang
Pages: 169 - 181

Signature matching: a key to reuse
Amy Moormann Zaremski, Jeannette M. Wing
Pages: 182 - 190

Scalable software libraries
Don Batory, Vivek Singhal, Marty Sirkin, Jeff Thomas
Pages: 191 - 199

**4th international software process workshop**

Process modelling in software environments
V. Ashok, J. Ramanathan, S. Sarkar, V. Venugopal
Pages: 36 - 42

Process programming: passing into a new phase
R. Balzer
Pages: 43 - 45

Applying process programming to the model
B. Boeh, F. Belz
Pages: 46 - 56

Activity coordination programs
T. E. Cheatham, Jr.
Pages: 57 - 60

Duplicate instances of elements of a software process model
G. Chroust
Pages: 61 - 64

Design considerations for process-driven software environments
C. Fernstrom
Constructing a dialogic framework for software development
A. Finkelstein, H. Fuks, C. Niskier, M. Sadler
Pages: 68 - 72

A manager/controller for the software development process
C. A. Fritsch, D. L. Perry
Pages: 73 - 75

The process model of the aspect IPSE
P. Hitchcock
Pages: 76 - 78

Probing limits to automation: towards deeper process models
K. E. Huff
Pages: 79 - 81

The software engineering process: definition and scope
W. S. Humphrey
Pages: 82 - 83

Rule-based modelling of the software development process
G. E. Kaiser
Pages: 84 - 86

A hierarchical and functional approach to software process description
T. Katayama
Pages: 87 - 92

Representation formalisms for software process modelling
M. I. Kellner
Pages: 93 - 96

Limits to the mechanization of the detailing step paradigm
C. J. Koomen
Pages: 97 - 102

Enactable models for quantitative evolutionary software processes
L. Krzanik
Pages: 103 - 110

Some reservations on software process programming
M. M. Lehman
Pages: 111 - 112

A functional paradigm for software development
R. MacLean
Pages: 113 - 115

Applying software process models to the full lifecycle is premature
D. Notkin
Pages: 116 - 117
Process programming with Prolog
A. Ohki, K. Ochimizu
Pages: 118 - 121

Automated support for the enactment of rigorously described software processes
L. Osterweil
Pages: 122 - 125

Problems of scale and process models
D. E. Perry
Pages: 126 - 128

State change architecture: a protocol for executable process models
R. W. Phillips
Pages: 129 - 132

Software reuse processes
S. S. Redwine, Jr., W. E. Riddle
Pages: 133 - 135

Describing and acting process models with PML
C. Roberts
Pages: 136 - 141

A specification framework for software processes: formal specification and derivation of information base requirements
D. Rombach
Pages: 142 - 147

Capturing software processes through the generated objects
M. Rueher, D. Ladret, B. Legeard
Pages: 148 - 152

Modelling software evolution: a knowledge-based approach
W. Scacchi
Pages: 153 - 155

Advanced models of the software process
W. L. Sutton
Pages: 156 - 158

Software process models and programs: observations on their nature and context
C. J. Tully
Pages: 159 - 162

Facilitating process prototyping by controlling the impact of change
J. C. Wileden, L. A. Clarke, Alexander L. Wolf
Pages: 163 - 166

A behavioral approach to software process modelling
L. G. Williams
Pages: 167 - 170
Tolerating inconsistency
Robert Balzer
Pages: 41 - 42

5th international software process workshop

Experiences with the spiral model as a process model generator
Barry Boehm, Frank Belz
Pages: 43 - 45

System dynamics models of software developments
David Carr, Ron Koestler
Pages: 46 - 48

Process programming and process models
Thomas E. Cheatham
Pages: 49 - 51

Studying the process of software design teams
Bill Curtis, Diane Walz, Joyce Elam
Pages: 52 - 53

Process programming: a structured multi-paradigm approach could be achieved
W. Deiters, V. Gruhn, W. Schäfer
Pages: 54 - 57

Software process support through software configuration management
Peter P. FEILER
Pages: 58 - 60

A structural framework for the formal representation of cooperation
Anthony Finkelstein
Pages: 61 - 63

The role of formalized domain-specific software frameworks
David Garlan
Pages: 64 - 66

P4: a logic language for process programming
Dennis Heimbigner
Pages: 67 - 70

Software process instantiation and the planning paradigm
Karen E. Huff
Pages: 71 - 73

Modeling implications of the personal software process
Watts S. Humphrey
Pages: 74 - 77

Mechanisms for software process dynamics
Takuya Katayama, Masato Suzuki
Experience with Marvel
Gail E. Kaiser
Pages: 82 - 88

Experience with enactable software process models
Marc I. Kellner
Pages: 85 - 88

Position paper for ISPW
Kouichi Kishida
Pages: 89 - 90

The role of process models in software and systems development and evolution
M. M. Lehman
Pages: 91 - 94

A software process model based on unit workload network
Yohihiro Matsumoto, Kiyoshi Agusa, Tsuneo Ajisaka
Pages: 95 - 97

Law-governed software processes
Naftaly H. Minsky
Pages: 98 - 100

Product-based process models
Ataru T. Nakagawa, Kokichi Futatsugi
Pages: 101 - 105

A method for recording and analyzing design processes
Tsuyoshi Nakajima
Pages: 106 - 108

Experiences with process programming
Leon J. Osterweil
Pages: 109 - 111

Acquiring experiences with executable process models
Maria H. Penedo
Pages: 112 - 115

Software process dependability
Samuel T. Redwine, Jr.
Pages: 116 - 120

Conceptual programming
Steven P. Reiss
Pages: 121 - 123

Dynamics of process models in PML
Clive Roberts, Alun Jones
Pages: 124 - 126
Specification of software process measurement
H. Dieter Rombach
Pages: 127 - 129

Experiences with operational software process modeling in the system factory project
Walt Scacchi
Pages: 130 - 133

OPM: an object process modeling environment
Yasuhiro Sugiyama, Ellis Horowitz
Pages: 134 - 136

The software process as a goal-directed activity
Ian Thomas
Pages: 137 - 139

Experiments with typing in process modeling
Jack C. Wileden
Pages: 140 - 144

Domain understanding and the software process
Pamela Zave
Pages: 145 - 147

An example of software development modeling
Anthony Finkelstein
Pages: 151 - 153

An example P4 process program for Rebus
Dennis Heimbigner
Pages: 154 - 155

GRAPPLE example: processes as plans
Karen E. Huff
Pages: 156 - 158

Marvel strategy language example
Gail E. Kaiser
Pages: 159 - 160

An example of process description in HFSP
Takuya Katayama, Masato Suzuki
Pages: 161 - 162

Software process modeling example
Marc I. Kellner
Pages: 163 - 166

An example of a software process model based on unit workload network
Yoshihiro Matsumoto, Tsuneo Ajisaka
Pages: 167 - 168
Example process program code, coded in Appl/A
Leon Osterweil
Pages: 169 - 170

Describing working environments in OPM
Yasuhiro Sugiyama, Ellis Horowitz
Pages: 171 - 172

An overview of the ALF project
Colin Potts
Page: 175

The Arcadia research project
Arcadia Consortium
Pages: 176 - 177

ESF and software process modeling
W. Schäfer, P. Broekman, L. Hubert, J. Scott
Page: 178

Dynamics of process models in PML
R. A. Snowdon, Clive Roberts
Pages: 179 - 180

Summary of software designer's associates project
Masatoshi Matsuo
Page: 181

Proceedings of Symposium on Software Development Environments, 1 - 5

A small contribution to editing with a syntax directed editor
Marvin V. Zelkowits
Pages: 1 - 6

Techniques for improving language-based editors
J. R. Horgan, D. J. Moore
Pages: 7 - 14

