Re-animation of computer programs
Meijer, F.J.

Citation for published version (APA):
Meijer, F. J. (2001). Re-animation of computer programs
INDEX

8080 71  
8086 processor 33

absolute address 53  
accumulator 72  
ADA 37  
adaptation 28  
agents 20  
ALGOL 12, 23  
algorithm 12  
Allen 41  
ambiguity problem 43-47, 57, 69  
Andersen 32  
application program 49  
architecturally correct 57, 60, 84  
ASCII code 50, 52, 58  
assembly language 22-25, 38, 48, 62, 72  
ATARI 62

Babbage 37  
Backhouse 19-20  
Barbe 38  
BASIC 33, 71  
BBC's Color Computer 14  
Bemer 37  
Biggerstaff 14  
bit strings 16, 43-53, 71, 79  
Blok 72  
branch addresses 51, 74  
branch instruction 47-51, 65  
breakpoints 71  
Brown 25, 42  
C++ 22  
calculating machine 20  
Calkin, Clark A. 90  
call to subroutine 61  
calls to the operating system 77  
Catell 21  
Cathey 71  
CCSO 90  
CD rom 13  
Ceruzzi, P.E. 85  
Chae Woo Yoo 40, 58  
Cheng 39  
Claus 21  
cloning 25  
COBOL 22  
code disassembly 27  
code generation 72  
code optimization 40  
code reconstruction 27, 86, 88  
code reconstruction problem 14, 19-29, 83, 84  
code transformation 19  
code transformation problem 16, 19-23, 47, 56  
Coleman 34-35  
color monitors 13  
Commodore 64 14  
competence enhancement 86  
compiler 15, 23, 25  
computer language 12, 26  
computer science 14  
computer-assisted learning 13  
conditional branches 51, 75  
control operands 45  
Conway 22, 35  
Cowell 42  
CP/M 15, 33, 62, 66, 71, 78-82, 88
Crockett 38
Cullough 14
cursor 81

Dahlstrand 14, 21, 37
data 11, 40, 79
data description language 38
data memory 55, 75
datum 49
debugger 66-67, 71, 77
decompile 40
decomposition 43
detranslation 40
Digital Equipment Corporation 12
Digital Equipment, Maynard, U.S.A. 11
Dijkstra 58-61
disassembler 40, 79
Druseikis 34
dynamic analysis 40, 48, 49, 88

emulation 13, 19-28, 31, 40, 83
emulators 34, 38
entry point 77
error condition 40
error free 57, 60, 84
error message 75
event sequence 19-28, 40, 43, 53, 72, 73, 84
events 20-23
executable code 53
execution time 78
emulator 72

Fairfield 32, 58
Feurer 14
first order error 61-65, 74, 77
Fisher 38
Fitz 38
FOCAL 12
FORTH 33
genenerated code 52
graphical user interfaces 32
Grass, Eric 90

Haaster, van 87
Hamlet 39
Haralick 39
hardware 12
hardware problem 24
hardware requirements 66
hexadecimal value 50, 52
higher-level language 12, 14-15, 22-23, 25, 33, 35, 54, 62
Horspool 14, 21, 26, 40, 44, 57, 58
Hull, R. 85
human activity language 22
human agents 22
I/O devices 77
IBM 34
icons 13
immediate datum 45
indirect address 45
information 19
Inglis 58
instructions 16, 40, 43, 49, 79
interpreter 12, 25
intervention 16

JANUS 35
JAVA 22
jump table 62-65

Kaindl 37, 38
Kalaidjieva 27
kernel 39
Kesselmeier 39
keyboard 87
Kochenuhger 20
Krueger 39

labels 74
language level 22
large-scale integration 11
Lavington 19
Leaume 38
Liebenau 19-20
LINUX 14, 89
INDEX

LISP 22
location 50

Lovelace 37

machine code 15, 40, 44, 62, 71, 88
machine language 15, 16, 22, 25, 85
maintenance 11, 59
manual intervention 52, 58
Marovac 14, 21, 40, 44, 57-58
May 34, 41, 44
Mc Carthy 36
Mealy 36
memory 43, 80
memory address 45
memory locations 49
menus 13, 81
microprocessor 38
Microsoft 85
migrability 24
mnemonic code 24, 52
modules 39, 41
Morenoff 36
Morovac 26
mouse 13, 81, 87
Multi Arcade Machine Emulator 14

natural language 22
non–documented calls 64

object code 24, 80
object computer 21, 25-26, 43, 53, 66, 77
obsolete programs 16
operands 16, 21, 45
operating system 13, 25, 46, 49, 71, 77
operators 21, 45
Opler 44
order of the error 77
output 61
overlay 80

PDP/8 11
Pellisier Gart 38
Perlis 14
personal computers 46

PILOT 13
platform 13, 33, 39
portability 32, 39
problem operands 45
problem solving 22
problem–oriented languages 35
procedure 72
process 11
processing speed 29
processor 25, 86
program 23, 28
program design 23, 28
program flow 46, 48, 71, 75
programming 11
programming errors 12
programming rules 54
programming standards 37
programming technique 12

random access 64
re–engineering 39, 83
re–use 39, 83
read only memory (ROM) 33
reconstructed code 53, 63
reconstruction problem 41, 44
registers 38, 49, 53, 63, 71-73
reprogramming 38
return address 63
reverse engineering 39
routines 39
run time system 63, 75-76

Sable 38
SCfT 14
second order error 61, 62, 77
Seigne, Jean–Louis 90
self–modifying code 57
Skelton 24
software 12, 19
software error 59
Software Migrations Ltd 90
software module 61
source code 15, 20, 24, 40, 44, 71, 83, 85, 89
source computer 53, 66, 77, 79
Source Recovery Company 89
stack pointer 63
static analysis 40, 44, 47, 52
Steel Jr, T.B. 22
storage 11
Strong 35, 38
subroutine 78
support system 87
Swift, Jonathan 35
system calls 55, 64, 70, 71
Tanenbaum 34-35
target address 51, 77
Telnet 13
terminal 13
test set 60
Texas Instruments 14
The Online Software Museum 13
The Operating System (TOS) 66
third order error 61, 77
Thomas 42
Ticalc 89
trace flag 75
tracer 72
transfer 59
transfer or migration problem 24
transferability 24, 38
transformation 13-14
transformation package 49
transformation problem 20, 25, 26, 69
transformation process 52
transformed code 46, 73, 75
transforming software 24
translation back 44
Turcio 20

Ulman 39
UNCOL 34-36
universal assembly language 38
universal higher level programming language 22
universal operating system 39
Universiteit van Amsterdam 11

University of Queensland 89
unused code 52
user intervention 52
user language 21, 87
user-friendly languages 22, 85
utility programs 12

Waite 35
Waites 25
Wallis 37
Walraet 12, 14
Ward 38
word processor 46
working environment 71

Yoo 14, 26, 44
Young, Paul 90

Z80 15, 33, 71