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

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

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
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
Blokh 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
compiler 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
exmulator 72

Fairfield 32, 58
Feurer 14
first order error 61-65, 74, 77
Fisher 38
Fitz 38
FOCAL 12
FORTH 33

generated 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
Kalaidjiev 27
kernel 39
Kesselmeier 39
keyboard 87
Kochenburger 20
Krueger 39

labels 74
language level 22
large-scale integration 11
Lavington 19
Leaene 38
Liebenau 19-20
LINUX 14, 89
<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LISP</td>
<td>22</td>
</tr>
<tr>
<td>location</td>
<td>50</td>
</tr>
<tr>
<td>Lovelace</td>
<td>37</td>
</tr>
<tr>
<td>machine code</td>
<td>15, 40, 44, 62, 71, 88</td>
</tr>
<tr>
<td>machine language</td>
<td>15, 16, 22, 25, 85</td>
</tr>
<tr>
<td>maintenance</td>
<td>11, 59</td>
</tr>
<tr>
<td>manual intervention</td>
<td>52, 58</td>
</tr>
<tr>
<td>Marovac</td>
<td>14, 21, 40, 44, 57-58</td>
</tr>
<tr>
<td>May</td>
<td>34, 41, 44</td>
</tr>
<tr>
<td>McCarthy</td>
<td>36</td>
</tr>
<tr>
<td>Mealy</td>
<td>36</td>
</tr>
<tr>
<td>memory</td>
<td>43, 80</td>
</tr>
<tr>
<td>memory address</td>
<td>45</td>
</tr>
<tr>
<td>memory locations</td>
<td>49</td>
</tr>
<tr>
<td>menus</td>
<td>13, 81</td>
</tr>
<tr>
<td>microprocessor</td>
<td>38</td>
</tr>
<tr>
<td>Microsoft</td>
<td>85</td>
</tr>
<tr>
<td>migrability</td>
<td>24</td>
</tr>
<tr>
<td>mnemonic code</td>
<td>24, 52</td>
</tr>
<tr>
<td>modules</td>
<td>39, 41</td>
</tr>
<tr>
<td>Morenoff</td>
<td>36</td>
</tr>
<tr>
<td>Morovac</td>
<td>26</td>
</tr>
<tr>
<td>mouse</td>
<td>13, 81, 87</td>
</tr>
<tr>
<td>Multi Arcade Machine Emulator</td>
<td>14</td>
</tr>
<tr>
<td>natural language</td>
<td>22</td>
</tr>
<tr>
<td>non–documented calls</td>
<td>64</td>
</tr>
<tr>
<td>object code</td>
<td>24, 80</td>
</tr>
<tr>
<td>object computer</td>
<td>21, 25-26, 43, 53, 66, 77</td>
</tr>
<tr>
<td>obsolete programs</td>
<td>16</td>
</tr>
<tr>
<td>operands</td>
<td>16, 21, 45</td>
</tr>
<tr>
<td>operating system</td>
<td>13, 25, 46, 49, 71, 77</td>
</tr>
<tr>
<td>operators</td>
<td>21, 45</td>
</tr>
<tr>
<td>Opler</td>
<td>44</td>
</tr>
<tr>
<td>order of the error</td>
<td>77</td>
</tr>
<tr>
<td>output</td>
<td>61</td>
</tr>
<tr>
<td>overlay</td>
<td>80</td>
</tr>
<tr>
<td>PDP/8</td>
<td>11</td>
</tr>
<tr>
<td>Pellisier Gart</td>
<td>38</td>
</tr>
<tr>
<td>Perlis</td>
<td>14</td>
</tr>
<tr>
<td>personal computers</td>
<td>46</td>
</tr>
<tr>
<td>PILOT</td>
<td>13</td>
</tr>
<tr>
<td>platform</td>
<td>13, 33, 39</td>
</tr>
<tr>
<td>portability</td>
<td>32, 39</td>
</tr>
<tr>
<td>problem operands</td>
<td>45</td>
</tr>
<tr>
<td>problem solving</td>
<td>22</td>
</tr>
<tr>
<td>problem–oriented languages</td>
<td>35</td>
</tr>
<tr>
<td>procedure</td>
<td>72</td>
</tr>
<tr>
<td>process</td>
<td>11</td>
</tr>
<tr>
<td>processing speed</td>
<td>29</td>
</tr>
<tr>
<td>processor</td>
<td>25, 86</td>
</tr>
<tr>
<td>program</td>
<td>23, 28</td>
</tr>
<tr>
<td>program design</td>
<td>23, 28</td>
</tr>
<tr>
<td>program flow</td>
<td>46, 48, 71, 75</td>
</tr>
<tr>
<td>programming</td>
<td>11</td>
</tr>
<tr>
<td>programming errors</td>
<td>12</td>
</tr>
<tr>
<td>programming rules</td>
<td>54</td>
</tr>
<tr>
<td>programming standards</td>
<td>37</td>
</tr>
<tr>
<td>programming technique</td>
<td>12</td>
</tr>
<tr>
<td>random access</td>
<td>64</td>
</tr>
<tr>
<td>re–engineering</td>
<td>39, 83</td>
</tr>
<tr>
<td>re–use</td>
<td>39, 83</td>
</tr>
<tr>
<td>read only memory (ROM)</td>
<td>33</td>
</tr>
<tr>
<td>reconstructed code</td>
<td>53, 63</td>
</tr>
<tr>
<td>reconstruction problem</td>
<td>41, 44</td>
</tr>
<tr>
<td>registers</td>
<td>38, 49, 53, 63, 71-73</td>
</tr>
<tr>
<td>reprogramming</td>
<td>38</td>
</tr>
<tr>
<td>return address</td>
<td>63</td>
</tr>
<tr>
<td>reverse engineering</td>
<td>39</td>
</tr>
<tr>
<td>routines</td>
<td>39</td>
</tr>
<tr>
<td>run time system</td>
<td>63, 75-76</td>
</tr>
<tr>
<td>Sable</td>
<td>38</td>
</tr>
<tr>
<td>SCfT</td>
<td>14</td>
</tr>
<tr>
<td>second order error</td>
<td>61, 62, 77</td>
</tr>
<tr>
<td>Seigne, Jean-Louis</td>
<td>90</td>
</tr>
<tr>
<td>self-modifying code</td>
<td>57</td>
</tr>
<tr>
<td>Skelton</td>
<td>24</td>
</tr>
<tr>
<td>software</td>
<td>12, 19</td>
</tr>
<tr>
<td>software error</td>
<td>59</td>
</tr>
<tr>
<td>Software Migrations Ltd</td>
<td>90</td>
</tr>
<tr>
<td>software module</td>
<td>61</td>
</tr>
<tr>
<td>source code</td>
<td>15, 20, 24, 40, 44, 71, 83, 85, 89</td>
</tr>
</tbody>
</table>

**INDEX**
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