IDEOSY: An Ideographic and Interactive Program Description System
Alessandro Giacalone, Martin C. Rinard, Thomas W. Doeppner, Jr.
Pages: 15 - 20

The Poe language-based editor project
C. N. Fischer, Gregory F. Johnson, Jon Mauney, Anil Pal, Daniel L. Stock
Pages: 21 - 29

Graphical program development with PECAN program development systems
Steven P. Reiss
Pages: 30 - 41
The synthesizer generator
Thomas Reps, Tim Teitelbaum
Pages: 42 - 48

Viewing a programming environment as a single tool
Norman M. Delisle, David E. Menicosy, Mayer D. Schwartz
Pages: 49 - 56

Arcturus: A prototype advanced Ada programming environment
Thomas A. Standish, Richard N. Taylor
Pages: 57 - 64

GNOME: An introductory programming environment based on a family of structure editors
David B. Garlan, Philip L. Miller
Pages: 65 - 72

The SAGA project: A system for software development
Roy H. Campbell, Peter A. Kirslis
Pages: 73 - 80

ISDE: An Interactive Software Development Environment
M. Chesi, E. Dameri, M. P. Franceschi, M. G. Gatti, C. Simonelli
Pages: 81 - 88

The Darwin software-evolution environment
N. Minsky, A. Borgida
Pages: 89 - 95

Phoan: An intelligent system for distributed control synthesis
John DeTreville
Pages: 96 - 103

Computer-Aided Software Engineering in a distributed workstation environment
David B. Leblang, Robert P. Chase, Jr.
Pages: 104 - 112

Preliminary experience from the dice system a distributed incremental compiling environment
Peter Fritzson
Pages: 113 - 123
Monitoring in a software development environment: A relational approach
Richard Snodgrass
Pages: 124 - 131

Implementing relational views of programs
Mark A. Linton
Pages: 132 - 140

Document structure and modularity in mentor
V. Donzeau-Gouge, G. Kahn, B. Lang, B. Mélèse
Pages: 141 - 148

Preliminary experience with a configuration control system for modular programs
J. Estublier, S. Ghoul, S. Krakowiak
Pages: 149 - 156

Generation Scavenging: A non-disruptive high performance storage reclamation algorithm
David Ungar
Pages: 157 - 167

US&R: A new framework for redoing (Extended Abstract)
Jeffrey Scott Vitter
Pages: 168 - 176

The program dependence graph in a software development environment
Karl J. Ottenstein, Linda M. Ottenstein
Pages: 177 - 184

A programming environment for a timeshared system
Richard P. Gabriel, Martin E. Frost
Pages: 185 - 192

A cognitively-based methodology for designing languages/environments/methodologies
Elliot Soloway
Pages: 193 - 196

Prototyping a project master database for software engineering environments
Maria H. Penedo
Pages: 1 - 11
The object management system of PCTE as a software engineering database management system
Ferdinando Gallo, Regis Minot, Ian Thomas
Pages: 12 - 15

Worlds: an organizing structure for object-bases
David S Wile, Dennis G Allard
Pages: 16 - 26

ISTAR—an integrated project support environment
Mark Dowson
Pages: 27 - 33

A programming environment for CSP
Norman Delisle, Mayer Schwartz
Pages: 34 - 41

Integral-C—a practical environment for C programming
Graham Ross
Pages: 42 - 48

Engineering VAX Ada for a multi-language programming environment
Charles Z Mitchell
Pages: 49 - 58

An information workstation for software managers
M A Kret
Pages: 59 - 69

A case study of workstation usage during the early phases of the software development life cycle
Rakesh M Mital, Marcia M Kim, Richard A Berg
Pages: 70 - 76

Rapidly evolving software and the OVERSEE environment
Steven Wartik
Pages: 77 - 83

Experience with a data base of programs
N Belkatir, J Estublier
Pages: 84 - 91

Editing and compiling whole programs
Keith D Copper, Ken Kennedy, Linda Torczon
StarLite, a visual simulation package for software prototyping
Robert P Cook, Rihard J Auletta
Pages: 102 - 110

AWB-ADE: an application development environment for interactive, integrated systems
Carolyn Childs, Filippo I Vokolos
Pages: 111 - 120

Jasmine: a software system modelling facility
Keith Marzullo, Douglas Wiebe
Pages: 121 - 130

A graphical, extensible integrated environment for software development
Anthony I Wasserman, Peter A Pircher
Pages: 131 - 142

Implementing a user interface as a system of attributes
Roger King, Scott Hudson
Pages: 143 - 149

Dost: an environment to support automatic generation of user interfaces
Prasun Dewan, Marvin Solomon
Pages: 150 - 159

A structural approach to the maintenance of structure-oriented environments
Barbara J Staudt, Charles W Krueger, David Garlan
Pages: 160 - 170

An incremental algorithm for software analysis
Martin Carroll, Barbara G Ryder
Pages: 171 - 179

Efficient recompilation of module interfaces in a software development environment
Hausi A Muller, Robert Hood, Ken Kennedy
Pages: 180 - 189

On the structure of structure-oriented editors for different applications
G Engels, M Nagl, W Schafer
Pages: 190 - 198
A methodology for evaluating environments
Nelson H Weiderman, A Nico Habermann, Mark W Borger, Mark H Klein
Pages: 199 - 207

The kernel of a generic software development environment
A van Lamsweerde, M Buyse, B Delcourt, E Delor, M Ervier, M C Schayes
Pages: 208 - 217

A foundation for programming environments
J H Reppy, E R Gansner
Pages: 218 - 227

Foundations for the Arcadia environment architecture
Richard N. Taylor, Frank C. Belz, Lori A. Clarke, Leon Osterweil, Richard W. Selby, Jack C. Wileden, Alexander L. Wolf, Michael Young
Pages: 1 - 13

Centaur: the system
P. Borras, D. Clement, Th. Despeyroux, J. Incerpi, G. Kahn, B. Lang, V. Pascual
Pages: 14 - 24

The ergo support system: an integrated set of tools for prototyping integrated environments
Peter Lee, Frank Pfenning, Gene Rollins, William Scherlis
Pages: 25 - 34

The Software Life Cycle Support Environment (SLCSE): a computer based framework for developing software systems
Tom Strelich
Pages: 35 - 44

(ISEF): an integrated industrial-strength software engineering framework
Shaye Koenig
Pages: 45 - 54

The workshop system: a practical knowledge-based software environment
Geoffrey M. Clemm
Pages: 55 - 64

A software development environment for law-governed systems
Naftaly H. Minsky, David Rozenshtein
Pages: 65 - 75

Interacting with an active, integrated environment
Thomas Rodden, Pete Sawyer, Ian Sommerville
Pages: 76 - 84

Object database support for a software project management environment
Lung-Chun Liu, Ellis Horowitz
A plan-based intelligent assistant that supports the software development
Karen E. Huff, Victor R. Lesser
Pages: 97 - 106

The relationship between software development environments and the
software process
David Notkin
Pages: 107 - 109

The ergo attribute system
Robert L. Nord, Frank Pfenning
Pages: 110 - 120

Graph attribution as a specification paradigm
Bowen Alpern, Alan Carle, Barry Rosen, Peter Sweeney, Kenneth Zadeck
Pages: 121 - 129

Pgraphite: an experiment in persistent typed object management
Jack C. Wileden, Alexander L. Wolf, Charles D. Fisher, Peri L. Tarr
Pages: 130 - 142

User interface support for the integration of software tools: an iconic
model of interaction
Michel Beaudouin-Lafon
Pages: 143 - 152

TEAM: a support environment for testing, evaluation, and analysis
Lori A. Clarke, Debra J. Richardson, Steven J. Zeil
Pages: 153 - 162

