A Software Physics Analysis of Akiyama's Debugging Data

Yasao Funami

M. H. Halstead

Report Number:
75-144
A SOFTWARE PHYSICS ANALYSIS OF
AKIYAMA'S DEBUGGING DATA

Yasao Funami
and
M. H. Halstead

May 1975

CSD-TR 144
F. Akiyama [1] has published a careful study of the number of bugs which occurred in the programming of each of the nine modules of a 100 man-month software system called SAMPLE. All of his observed data are reproduced in Table 1. In the case of Module MC, 53 bugs were reported before machine runs were obtained, and these have been included.

Table 1. Akiyama's Observations

<table>
<thead>
<tr>
<th>Program Module</th>
<th>MA</th>
<th>MB</th>
<th>MC</th>
<th>MD</th>
<th>ME</th>
<th>MF</th>
<th>MG</th>
<th>MH</th>
<th>MX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Steps (S)</td>
<td>4032</td>
<td>1329</td>
<td>5453</td>
<td>1674</td>
<td>2051</td>
<td>2513</td>
<td>699</td>
<td>3792</td>
<td>3412</td>
</tr>
<tr>
<td>Decisions (D)</td>
<td>372</td>
<td>215</td>
<td>552</td>
<td>111</td>
<td>315</td>
<td>217</td>
<td>104</td>
<td>233</td>
<td>416</td>
</tr>
<tr>
<td>Calls (J)</td>
<td>283</td>
<td>44</td>
<td>362</td>
<td>130</td>
<td>197</td>
<td>186</td>
<td>32</td>
<td>110</td>
<td>230</td>
</tr>
<tr>
<td>Number of Bugs (B)</td>
<td>102</td>
<td>18</td>
<td>146</td>
<td>26</td>
<td>71</td>
<td>37</td>
<td>16</td>
<td>50</td>
<td>80</td>
</tr>
</tbody>
</table>

In presenting his data, Akiyama reported that the coefficient of correlation between number of bugs and number of program steps was 0.83, while the correlation between bugs and the sum of decisions plus calls was much higher, at 0.92.

An interesting and quantitative explanation of this result is provided by the theory of software physics [2]. According to that theory, the number of effective mental discriminations, \(E\), required for the implementation of a program is given by:

\[
E = \frac{V/L}{\left(\eta_1 \eta_2/n_1 N_2\right)}
\]  

(1)
where:

- \( V \) = Program volume.
- \( L \) = Program level.
- \( \eta^*_1 = 2 \) = Unique operators required by a call.
- \( \eta_1 \) = Unique operators used in the program.
- \( \eta_2 \) = Unique operands used in the program.
- \( N_2 \) = Total usage of operands.
- \( N \) = Total usage of operands and operators.
- \( n = n_1 + n_2 \)

While Akiyama's data do not include these parameters directly, they do supply observations from which they may be estimated. If we assume that each of the S machine language steps includes one operator and one operand, then:

\[ N_2 = S \quad (2) \]

and \( N = 2S \). \quad (3) \]

The number of unique operators, \( n_1 \), is composed of three classes of operators. The first is the number of distinct operators used from the machine's repertoire of instructions. For large programs, this component may be roughly approximated as an octal hundred. Second is the number of distinct operations provided by functions or subroutines. This component should correspond to item J in Table 1. Finally, each transfer to a unique location has been shown by Bulut [3] to contribute directly to \( n_1 \). Since the number of transfers implied by item D in Table 1 do not each involve transfer to a unique location, only a fraction, perhaps one third, should contribute to \( n_1 \). We then have, roughly:

\[ n_1 = D/3 + J + 64 \quad (4) \]

At this point, we need only an estimate of \( n_2 \) to be able to calculate \( E \). From the length equation as presented by Halstead and Bayer [4] and independently validated by Bohrer [5].
\[ N = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2 \]  

It is possible to find \( \eta_2 \) when \( \eta_1 \) and \( N \) are known.

Using equations 1 through 5, the data of Table 1 yield the results shown in Table 2.

Table 2. Software Physics Parameters derived from Table 1.

<table>
<thead>
<tr>
<th>Module</th>
<th>MA</th>
<th>MB</th>
<th>MC</th>
<th>HD</th>
<th>ME</th>
<th>MF</th>
<th>MG</th>
<th>MH</th>
<th>MX</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>8064</td>
<td>2658</td>
<td>10906</td>
<td>3348</td>
<td>4102</td>
<td>5026</td>
<td>1398</td>
<td>7584</td>
<td>6824</td>
</tr>
<tr>
<td>( N_2 )</td>
<td>4032</td>
<td>1329</td>
<td>5453</td>
<td>1674</td>
<td>2051</td>
<td>2513</td>
<td>699</td>
<td>3792</td>
<td>3412</td>
</tr>
<tr>
<td>( \eta_1 )</td>
<td>471</td>
<td>180</td>
<td>610</td>
<td>231</td>
<td>366</td>
<td>322</td>
<td>131</td>
<td>252</td>
<td>433</td>
</tr>
<tr>
<td>( \eta_2 )</td>
<td>442</td>
<td>176</td>
<td>574</td>
<td>201</td>
<td>138</td>
<td>287</td>
<td>76</td>
<td>603</td>
<td>357</td>
</tr>
<tr>
<td>( E ) (Millions)</td>
<td>170.3</td>
<td>15.3</td>
<td>322.6</td>
<td>28.2</td>
<td>100.2</td>
<td>65.5</td>
<td>6.5</td>
<td>58.5</td>
<td>135.9</td>
</tr>
</tbody>
</table>

The correlation coefficient between number of effective mental discriminations, \( E \), and the reported number of bugs, \( B \), is 0.982, indicating that most of the variation has been explained.

Further, by using the usual figure of 18 mental discriminations per second for fluent, concentrating programmers [6], (Stroud [7] gives the range as 5 to 20 per second), and summing the values of \( E \), one obtains the total effort of the task as \( 903 \times 10^6 \) effective discriminations, or 84 man-months. This figure compares reasonably well with the 100 man-months reported by Akiyama.
REFERENCES:


