

Kerry M. Atchinson
Boeing Military Airplane Company
Wichita, Kansas

ABSTRACT

Generation and support of documentation accompanying software development is historically a low-efficiency, high-cost undertaking. Frequently the situation arises where a choice must be made to fulfill documentation requirements and incur cost and schedule overruns, or to complete the software product in a reasonable time and deliver less than adequate documentation. The primary difficulties are efficient generation, quick and thorough update, and document correlation on large projects. Automated methods can alleviate these problems by (1) application of word processing systems to the generation and editing of descriptive text, (2) use of a data base manager-type control of system interface definition and document correlation, (3) use of pseudo-code for first-time design and flowchart generation, and (4) the use of special purpose software tools to perform analysis of code for flowchart and module interface updating.

Documentation of software can be performed on several levels of complexity. The lowest level is perhaps the basic user's guide that accompanies small software items or those packages where a detailed knowledge of the software internals is unneeded or undesirable. This guide is usually textual, giving only installation and usage instructions. A higher level of complexity is encountered when system descriptions are included, as may be the case with a vendor's operating system. The user in these cases is given needed information on the interplay of the system's elements and possibly a fair amount of detail on the internal workings of the system in addition to the installation and usage information. Still omitted are the detailed design information, engineering trade-offs, and associated background information on philosophy, methodology, etc., that must accompany the most complex level of documentation — that given to the military or the government in general. Examples are compliance with the "Part II Specification" called out by USAF Data Item Descriptions DI-E-3120B/M1 Computer Program Product Specifications, and the DI-H-3277/M7 Training Equipment Computer Program Documentation.

The Part II Specification includes computer program descriptions, table and data base descriptions, a real-time cross reference, programmer's notebooks, time and memory allocation tracking data, a Computer Program System (CPS) guide, and of course, any associated vendor manuals. The general contents of the Part II Specification are depicted in Figure 1. Of the documentation items listed, only two items are easily supplied: the listings and vendor manuals.

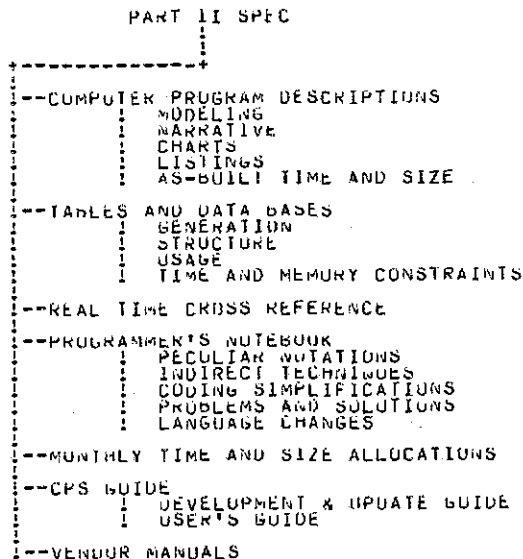


Figure 1. Part II Specification Structure

On a large software project, the effort required to produce this documentation can be staggering. The Boeing Military Airplane Company (BMAC), in building the B-52/KC-135 Weapons Systems

Trainer (WST) prototypes for the USAF, produced approximately 1,000,000 lines of source code. This resulted in roughly 91,000 sheets of non-listing detailed design documentation. This does not include the amount of documentation produced in the generation of functional specifications, implementation specifications, or the programmers' notebooks, CPS guide, and other items of the Part II Spec. Additionally, there were in excess of 7,000 revisions to the source parts comprising the CPS. Assuming that each revision and associated document release required the modification of a minimum of three sheets (one flowchart, one narrative, and one revision status sheet), there were at least 21,000 regenerated documentation sheets for a total of 112,000 effective non-listing pages of design documents.

Naturally, there are impediments to the completion of a task of this magnitude. One of the problems with which many organizations must deal is the lack of sufficient numbers of skilled engineers and programmers to perform the designing and coding tasks. Engineering support personnel who should undertake much of the documentation task are therefore frequently utilized to perform in an engineering role. This places a greater burden of document generation upon the engineers and programmers which adds to the schedule impact.

The major issue in documenting software, even with an abundance of designers and support personnel, is the cost. A reasonable cost estimate for a delivered page of documentation is three manhours (MH).⁽¹⁾ It can be assumed that about one MH for one round of editing and revising with an additional 0.5 MH for typing⁽²⁾ are embedded in this figure. Each additional revision of a page can then be estimated at 1.5 MH. The documentation cost of a WST-sized project could then be expected to be 304,500 MH, or approximately 1,750 man months (MM). This can be further emphasized by noting that a comparison of document generation to code generation shows that documentation accounts directly and indirectly for about 60 percent of project costs as opposed to 40 percent of project costs for coding.⁽³⁾ Obviously, documentation is a candidate for cost reduction.

How can this cost be reduced? Perhaps the most apparent remedy is the application of word processing to those areas of the documents that are textual. In-house BMAC experience with word processing indicates that savings of 60-70 percent are possible when revision and retyping are undertaken through word processing rather than conventional secretarial methods. The 7,000-plus released revisions on WST could be translated to an expected expenditure of 31,500 MH. A 60 percent reduction via word processing brings about a possible savings of 18,900 MH. If the cost of revising a document page is conservatively set at one MH, the savings over the 21,000 revised pages is still 12,600 MH. Applying these possible savings to the embedded revision in the three MH per page yields a modified page cost of 2.4 MH, implying a savings of 20 percent over the basic cost of the 91,000 non-revision sheets, or 54,600 MH (still using one MH per embedded revision). When considered together, the reductions amount to about 67,200 MH or 22 percent. Of course, the major benefit of this type of reduction is not that the job could be done more efficiently. Rather, a less ominous spectre of the documentation cost would allow the job to be performed in the first place.

The word processor would be the mainstay of smaller projects and those supplying only the lower levels of documentation. Indeed, for

the large project generating a Part II Spec, the word processor can play a very large role, particularly in supporting the CPS guide, the programmer's notebooks, and the modeling and narrative sections of the computer program descriptions. Other areas are better served in other manners, especially the module design flowcharts, interface lists, data base structure and memory allocation, and the real-time cross reference. Of these, the most extensive and time-consuming in generation are the flowcharts and the interface lists.

Flowcharting an 'as-built' piece of software is prone to similar problems as writing the narrative text - adherence to drawing standards, typing comments into the flow elements, analysis of the code, etc. An alternative is to implement an automatic flowcharting tool. Without the specific approval of the procuring agency, however, this is proscribed by data item descriptions such as the DI-H-3277/M7. Why is there an aversion to auto-flowcharting? The usual auto-flow tool has been used to generate what amounts to an additional listing of the code which fails to enhance the software's supportability. The secondary goal of an auto-flow tool then should be to chart the design in an understandable and useful form. One of the more highly acclaimed design approaches is the top-down method, where the basic programming problem is iteratively divided into subsets of less general, more detailed problems that can eventually be easily solved on an individual basis. Flowcharting a top-down design in a likewise manner greatly enhances the usefulness of the documentation and its acceptability to the user.

BMAC has an auto-flow tool in the final stages of development, the Document Support System (DSS), which performs the flowchart generation in a top-down manner thus directly reflecting the levels of the design. At the same time, the source code image is separated into design levels corresponding to the generated flowcharts and formatted to reflect the logic nesting of each level. For illustration, a simple two-design level program has been generated and analyzed. Figure 2 shows the entire source image listing of the program which has been written for a Gould SEL 32/55 using a subset of S-FORTRAN developed by Caine, Farber & Gordon, Inc.

```

*      PART NUMBER
PROGRAM X
DATAPOOL LBPPHUGD
LOGICAL * 1 LBPPHUGD

1.0 THIS IS THE TOP LEVEL
? SHALL WE DO IT ?
IF ( L )
1.1 DO THE FIRST THING
I = 1
1.2 DO THE SECOND THING

1.2.1 FIRST PART, SECOND THING
K = 3
? IS INDICATION NECESSARY ?
IF ( I.EQ.2 )
1.2.2 SET THE INDICATOR
LBPPHUGD = .FALSE.
END IF
END IF
END

```

Figure 2. Example Program

Separation of the code and attendant flowcharts per design level is depicted in Figures 3 through 7. The leading block of code, used for variable mnemonic attribute generation, common area references, and data initialization has been set aside in Figure 3. Figures 4 and 5 represent the top design level of 'executable' code. In Figure 4, code subsection 1.1 is shown with a replacement statement highlighted by dashed lines, while subsection 1.2 is represented only by a comment. This makes it immediately apparent that 1.1 is not subdivided into lower design levels, while 1.2 is. The flowchart of Figure 5 shows the logical flow of the top design level. Branching due to TRUE or FALSE states of the decision is indicated by 'T' or 'FF' within the logic paths leading from the decision block. Figures 6 and 7 show the second level of design (code subsection 1.2) pictured with the corresponding comment and process block displayed in the previous design level to aid in identification of the listing and flowchart. Those separated

segments of code and flowchart lend themselves quite readily to integration with descriptive text for each of the design levels.

```

REV :  STATEMENT
-----
- :      *
- :      PART NUMBER
- :      PROGRAM X
- :      DATAPOOL LBPPHUGD
- :      LOGICAL * 1 LBPPHUGD
- :      C

```

Figure 3. Leading Code Block

```

REV :  STATEMENT
-----
- :      C 1.0 THIS IS THE TOP LEVEL
- :      C 2 SHALL WE DO IT ?
- :      C IF ( L )
- :      C 1.1 DO THE FIRST THING
- :      C 1.2 DO THE SECOND THING
- :      C 1.2.1 FIRST PART, SECOND THING
- :      C 1.2.2 SET THE INDICATOR
- :      C 1.2.2.1 LBPPHUGD = .FALSE.
- :      C 1.2.2.2 END IF
- :      C 1.2.2.3 END IF
- :      C 1.2.2.4 END IF
- :      C 1.2.2.5 END IF
- :      C 1.2.2.6 END IF
- :      C 1.2.2.7 END IF
- :      C 1.2.2.8 END IF
- :      C 1.2.2.9 END IF
- :      C 1.2.2.10 END IF
- :      C 1.2.2.11 END IF
- :      C 1.2.2.12 END IF
- :      C 1.2.2.13 END IF
- :      C 1.2.2.14 END IF
- :      C 1.2.2.15 END IF
- :      C 1.2.2.16 END IF
- :      C 1.2.2.17 END IF
- :      C 1.2.2.18 END IF
- :      C 1.2.2.19 END IF
- :      C 1.2.2.20 END IF
- :      C 1.2.2.21 END IF
- :      C 1.2.2.22 END IF
- :      C 1.2.2.23 END IF
- :      C 1.2.2.24 END IF
- :      C 1.2.2.25 END IF
- :      C 1.2.2.26 END IF
- :      C 1.2.2.27 END IF
- :      C 1.2.2.28 END IF
- :      C 1.2.2.29 END IF
- :      C 1.2.2.30 END IF
- :      C 1.2.2.31 END IF
- :      C 1.2.2.32 END IF
- :      C 1.2.2.33 END IF
- :      C 1.2.2.34 END IF
- :      C 1.2.2.35 END IF
- :      C 1.2.2.36 END IF
- :      C 1.2.2.37 END IF
- :      C 1.2.2.38 END IF
- :      C 1.2.2.39 END IF
- :      C 1.2.2.40 END IF
- :      C 1.2.2.41 END IF
- :      C 1.2.2.42 END IF
- :      C 1.2.2.43 END IF
- :      C 1.2.2.44 END IF
- :      C 1.2.2.45 END IF
- :      C 1.2.2.46 END IF
- :      C 1.2.2.47 END IF
- :      C 1.2.2.48 END IF
- :      C 1.2.2.49 END IF
- :      C 1.2.2.50 END IF
- :      C 1.2.2.51 END IF
- :      C 1.2.2.52 END IF
- :      C 1.2.2.53 END IF
- :      C 1.2.2.54 END IF
- :      C 1.2.2.55 END IF
- :      C 1.2.2.56 END IF
- :      C 1.2.2.57 END IF
- :      C 1.2.2.58 END IF
- :      C 1.2.2.59 END IF
- :      C 1.2.2.60 END IF
- :      C 1.2.2.61 END IF
- :      C 1.2.2.62 END IF
- :      C 1.2.2.63 END IF
- :      C 1.2.2.64 END IF
- :      C 1.2.2.65 END IF
- :      C 1.2.2.66 END IF
- :      C 1.2.2.67 END IF
- :      C 1.2.2.68 END IF
- :      C 1.2.2.69 END IF
- :      C 1.2.2.70 END IF
- :      C 1.2.2.71 END IF
- :      C 1.2.2.72 END IF
- :      C 1.2.2.73 END IF
- :      C 1.2.2.74 END IF
- :      C 1.2.2.75 END IF
- :      C 1.2.2.76 END IF
- :      C 1.2.2.77 END IF
- :      C 1.2.2.78 END IF
- :      C 1.2.2.79 END IF
- :      C 1.2.2.80 END IF
- :      C 1.2.2.81 END IF
- :      C 1.2.2.82 END IF
- :      C 1.2.2.83 END IF
- :      C 1.2.2.84 END IF
- :      C 1.2.2.85 END IF
- :      C 1.2.2.86 END IF
- :      C 1.2.2.87 END IF
- :      C 1.2.2.88 END IF
- :      C 1.2.2.89 END IF
- :      C 1.2.2.90 END IF
- :      C 1.2.2.91 END IF
- :      C 1.2.2.92 END IF
- :      C 1.2.2.93 END IF
- :      C 1.2.2.94 END IF
- :      C 1.2.2.95 END IF
- :      C 1.2.2.96 END IF
- :      C 1.2.2.97 END IF
- :      C 1.2.2.98 END IF
- :      C 1.2.2.99 END IF
- :      C 1.2.2.100 END IF
- :      C 1.2.2.101 END IF
- :      C 1.2.2.102 END IF
- :      C 1.2.2.103 END IF
- :      C 1.2.2.104 END IF
- :      C 1.2.2.105 END IF
- :      C 1.2.2.106 END IF
- :      C 1.2.2.107 END IF
- :      C 1.2.2.108 END IF
- :      C 1.2.2.109 END IF
- :      C 1.2.2.110 END IF
- :      C 1.2.2.111 END IF
- :      C 1.2.2.112 END IF
- :      C 1.2.2.113 END IF
- :      C 1.2.2.114 END IF
- :      C 1.2.2.115 END IF
- :      C 1.2.2.116 END IF
- :      C 1.2.2.117 END IF
- :      C 1.2.2.118 END IF
- :      C 1.2.2.119 END IF
- :      C 1.2.2.120 END IF
- :      C 1.2.2.121 END IF
- :      C 1.2.2.122 END IF
- :      C 1.2.2.123 END IF
- :      C 1.2.2.124 END IF
- :      C 1.2.2.125 END IF
- :      C 1.2.2.126 END IF
- :      C 1.2.2.127 END IF
- :      C 1.2.2.128 END IF
- :      C 1.2.2.129 END IF
- :      C 1.2.2.130 END IF
- :      C 1.2.2.131 END IF
- :      C 1.2.2.132 END IF
- :      C 1.2.2.133 END IF
- :      C 1.2.2.134 END IF
- :      C 1.2.2.135 END IF
- :      C 1.2.2.136 END IF
- :      C 1.2.2.137 END IF
- :      C 1.2.2.138 END IF
- :      C 1.2.2.139 END IF
- :      C 1.2.2.140 END IF
- :      C 1.2.2.141 END IF
- :      C 1.2.2.142 END IF
- :      C 1.2.2.143 END IF
- :      C 1.2.2.144 END IF
- :      C 1.2.2.145 END IF
- :      C 1.2.2.146 END IF
- :      C 1.2.2.147 END IF
- :      C 1.2.2.148 END IF
- :      C 1.2.2.149 END IF
- :      C 1.2.2.150 END IF
- :      C 1.2.2.151 END IF
- :      C 1.2.2.152 END IF
- :      C 1.2.2.153 END IF
- :      C 1.2.2.154 END IF
- :      C 1.2.2.155 END IF
- :      C 1.2.2.156 END IF
- :      C 1.2.2.157 END IF
- :      C 1.2.2.158 END IF
- :      C 1.2.2.159 END IF
- :      C 1.2.2.160 END IF
- :      C 1.2.2.161 END IF
- :      C 1.2.2.162 END IF
- :      C 1.2.2.163 END IF
- :      C 1.2.2.164 END IF
- :      C 1.2.2.165 END IF
- :      C 1.2.2.166 END IF
- :      C 1.2.2.167 END IF
- :      C 1.2.2.168 END IF
- :      C 1.2.2.169 END IF
- :      C 1.2.2.170 END IF
- :      C 1.2.2.171 END IF
- :      C 1.2.2.172 END IF
- :      C 1.2.2.173 END IF
- :      C 1.2.2.174 END IF
- :      C 1.2.2.175 END IF
- :      C 1.2.2.176 END IF
- :      C 1.2.2.177 END IF
- :      C 1.2.2.178 END IF
- :      C 1.2.2.179 END IF
- :      C 1.2.2.180 END IF
- :      C 1.2.2.181 END IF
- :      C 1.2.2.182 END IF
- :      C 1.2.2.183 END IF
- :      C 1.2.2.184 END IF
- :      C 1.2.2.185 END IF
- :      C 1.2.2.186 END IF
- :      C 1.2.2.187 END IF
- :      C 1.2.2.188 END IF
- :      C 1.2.2.189 END IF
- :      C 1.2.2.190 END IF
- :      C 1.2.2.191 END IF
- :      C 1.2.2.192 END IF
- :      C 1.2.2.193 END IF
- :      C 1.2.2.194 END IF
- :      C 1.2.2.195 END IF
- :      C 1.2.2.196 END IF
- :      C 1.2.2.197 END IF
- :      C 1.2.2.198 END IF
- :      C 1.2.2.199 END IF
- :      C 1.2.2.200 END IF
- :      C 1.2.2.201 END IF
- :      C 1.2.2.202 END IF
- :      C 1.2.2.203 END IF
- :      C 1.2.2.204 END IF
- :      C 1.2.2.205 END IF
- :      C 1.2.2.206 END IF
- :      C 1.2.2.207 END IF
- :      C 1.2.2.208 END IF
- :      C 1.2.2.209 END IF
- :      C 1.2.2.210 END IF
- :      C 1.2.2.211 END IF
- :      C 1.2.2.212 END IF
- :      C 1.2.2.213 END IF
- :      C 1.2.2.214 END IF
- :      C 1.2.2.215 END IF
- :      C 1.2.2.216 END IF
- :      C 1.2.2.217 END IF
- :      C 1.2.2.218 END IF
- :      C 1.2.2.219 END IF
- :      C 1.2.2.220 END IF
- :      C 1.2.2.221 END IF
- :      C 1.2.2.222 END IF
- :      C 1.2.2.223 END IF
- :      C 1.2.2.224 END IF
- :      C 1.2.2.225 END IF
- :      C 1.2.2.226 END IF
- :      C 1.2.2.227 END IF
- :      C 1.2.2.228 END IF
- :      C 1.2.2.229 END IF
- :      C 1.2.2.230 END IF
- :      C 1.2.2.231 END IF
- :      C 1.2.2.232 END IF
- :      C 1.2.2.233 END IF
- :      C 1.2.2.234 END IF
- :      C 1.2.2.235 END IF
- :      C 1.2.2.236 END IF
- :      C 1.2.2.237 END IF
- :      C 1.2.2.238 END IF
- :      C 1.2.2.239 END IF
- :      C 1.2.2.240 END IF
- :      C 1.2.2.241 END IF
- :      C 1.2.2.242 END IF
- :      C 1.2.2.243 END IF
- :      C 1.2.2.244 END IF
- :      C 1.2.2.245 END IF
- :      C 1.2.2.246 END IF
- :      C 1.2.2.247 END IF
- :      C 1.2.2.248 END IF
- :      C 1.2.2.249 END IF
- :      C 1.2.2.250 END IF
- :      C 1.2.2.251 END IF
- :      C 1.2.2.252 END IF
- :      C 1.2.2.253 END IF
- :      C 1.2.2.254 END IF
- :      C 1.2.2.255 END IF
- :      C 1.2.2.256 END IF
- :      C 1.2.2.257 END IF
- :      C 1.2.2.258 END IF
- :      C 1.2.2.259 END IF
- :      C 1.2.2.260 END IF
- :      C 1.2.2.261 END IF
- :      C 1.2.2.262 END IF
- :      C 1.2.2.263 END IF
- :      C 1.2.2.264 END IF
- :      C 1.2.2.265 END IF
- :      C 1.2.2.266 END IF
- :      C 1.2.2.267 END IF
- :      C 1.2.2.268 END IF
- :      C 1.2.2.269 END IF
- :      C 1.2.2.270 END IF
- :      C 1.2.2.271 END IF
- :      C 1.2.2.272 END IF
- :      C 1.2.2.273 END IF
- :      C 1.2.2.274 END IF
- :      C 1.2.2.275 END IF
- :      C 1.2.2.276 END IF
- :      C 1.2.2.277 END IF
- :      C 1.2.2.278 END IF
- :      C 1.2.2.279 END IF
- :      C 1.2.2.280 END IF
- :      C 1.2.2.281 END IF
- :      C 1.2.2.282 END IF
- :      C 1.2.2.283 END IF
- :      C 1.2.2.284 END IF
- :      C 1.2.2.285 END IF
- :      C 1.2.2.286 END IF
- :      C 1.2.2.287 END IF
- :      C 1.2.2.288 END IF
- :      C 1.2.2.289 END IF
- :      C 1.2.2.290 END IF
- :      C 1.2.2.291 END IF
- :      C 1.2.2.292 END IF
- :      C 1.2.2.293 END IF
- :      C 1.2.2.294 END IF
- :      C 1.2.2.295 END IF
- :      C 1.2.2.296 END IF
- :      C 1.2.2.297 END IF
- :      C 1.2.2.298 END IF
- :      C 1.2.2.299 END IF
- :      C 1.2.2.300 END IF
- :      C 1.2.2.301 END IF
- :      C 1.2.2.302 END IF
- :      C 1.2.2.303 END IF
- :      C 1.2.2.304 END IF
- :      C 1.2.2.305 END IF
- :      C 1.2.2.306 END IF
- :      C 1.2.2.307 END IF
- :      C 1.2.2.308 END IF
- :      C 1.2.2.309 END IF
- :      C 1.2.2.310 END IF
- :      C 1.2.2.311 END IF
- :      C 1.2.2.312 END IF
- :      C 1.2.2.313 END IF
- :      C 1.2.2.314 END IF
- :      C 1.2.2.315 END IF
- :      C 1.2.2.316 END IF
- :      C 1.2.2.317 END IF
- :      C 1.2.2.318 END IF
- :      C 1.2.2.319 END IF
- :      C 1.2.2.320 END IF
- :      C 1.2.2.321 END IF
- :      C 1.2.2.322 END IF
- :      C 1.2.2.323 END IF
- :      C 1.2.2.324 END IF
- :      C 1.2.2.325 END IF
- :      C 1.2.2.326 END IF
- :      C 1.2.2.327 END IF
- :      C 1.2.2.328 END IF
- :      C 1.2.2.329 END IF
- :      C 1.2.2.330 END IF
- :      C 1.2.2.331 END IF
- :      C 1.2.2.332 END IF
- :      C 1.2.2.333 END IF
- :      C 1.2.2.334 END IF
- :      C 1.2.2.335 END IF
- :      C 1.2.2.336 END IF
- :      C 1.2.2.337 END IF
- :      C 1.2.2.338 END IF
- :      C 1.2.2.339 END IF
- :      C 1.2.2.340 END IF
- :      C 1.2.2.341 END IF
- :      C 1.2.2.342 END IF
- :      C 1.2.2.343 END IF
- :      C 1.2.2.344 END IF
- :      C 1.2.2.345 END IF
- :      C 1.2.2.346 END IF
- :      C 1.2.2.347 END IF
- :      C 1.2.2.348 END IF
- :      C 1.2.2.349 END IF
- :      C 1.2.2.350 END IF
- :      C 1.2.2.351 END IF
- :      C 1.2.2.352 END IF
- :      C 1.2.2.353 END IF
- :      C 1.2.2.354 END IF
- :      C 1.2.2.355 END IF
- :      C 1.2.2.356 END IF
- :      C 1.2.2.357 END IF
- :      C 1.2.2.358 END IF
- :      C 1.2.2.359 END IF
- :      C 1.2.2.360 END IF
- :      C 1.2.2.361 END IF
- :      C 1.2.2.362 END IF
- :      C 1.2.2.363 END IF
- :      C 1.2.2.364 END IF
- :      C 1.2.2.365 END IF
- :      C 1.2.2.366 END IF
- :      C 1.2.2.367 END IF
- :      C 1.2.2.368 END IF
- :      C 1.2.2.369 END IF
- :      C 1.2.2.370 END IF
- :      C 1.2.2.371 END IF
- :      C 1.2.2.372 END IF
- :      C 1.2.2.373 END IF
- :      C 1.2.2.374 END IF
- :      C 1.2.2.375 END IF
- :      C 1.2.2.376 END IF
- :      C 1.2.2.377 END IF
- :      C 1.2.2.378 END IF
- :      C 1.2.2.379 END IF
- :      C 1.2.2.380 END IF
- :      C 1.2.2.381 END IF
- :      C 1.2.2.382 END IF
- :      C 1.2.2.383 END IF
- :      C 1.2.2.384 END IF
- :      C 1.2.2.385 END IF
- :      C 1.2.2.386 END IF
- :      C 1.2.2.387 END IF
- :      C 1.2.2.388 END IF
- :      C 1.2.2.389 END IF
- :      C 1.2.2.390 END IF
- :      C 1.2.2.391 END IF
- :      C 1.2.2.392 END IF
- :      C 1.2.2.393 END IF
- :      C 1.2.2.394 END IF
- :      C 1.2.2.395 END IF
- :      C 1.2.2.396 END IF
- :      C 1.2.2.397 END IF
- :      C 1.2.2.398 END IF
- :      C 1.2.2.399 END IF
- :      C 1.2.2.400 END IF
- :      C 1.2.2.401 END IF
- :      C 1.2.2.402 END IF
- :      C 1.2.2.403 END IF
- :      C 1.2.2.404 END IF
- :      C 1.2.2.405 END IF
- :      C 1.2.2.406 END IF
- :      C 1.2.2.407 END IF
- :      C 1.2.2.408 END IF
- :      C 1.2.2.409 END IF
- :      C 1.2.2.410 END IF
- :      C 1.2.2.411 END IF
- :      C 1.2.2.412 END IF
- :      C 1.2.2.413 END IF
- :      C 1.2.2.414 END IF
- :      C 1.2.2.415 END IF
- :      C 1.2.2.416 END IF
- :      C 1.2.2.417 END IF
- :      C 1.2.2.418 END IF
- :      C 1.2.2.419 END IF
- :      C 1.2.2.420 END IF
- :      C 1.2.2.421 END IF
- :      C 1.2.2.422 END IF
- :      C 1.2.2.423 END IF
- :      C 1.2.2.424 END IF
- :      C 1.2.2.425 END IF
- :      C 1.2.2.426 END IF
- :      C 1.2.2.427 END IF
- :      C 1.2.2.428 END IF
- :      C 1.2.2.429 END IF
- :      C 1.2.2.430 END IF
- :      C 1.2.2.431 END IF
- :      C 1.2.2.432 END IF
- :      C 1.2.2.433 END IF
- :      C 1.2.2.434 END IF
- :      C 1.2.2.435 END IF
- :      C 1.2.2.436 END IF
- :      C 1.2.2.437 END IF
- :      C 1.2.2.438 END IF
- :      C 1.2.2.439 END IF
- :      C 1.2.2.440 END IF
- :      C 1.2.2.441 END IF
- :      C 1.2.2.442 END IF
- :      C 1.2.2.443 END IF
- :      C 1.2.2.444 END IF
- :      C 1.2.2.445 END IF
- :      C 1.2.2.446 END IF
- :      C 1.2.2.447 END IF
- :      C 1.2.2.448 END IF
- :      C 1.2.2.449 END IF
- :      C 1.2.2.450 END IF
- :      C 1.2.2.451 END IF
- :      C 1.2.2.452 END IF
- :      C 1.2.2.453 END IF
- :      C 1.2.2.454 END IF
- :      C 1.2.2.455 END IF
- :      C 1.2.2.456 END IF
- :      C 1.2.2.457 END IF
- :      C 1.2.2.458 END IF
- :      C 1.2.2.459 END IF
- :      C 1.2.2.460 END IF
- :      C 1.2.2.461 END IF
- :      C 1.2.2.462 END IF
- :      C 1.2.2.463 END IF
- :      C 1.2.2.464 END IF
- :      C 1.2.2.465 END IF
- :      C 1.2.2.466 END IF
- :      C 1.2.2.467 END IF
- :      C 1.2.2.468 END IF
- :      C 1.2.2.469 END IF
- :      C 1.2.2.470 END IF
- :      C 1.2.2.471 END IF
- :      C 1.2.2.472 END IF
- :      C 1.2.2.473 END IF
- :      C 1.2.2.474 END IF
- :      C 1.2.2.475 END IF
- :      C 1.2.2.476 END IF
- :      C 1.2.2.477 END IF
- :      C 1.2.2.478 END IF
- :      C 1.2.2.479 END IF
- :      C 1.2.2.480 END IF
- :      C 1.2.2.481 END IF
- :      C 1.2.2.482 END IF
- :      C 1.2.2.483 END IF
- :      C 1.2.2.484 END IF
- :      C 1.2.2.485 END IF
- :      C 1.2.2.486 END IF
- :      C 1.2.2.487 END IF
- :      C 1.2.2.488 END IF
- :      C 1.2.2.489 END IF
- :      C 1.2.2.490 END IF
- :      C 1.2.2.491 END IF
- :      C 1.2.2.492 END IF
- :      C 1.2.2.493 END IF
- :      C 1.2.2.494 END IF
- :      C 1.2.2.495 END IF
- :      C 1.2.2.496 END IF
- :      C 1.2.2.497 END IF
- :      C 1.2.2.498 END IF
- :      C 1.2.2.499 END IF
- :      C 1.2.2.500 END IF
- :      C 1.2.2.501 END IF
- :      C 1.2.2.502 END IF
- :      C 1.2.2.503 END IF
- :      C 1.2.2.504 END IF
- :      C 1.2.2.505 END IF
- :      C 1.2.2.506 END IF
- :      C 1.2.2.507 END IF
- :      C 1.2.2.508 END IF
- :      C 1.2.2.509 END IF
- :      C 1.2.2.510 END IF
- :      C 1.2.2.511 END IF
- :      C 1.2.2.512 END IF
- :      C 1.2.2.513 END IF
- :      C 1.2.2.514 END IF
- :      C 1.2.2.515 END IF
- :      C 1.2.2.516 END IF
- :      C 1.2.2.517 END IF
- :      C 1.2.2.518 END IF
- :      C 1.2.2.519 END IF
- :      C 1.2.2.520 END IF
- :      C 1.2.2.521 END IF
- :      C 1.2.2.522 END IF
- :      C 1.2.2.523 END IF
- :      C 1.2.2.524 END IF
- :      C 1.2.2.525 END IF
- :      C 1.2.2.526 END IF
- :      C 1.2.2.527 END IF
- :      C 1.2.2.528 END IF
- :      C 1.2.2.529 END IF
- :      C 1.2.2.530 END IF
- :      C 1.2.2.531 END IF
- :      C 1.2.2.532 END IF
- :      C 1.2.2.533 END IF
- :      C 1.2.2.534 END IF
- :      C 1.2.2.535 END IF
- :      C 1.2.2.536 END IF
- :      C 1.2.2.537 END IF
- :      C 1.2.2.538 END IF
- :      C 1.2.2.539 END IF
- :      C 1.2.2.540 END IF
- :      C 1.2.2.541 END IF
- :      C 1.2.2.542 END IF
- :      C 1.2.2.543 END IF
- :      C 1.2.2.544 END IF
- :      C 1.2.2.545 END IF
- :      C 1.2.2.546 END IF
- :      C 1.2.2.547 END IF
- :      C 1.2.2.548 END IF
- :      C 1.2.2.549 END IF
- :      C 1.2.2.550 END IF
- :      C 1.2.2.551 END IF
- :      C 1.2.2.552 END IF
- :      C 1.2.2.553 END IF
- :      C 1.2.2.554 END IF
- :      C 1.2.2.555 END IF
- :      C 1.2.2.556 END IF
- :      C 1.2.2.557 END IF
- :      C 1.2.2.558 END IF
- :      C 1.2.2.559 END IF
- :      C 1.2.2.560 END IF
- :      C 1.2.2.561 END IF
- :      C 1.2.2.562 END IF
- :      C 1.2.2.563 END IF
- :      C 1.2.2.564 END IF
- :      C 1.2.2.565 END IF
- :      C 1.2.2.566 END IF
- :      C 1.2.2.567 END IF
- :      C 1.2.2.568 END IF
- :      C 1.2.2.569 END IF
- :      C 1.2.2.570 END IF
- :      C 1.2.2.571 END IF
- :      C 1.2.2.572 END IF
- :      C 1.2.2.573 END IF
- :      C 1.2.2.574 END IF
- :      C 1.2.2.575 END IF
- :      C 1.2.2.576 END IF
- :      C 1.2.2.577 END IF
- :      C 1.2.2.578 END IF
- :      C 1.2.2.579 END IF
- :      C 1.2.2.580 END IF
- :      C 1.2.2.581 END IF
- :      C 1.2.2.582 END IF
- :      C 1.2.2.583 END IF
- :      C 1.2.2.584 END IF
- :      C 1.2.2.585 END IF
- :      C 1.2.2.586 END IF
- :      C 1.2.2.587 END IF
- :      C 1.2.2.588 END IF
- :      C 1.2.2.589 END IF
- :      C 1.2.2.590 END IF
- :      C 1.2.2.591 END IF
- :      C 1.2.2.592 END IF
- :      C 1.2.2.593 END IF
- :      C 1.2.2.594 END IF
- :      C 1.2.2.595 END IF
- :      C 1.2.2.596 END IF
- :      C 1.2.2.597 END IF
- :      C 1.2.2.598 END IF
- :      C 1.2.2.599 END IF
- :      C 1.2.2.600 END IF
- :      C 1.2.2.601 END IF
- :      C 1.2.2.602 END IF
- :      C 1.2.2.603 END IF
- :      C 1.2.2.604 END IF
- :      C 1.2.2.605 END IF
- :      C 1.2.2.606 END IF
- :      C 1.2.2.607 END IF
- :      C 1.2.2.608 END IF
- :      C 1.2.2.609 END IF
- :      C 1.2.2.610 END IF
- :      C 1.2.2.611 END IF
- :      C 1.2.2.612 END IF
- :      C 1.2.2.613 END IF
- :      C 1.2.2.614 END IF
- :      C 1.2.2.615 END IF
- :      C 1.2.2.616 END IF
- :      C 1.2.2.617 END IF
- :      C 1.2.2.618 END IF
- :      C 1.2.2.619 END IF
- :      C 1.2.2.620 END IF
- :      C 1.2.2.621 END IF
- :      C 1.2.2.622 END IF
- :      C 1.2.2.623 END IF
- :      C 1.2.2.624 END IF
- :      C 1.2.2.625 END IF
- :      C 1.2.2.626 END IF
- :      C 1.2.2.627 END IF
- :      C 1.2.2.628 END IF
- :      C 1.2.2.629 END IF
- :      C 1.2.2.630 END IF
- :      C 1.2.2.631 END IF
- :      C 1.2.2.632 END IF
- :      C 1.2.2.633 END IF
- :      C 1.2.2.634 END IF
- :      C 1.2.2.635 END IF
- :      C 1.2.2.636 END IF
- :      C 1.2.2.637 END IF
- :      C 1.2.2.638 END IF
- :      C 1.2.2.639 END IF
- :      C 1.2.2.640 END IF
- :      C 1.2.2.641 END IF
- :      C 1.2.2.642 END IF
- :      C 1.2.2.643 END IF
- :      C 1.2.2.644 END IF
- :      C 1.2.2.645 END IF
- :      C 1.2.2.646 END IF
- :      C 1.2.2.647 END IF
- :      C 1.2.2.648 END IF
- :      C 1.2.2.649 END IF
- :      C 1.2.2.650 END IF
- :      C 1.2.2.651 END IF
- :      C 1.2.2.652 END IF
- :      C 1.2.2.653 END IF
- :      C 1.2.2.654 END IF
- :      C 1.2.2.6
```

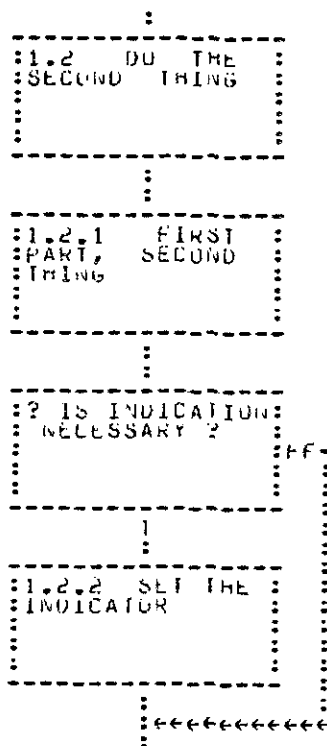


Figure 7. Second Design Level Flowchart

As evidenced, DSS can be used for analysis and documentation of existing code, provided certain coding standards have been followed. Additional applications are the generation of first-time flowcharts and the update of documentation per changes to design. Module designers generate a pseudo code of comments and generic logic structures; then use DSS to construct the design's flowchart. This gives excellent visibility to the design, its evolution, and its adherence to top-down and structured techniques. The example program of Figure 2 is in much the same format as would be generated by a designer under these circumstances. The two variables L and I have been inserted to provide proper syntax for the 'IF' logic structures, and would be replaced with proper mnemonics as they are defined.

An extremely optimistic expectation for the cost of the flowcharts and design level listings would be 1/4 MH per sheet. The amount of time required to generate all the sheets represented by Figures 3 through 7 was about 15 seconds, or 1/4-minute of real-time, leaving a great deal of spare computer time within the 15-second span. If we assume that text and 'boilerplate' comprise about one-half of each design document's non-listing pages, and the other half consists of flowcharts and design-level listings, we can arrive at a cost-savings estimate for combined word processing and auto-flowcharting.

Applying the 22 percent word processing savings to one-half of the original 304,500 MH cost, or 152,250 MH, yields a savings of 33,830 MH. If we then assume the 15-second expenditure for generating the analysis of the example program applies to each sheet rather than all of them, then each sheet costs 15/3600 MH or 1/240 MH. Compared to our conservative 1/4 MH estimate for manual generation, this represents a savings of over 98 percent. The original cost of the 56,000 flowchart and design level sheets would be 14,000 MH at the 1/4 MH rate. A 98 percent reduction would be 13,720 MH for a total savings of 47,550 MH. Applied to a revised original cost of 166,250 MH (allowing 1/4 MH each for the flowcharts and design level lists assumed to comprise one-half of the 112,000 effective pages), the results are a savings of nearly 27 percent. Possibilities of this sort, especially given the conservative estimating, obviously start to bring high quality large documentation tasks into the realm of feasibility.

Documentation of module and system interfaces in a large system can be substantially more difficult than the documentation of the designs, given that many of the designs must be analyzed and correlated to give proper information for a few interfaces. This, of course, should then be accomplished in parallel with the progression of the design as the interfaces evolve. A prime candidate for support of this would be a Data Base Management System (DBMS) which correlates and maintains a data base of system interfaces. For this purpose, BMAC has instituted a DBMS with the MAXIMUM data base manager by California Software Products, Inc., as its nucleus. This DBMS maintains interfaces which are implemented through the Gould SEL Datapool common memory facility and creates the mnemonic dictionaries used to link software load modules to the appropriate data spaces within the Datapool. In addition to the variable mnemonic, recorded data include variable attributes, dates of entry to the system and modification, logical location within the appropriate Datapool, software modules using the variable, and computers where using modules are resident. This obviously eases the pain of document generation as a great deal of this information can be gleaned from software module source code. Others, such as logical location within the Datapool, can be generated by the DBMS in response to variable attributes, unless constrained by operator input. An example of documented interface variables is given in Figure 8. Each of the two variable data entries gives the basic information stated above plus other pertinent facts such as the frequency at which the data is used and generated. Such listings can be easily generated for entire Datapool mnemonic dictionaries, or subsets thereof, to document tables and data bases according to the requirements contained within the Part II Spec.

The individual software module documentation is enhanced by the DBMS-produced module interface listing, shown in part in Figure 9. The module in question is shown boxed in asterisks on the left with individual interface linkages drawn to related modules shown boxed on the right portion of the figure. Each link shows the variable mnemonic for the data space that it represents, the type of data space (i.e., integer byte, logical byte, etc.), the data space array size, and the relative flow of data depicted by an appropriately oriented caret.

Interface documentation on the part of the DBMS is top-down in

KC-135 NAVIGATOR STATION PRIVATE MEMORY CPU 04 (QTP 14 / 03/06/81)									
IBPMFZRQ	MASTER FREEZE REQUEST	-UNITS-	-RANGE-	-ACCURACY-	-RESOLUTION-	-TYPE-	(REF PAGE K0999)	DATE CHANGED	12/31/1980
		N/A	0-255	N/A	+1	1 INTEGER BYTE			
	DATE ENTERED	24 MAY 1978							
GENERATED BY	-MNEMONIC-	-MODULE NAME-	-PART NUMBER-	-FREQ-	-MODE-	-CPU-	-DATE ENTERED-	-DATE CHANGED-	
USED BY	RH1BSEPS	OPSYS/CPU 4 ST EX PC	291-47104-1	F	20 HZ ALL	4	05/25/1978	02/19/1980	
	RH1BSEPS	PROCESS SIMULATOR ST	291-60230-4	A	20 HZ BOTH	4	02/25/1981	12/31/1980	
	RH1BSEPS	OPSYS/CPU 4 ST EX PC	291-47104-1	F	20 HZ ALL	4	05/25/1978	02/19/1980	
	DETAIL DESCRIPTION : (BASE801)								
IBPPOSPT	POINTER TO THE ARRAY POSITION OF	ELAPSED FRAME TIME; VALUE IS RELATED TO CPU	DATE CHANGED	12/31/1980					
	-UNITS-	-RANGE-	-ACCURACY-	-RESOLUTION-	-TYPE-	(REF PAGE 00999)	INITIALIZE TO ZERO		
	TBD	1-7	TBD	TBD	1 INTEGER BYTE				
	DATE ENTERED	26 SEP 1977							
GENERATED BY	-MNEMONIC-	-MODULE NAME-	-PART NUMBER-	-FREQ-	-MODE-	-CPU-	-DATE ENTERED-	-DATE CHANGED-	
USED BY	CPU INIT	CPU 05 INIT DATA	291-47100-9	N/A	N/A	5	12/31/1980	02/24/1980	
	CPU INIT	CPU 04 INIT DATA	291-47100-8	M	N/A	4	12/31/1980	02/24/1980	
	MMOBINIT	INITIALIZE MAINT. EX	291-40726-2	N/A	ALL	5	03/03/1980	04/01/1980	
	MMIBEXEC	MAINTENANCE EXEC	291-40726-1	D	20 HZ ALL	5	11/06/1978	02/27/1981	
	RM1BPEXX	OPSYS/CPU EX NUC PEC	291-40229-1	E	20 HZ ALL	4	05/25/1978	02/24/1980	
	CPU INIT	CPU 05 INIT DATA	291-47100-9	N/A	N/A	5	12/31/1980	02/24/1980	
	SMOLIS	INSTRUCTOR STA CONT	291-47100-8	J	BKGRD ALL	5	02/14/1978	02/24/1980	
	CPU INIT	CPU 04 INIT DATA	291-47100-8	M	N/A	4	12/31/1980	02/24/1980	
	RM1BPE05	OPSYS/CPU EX NUC PEC	RM1BPE05	C	20 HZ BOTH	5	08/26/1977	02/02/1980	
	DETAIL DESCRIPTION : (BASE803)								