Transformational derivation of programs using the focus system
Uday S. Reddy
Pages: 163 - 172

Extended programming in the large in a software development environment
Claus Lewerentz
Pages: 173 - 182

A unifying model for consistent distributed software development
environments
J. Walpole, G. S. Blair, J. Malik, J. R. Nicol
Pages: 183 - 190

An integrated toolset for engineering software configurations
Axel Mahler, Andreas Lampen
Pages: 191 - 200

Version management in Gypsy
Ellis S. Cohen, Dilip A. Soni, Raimund Gluecker, William M. Hasling,
Robert W. Schwanke, Michael E. Wagner
Pages: 201 - 215
A graph transform model for configuration management environments
Dennis Heimbigner, Steven Krane
Pages: 216 - 225

Object management issues for software engineering environments workshop report
Maria H. Penedo, Erhard Ploedereder, Ian Thomas
Pages: 226 - 234

An overview of DOD-STD-1838A (proposed) the common APSE interface set: revision
Robert Munck, Patricia Oberndorf, Erhard Ploedereder, Richard Thall
Pages: 235 - 247

An overview of PCTE and PCTE+
Gerard Boudier, Ferdinando Gallo, Regis Minot, Ian Thomas
Pages: 248 - 257

Low-cost, adaptable tool integration policies for integrated environments
David Garlan, Ehsan Ilias
Pages: 1 - 10

A distributed architecture for programming environments
Dominique Clément
Pages: 11 - 21

Reconciling environment integration and component independence
Kevin Sullivan, David Notkin
Pages: 22 - 33

Consistency management in a project management assistant
Xiaolei Qian, Richard Jullig, Marilyn Daum
Pages: 34 - 43

Configuration management by consensus: an application of law-governed systems
Naftaly H. Minsky, David Rozenshtein
Pages: 44 - 55

Coping with changes in an object management system based on attribute grammars
Lichao Tan, Yoichi Shinoda, Takuya Katayama
Pages: 56 - 65

An extensible programming environment for Modula-3
Mick Jordan
Pages: 66 - 76

The Pan language-based editing system for integrated development
Robert A. Ballance, Susan L. Graham, Michael L. Van De Vanter
Pages: 77 - 93

A multi-formalism specification environment
Edward A. Ipser, David S. Wile, Dean Jacobs
Pages: 94 - 106
Reusing software developments
Allen Goldberg
Pages: 107 - 119

An environment for documenting software features
C. Frederick Hart, John J. Shilling
Pages: 120 - 132

A program integration algorithm that accommodates semantics-preserving transformations
Wuu Yang, Susan Horwitz, Thomas Reps
Pages: 133 - 143

Fine grained data management to achieve evolution resilience in a software development environment
Richard Snodgrass, Karen Shannon
Pages: 144 - 156

A procedure and tools for transition engineering
Abe Lockman, John Salasin
Pages: 157 - 172

Configuration editing, generation and test within working contexts
Hans-Ulrich Kobialka
Pages: 173 - 182

Software process enactment in Oikos
V. Ambriola, P. Ciancarini, C. Montangero
Pages: 183 - 192

Managing software processes in the environment MELMAC
Volker Gruhn
Pages: 193 - 205

Language constructs for managing change in process-centered environments
Stanley M. Sutton, Jr., Dennis Heimbigner, Leon J. Osterweil
Pages: 206 - 217

Support for change in RPDE3
Harold Ossher, William Harrison
Pages: 218 - 228

Integrating configuration management into a generic environment
Axel Mahler, Andreas Lampen
Pages: 229 - 237

A query service for a software engineering database system
Mohamed Tedjini, Ian Thomas, Guy Benoliel, Fernando Gallo, Régis Minot
Pages: 238 - 248

HyperWeb: a framework for hypermedia-based environments
James C. Ferrans, David W. Hurst, Michael A. Sennett, Burton M. Covnot, Wenguang Ji, Peter Kajka, Wei Ouyang
Pages: 1 - 10
Supporting collaborative software development with ConversationBuilder
Simon M. Kaplan, William J. Tolone, Alan M. Carroll, Douglas P. Bogia, Celsina Bignoli
Pages: 11 - 20

Supporting cooperation in the Marvel process-centered SDE
Naser S. Barghouti
Pages: 21 - 31

obTIOS: a CAx-framework service for building concurrent engineering environments
Remco Quester
Pages: 32 - 40

An empirical investigation of menu design in language-based editors
Mark A. Toleman, Jim Welsh, Alan J. Chapman
Pages: 41 - 46

An empirical study of multiple-view software development
Scott Meyers, Steven P. Reiss
Pages: 47 - 57

Consistency in software system development: framework, model, techniques & tools
Torben Mejlvang Hangensen, Bent Bruun Kristensen
Pages: 58 - 67

A transformational approach to generating application-specific environments
David Garlan, Linxi Cai, Robert L. Nord
Pages: 68 - 77

Operation-based merging
Ernst Lippe, Norbert van Oosterom
Pages: 78 - 87

A reverse engineering environment based on spatial and visual software interconnection models
H. A. Müller, S. R. Tilley, M. A. Orgun, B. D. Corrie, N. H. Madhavji
Pages: 88 - 98

Integrating the heterogeneous control properties of software modules
Elizabeth L. White, James M. Purtilo
Pages: 99 - 108

Developing and integrating ProDAG in the Arcadia environment
Debra J. Richardson, T. Owen O'Malley, Cynthia Tittle Moore, Stephanie Leif Aha
Pages: 109 - 119

Real theorem provers deserve real user-interfaces
Laurent Théry, Yves Bertot, Gilles Kahn
Pages: 120 - 129
The Proteus presentation system
Susan L. Graham, Michael A. Harrison, Ethan V. Munson
Pages: 130 - 138

An analysis technique for examining integration in a project support environment
Alan W. Brown, Peter H. Feiler
Pages: 139 - 148

An architecture for multi-user software development environments
Israel Z. Ben-Shaul, Gail E. Kaiser, George T. Heineman
Pages: 149 - 158

The ProcessWall: a process state server approach to process programming
Dennis Heimbigner
Pages: 159 - 168

Issues encountered in building a flexible software development environment: lessons from the Arcadia project
R. Kadia
Pages: 169 - 180

Symposium on Testing and Verification / International Symposium on Software Testing and Analysis

Validating programs without specifications
W. Howden
Pages: 2 - 9

Real-time interval logic for reasoning about executions of real-time programs
R. Razouk, M. Gorlick
Pages: 10 - 19

Verified compilation in micro-Gypsy
W. Young
Pages: 20 - 26

Theoretical comparison of testing methods
R. Hamlet
Pages: 28 - 37

Some observations on partition testing
B. Jeng, E. Weyuker
Pages: 38 - 47

Reducing the effects of infeasible paths in branch testing
D. Yates, N. Malevris
Pages: 48 - 54

PMothra: scheduling mutants for execution on a hypercube
B. Choi, A. Mathur, B. Pattison
Cesar: a static sequencing constraint analyzer
K. Olender, L. Osterweil
Pages: 66 - 74

Automating software analysis and testing using a program transformation system
G. Kotik, L. Markosian
Pages: 75 - 84

Approaches to specification-based testing
D. Richardson, O. O'Malley, C. Tittle
Pages: 86 - 96

Module test case generation
D. Hoffman, C. Brealey
Pages: 97 - 102

A practical approach to semantic configuration management
M. Moriconi
Pages: 103 - 113

The logic of propagation in the inscape environment
D. Perry
Pages: 114 - 121

Experiments in automated analysis of concurrent software systems
G. Avrunin, J. Wileden, L. Dillon
Pages: 124 - 130