Figure 8. Interface Data Base Documentation

```

*****
*
* LNKR      KF  0
* REDIFON
* REVISION:
*
*****

```

Figure 9. Inter-Module Interfaces

As with DSS, the DBMS process has the added benefit that discrepancies can almost always be detected and brought to light as soon as the offending source code is analyzed and its interface expectations are compared to the data base. the DBMS produces an output that documents these discrepancies, as depicted in Figure 12. This particular example indicates that the module RO2BBAOU has defined several mnemonics to be in a common memory area, but never references these mnemonics in executable code. Additionally, the mnemonics are shown not to exist within the data base. This has probably arisen from a previous deletion of the mnemonics from the data base and executable code, while a somewhat careless change implementation has allowed the common definitions to remain. Two other variables are shown to disagree in array sizes between those expected by the module and those implemented by the data base.

While cost estimates for the preparation of interface documentation are not really available, it should be apparent that the process is certainly no easier than manual design documenting. Suffice it to say that the process which generated the information, of which Figure 9 is a part, took less than four minutes of computer real-time. The possibilities of savings should speak for themselves.

In order to efficiently perform these flowcharting and interface documenting capabilities, some standards and conventions should be imposed upon the software to be documented. In the instance of the interface DBMS, variable mnemonics are required to utilize the first three characters to show variable attributes of type and size, and whether they are to reside in a particular region of the Datapool common area or are unique to the module. DSS requires that the code follow structured precepts and be designed in a top-down fashion. Submodules and design levels within the code must then be commented with a numbering system that indicates to which submodule and design level a segment of code belongs. Figure 13 illustrates the hierarchical numbering of the example program processed by DSS. It can be successfully argued that other benefits of such practices (e.g., increased reliability) far outweigh the trouble of standards implementation, without even considering the applications to documentation.

The basic requirements of a general Part II Spec can be satisfied with the word processing, auto-flowcharting, and auto-interface-documenting described above. Other items such as the Real-Time Cross Reference can be completed with an extension of the DBMS source code analysis. Once these or similar mechanisms are implemented, however, there is still room for improvement. Correlation of documentation is a consideration, as are consolidation of document integration and generation of such items as test discrepancy reports, change requests, etc. Document correlation may well take the form of a DBMS function that maintains data on functional requirements documents, implementation requirements documents, and design and test documents. A change to a functional requirement paragraph could then be linked to subordinate implementation requirements and design and test documents, generating a 'need to change' list. Likewise, a module design change could be picked up by the DBMS and result in an output indicating a need to change the appropriate text through word processing.

Figure 10. Example Program Interfaces



```

*** MODULE R02RBAU 291-43172-1 K 06/26/1982 EMPLOYES 110 DATAPOL VARIABLES IN THE FOLLOWING
CPUS = 9
THIS REVISION ( K) HAS PREVIOUSLY BEEN VERIFIED
VARIABLE IRPAOU10 SHOWS UP IN COMMON BLOCK / EXTENDED MEMORY OF PROGRAM BUT IS NEVER USED BY PROGRAM
VARIABLE IRPAOU10 DOES NOT APPEAR IN DATA BASE
VARIABLE IRPAOU20 SHOWS UP IN COMMON BLOCK / EXTENDED MEMORY OF PROGRAM BUT IS NEVER USED BY PROGRAM
VARIABLE IRPAOU20 DOES NOT APPEAR IN DATA BASE
VARIABLE IRPAOUCD SHOWS UP IN COMMON BLOCK / EXTENDED MEMORY OF PROGRAM BUT IS NEVER USED BY PROGRAM
VARIABLE IRPAOUCD DOES NOT APPEAR IN DATA BASE
VARIABLE IRSAOUPR SHOWS UP IN COMMON BLOCK / EXTENDED MEMORY OF PROGRAM BUT IS NEVER USED BY PROGRAM
VARIABLE IRSAOUPR DOES NOT APPEAR IN DATA BASE
VARIABLE IRSATGTI IN STATION HO - ARRAY MISMATCH (MODULE= 1) (DATA BASE= 200)
VARIABLE LBPPCDPA IN STATION RO - ARRAY MISMATCH (MODULE= 3) (DATA BASE= 5)

```

Figure 12. Interface Discrepancies

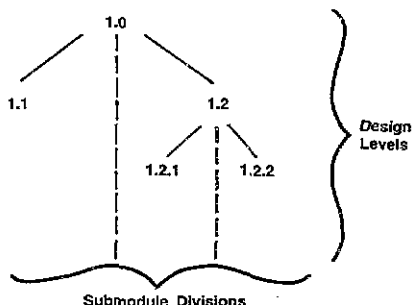


Figure 13. Example Program Design Levels and Submodules

Document integration could entail consolidation of the output of these various processes in order to avoid the manual sorting and shuffling of papers. Use of laser printers, plotters, and the like, if available, is not out of the question. With the obvious savings involved, the automatic processes explored can make giant strides toward high-quality, reasonably priced documentation.

REFERENCES

1. Boehm, B. W., "Software Engineering Economics," Prentice Hall, 1981, p. 574.
2. *ibid*, p. 572.
3. *ibid*, p. 574.

ACKNOWLEDGEMENTS

Special thanks to Stan Chilcott and Jim Richardson of Boeing Military Airplane Company, whose programming expertise was indispensable.

ABOUT THE AUTHOR

Mr. Kerry M. Atchinson received his BSEE from Oklahoma State University in 1977. He has six years experience at Boeing Military Airplane Company where he is currently a Simulation Software Engineer in the Military Training Systems organization. His main responsibilities are the development of organizational standards and guidelines for training systems software, and development of training system supervisory and support software.