The coupling effect: fact or fiction
A. Offutt
Pages: 131 - 140

Experimental comparison of three system test strategies preliminary report
J. Rowland, Y. Zuyuan
Pages: 141 - 149

Preliminary summary: FM89 assessment of formal methods for trustworthy computer systems
S. Gerhart
Pages: 152 - 155

Interprocedual data flow testing
Mary Jean Harrold, Mary Lou Soffa
Pages: 158 - 167

The implications of program dependencies for software testing, debugging, and maintenance
A. Podgurski, L. Clarke
Pages: 168 - 178
Estimating the number of test cases required to satisfy the all-du-paths testing criterion
J. Bieman, J. Schultz
Pages: 179 - 186

Testing the correctness of tasking supervisors with TSL specifications
D. Rosenblum, D. Luckham
Pages: 187 - 196

Integrated concurrency analysis in a software development environment
M. Young, R. Taylor, K. Forester, D. Brodbeck
Pages: 200 - 209

Automatic generation of test scripts from formal test specifications
M. Balcer, W. Hasling, T. Ostrand
Pages: 210 - 218

Integrating testing techniques through process programming
D. Richardson, S. Aha, L. Osterweil
Pages: 219 - 228

Comparison of program testing strategies
Elaine J. Weyuker, Stewart N. Weiss, Dick Hamlet
Pages: 1 - 10

Reliability, sampling, and algorithmic randomness
Andy Podgurski
Pages: 11 - 20

Data flow analysis of concurrent systems that use the rendezvous model of synchronization
Douglas Long, Lori A. Clarke
Pages: 21 - 35

Concurrency analysis in the presence of procedures using a data-flow framework
Evelyn Duesterwald, Mary Lou Soffa
Pages: 36 - 48

Compositional reachability analysis using process algebra
Wei Jen Yeh, Michal Young
Pages: 49 - 59

Dynamic slicing in the presence of unconstrained pointers
Hiralal Agrawal, Richard A. DeMillo, Eugene H. Spafford
Pages: 60 - 73

Data flow-based test adequacy analysis for languages with pointers
Thomas J. Ostrand, Elaine J. Weyuker
Pages: 74 - 86

Data flow coverage and the C language
J. R. Horgan, S. London
Pages: 87 - 97
Loop monotonic computations: an approach for the efficient run-time detection of races
Rajiv Gupta, Madalene Spezialetti
Pages: 98 - 111

Executing formal specifications: the ASTRAL to TRIO translation approach
Carlo Ghezzi, Richard A. Kennerer
Pages: 112 - 122

Run-time consistency checking of algebraic specifications
Sriram Sankar
Pages: 123 - 129

The effects of optimizing transformations on data-flow adequate test sets
Mary Jean Harrold
Pages: 130 - 138

Interprocedural Def-Use associations in C programs
Hemant D. Pande, William Landi
Pages: 139 - 153

An experimental comparison of the effectiveness of the all-uses and all-edges adequacy criteria
Phyllis G. Frankl, Stewart N. Weiss
Pages: 154 - 164

Case studies on testing object-oriented programs
Roong-Ko Doong, Phyllis G. Frankl
Pages: 165 - 177

Experiments with an improved constrained expression toolset
George S. Avrunin, Ugo A. Buy, James C. Corbett, Laura K. Dillon, Jack C. Wileden
Pages: 178 - 187

Formal methods versus software engineering: Is there a conflict
William D. Young
Pages: 188 - 189

The weak mutation hypothesis
Brian Marick
Pages: 190 - 199

How strong is weak mutation?
A. Jefferson Offutt, Stephen D. Lee
Pages: 200 - 213

Designing programs to check their work (abstract)
Manuel Blum
Page: 1

Specifying operational profiles for modules
Denise M. Woit
Test template framework: a specification-based testing case study  
P. Stocks, D. Carrington  
Pages: 11 - 18

Abstract analysis with aspect  
Daniel Jackson  
Pages: 19 - 27

Analyzing refinements of state based specifications: the case of TB nets  
Miguel Felder, Carlo Ghezzi, Mauro Pezzè  
Pages: 28 - 39

Metric-driven reengineering for static concurrency analysis  
David L. Levine, Richard N. Taylor  
Pages: 40 - 50

Using state space reduction methods for deadlock analysis in Ada tasking  
S. Duri, U. Buy, R. Devarapalli, S. M. Shatz  
Pages: 51 - 60

An integrated general purpose automated test environment  
Peter A. Vogel  
Pages: 61 - 69

Implementation-based analysis and testing of Prolog programs  
B. Belli, O. Jack  
Pages: 70 - 80

Structured IV & V for the space shuttle flight software  
Ronald W. Modes  
Page: 81

Load testing software using deterministic state testing  
Alberto Avritzer, Brian Larson  
Pages: 82 - 88

Faults on its sleeve: amplifying software reliability testing  
Dick Hamlet, Jeff Voas  
Pages: 89 - 98

STATEMATE applied to statistical software testing  
P. Thévenod-Fosse, N. Waeselynck  
Pages: 99 - 109

A practical technique for bounding the time between events in concurrent real-time systems  
J. Corbett, G. Avrunin  
Pages: 110 - 116

Analyzing timing requirements  
Joanne M. Atlee, John Gannon  
Pages: 117 - 127
The composability of ASTRAL realtime specifications
Alberto Coen-Porisini, Richard A. Kemmerer
Pages: 128 - 138

Mutation analysis using mutant schemata
Roland H. Untch, A. Jefferson Offutt, Mary Jean Harrold
Pages: 139 - 148

Improved serial algorithms for mutation analysis
Stewart N. Weiss, Vladimir N. Fleyshgakker
Pages: 149 - 158

Exploitation of software test technology
Edward F. Miller
Page: 159

Efficient construction of program dependence graphs
Mary Jean Harrold, Brian Malloy, Gregg Rothermel
Pages: 160 - 170

Dynamic impact analysis: a cost-effective technique to enforce error-propagation
Tarak Goradia
Pages: 171 - 181

An information flow model of fault detection
Margaret C. Thompson, Debra J. Richardson, Lori A. Clarke
Pages: 182 - 192

ACM SIGSOFT '91 Conference on Software for Critical Systems

Formal verification of algorithms for critical systems
John Rushby, Friedrich von Henke
Pages: 1 - 15

State-based model checking of event-driven system requirements
Joanne Atlee, John Gannon
Pages: 16 - 28

Rigorous development using RAISE
Bent Dandanell
Pages: 29 - 43

Specifying and verifying requirements of real-time systems
Kirsten M. Hansen, Anders P. Ravn, Hans Rischel
Pages: 44 - 54

A systematic kernel development
The infeasibility of experimental quantification of life-critical software reliability
Ricky W. Butler, George B. Finelli
Pages: 66 - 76

Assessing the fault-detecting ability of testing methods
Phyllis G. Frankl, Elaine J. Weyuker
Pages: 77 - 91

An exception handling model for parallel programming and its verification
Valérie Issarny
Pages: 92 - 100

A real-time transition model for analyzing behavioral compatibility of telecommunications services
E. Jane Cameron, Yow-Jian Lin
Pages: 101 - 111

Programming and verifying critical systems by means of the synchronous data-flow language LUSTRE
Christophe Ratel, Nicolas Halbwachs, Pascal Raymond
Pages: 112 - 119

Stepwise design of real-time systems
Reino Kurki-Suonio
Pages: 120 - 131

On satisfying timing constraints in hard-real-time systems
Jia Xu, David Lorge Parnas
Pages: 132 - 146

Automated analysis of bounded response time for two NASA expert systems
Chih-Kan Wang, Duu-Chung Tsou, Rwo-Hsi Wang, James C. Browne, Aloysius K. Mok
Pages: 147 - 161