An Evaluation of PC-based Modula-2 Compilers

David M. Fellows

June 26, 1990

TR90-049

1. INTRODUCTION

The purpose of this study was to recommend a Modula-2 compiler to be used by students taking our introductory programming courses, CS1013 and CS1035 which are "Introduction to Computer Science" courses, and also for subsequent courses in data structures, etc.

Computing support for these courses will be provided by diskless IBM PCs ("classics") each with 640 k of RAM and connected to a file server running the JANET network system. The network hardware is a 2 mbyte/sec network. Some students will have their own PCs and will want to use them.

In this environment students spend most of their time editing, compiling and diagnosing small (10-1000 line) programs. As soon as they get the programs working they throw them away.

Fast compilation, good error diagnostics and robustness, and convenient editing were the prime factors considered in this evaluation.

2. PRODUCTS CONSIDERED

The products evaluated were Stony Brook QuickMod[1] and TopSpeed[1] Modula 2 (versions 1.14 and 2.0). The shareware product, FST Modula, was given early consideration but was rejected because run-time errors produce only a hex dump of the machine state.

Both products are comprised of an "environment" containing at least an editor, a "make" program, a compiler, a linker, an interactive debugger and a run-time library. Both products have editors with a "Wordstar style" command set. Both editors can be customized to a certain degree. Commands can be assigned to your choice of key stroke sequences. Neither is programmable in the sense that Emacs is.

---

[1] QuickMod and TopSpeed are trademarks.

Both products have an interactive debugger associated with them. The debuggers were "played with" but not seriously evaluated. Both did the sorts of things they could be expected to do.[1]

2.1 Stony Brook QuickMod

This compiler is a product of Stony Brook Software Inc. List price is US $95.00. The QuickMod system was evidently designed for fast compilation speed. It has a companion optimizing compiler which was not tested. The minimum hardware requirements are 512 K RAM and 2-360 K floppy diskdrives. 640 K RAM is required to work with programs with many modules or large arrays. 640 K is also needed to use the debugger from within the environment. 720 K floppy disks are likely to be much more "comfortable" than 360 K disks. QuickMod is usable from 2-360 K floppies or 1-720 K floppy. Individual users will be "happier" with a hard disk. DOS 3.0 is apparently the earliest version of DOS that the system reliably works with.

The QuickMod runtime library includes a fairly close implementation of the library described in Wirth's report. It also includes a number of modules specific to MS-DOS and the IBM-PC. These were not tested, but appear to provide useful functions.

Unfortunately the source for the Definition modules making up the runtime library is not provided[2]. It is not possible to quickly look up the argument list for a particular function.

Documentation is in two volumes. A User's Guide, 128 pp., which describes the use of both QuickMod and its companion optimizing compiler and of course the editor and environment commands. The language and the "Definition Modules" for the runtime library are described in the Reference, 369 pp.

The documentation is adequate but not "great". Some interesting sample programs are included including the editor and a chess-playing program.

---

[1] Debugger performance was played down because I think that students learn more by doing their own diagnostic output.

[2] Source for the library Definition and Implementation modules is provided to educational site licencees.

2.2 TopSpeed Modula-2, Version 1.

This compiler is a product of Jenson & Partners International. List price was US$99.50. It is now available only to the education market. TopSpeed is an optimizing compiler. The design criteria were evidently fast execution and compatibility with other DOS programming tools generally. It has an air of "being all things to all people". Its stated minimum hardware requirements are 2-360K floppy diskettes. This configuration was not tested.

My other tests indicate that a hard disk would be required for satisfactory operation on an individual machine. 640K RAM is required in practise. DOS 3.0 or later is required. The interactive debugger was a US$59.95, list optional extra.

Documentation consists of a User's Manual which describes how to run the system components and describes the library interfaces and a Language Tutorial. The interactive debugger is described in a third document. Documentation is adequate.

The TopSpeed runtime library is quite different from the "standard" one. This produced some problems in running textbook problem "off the shelf". An implementation of Wirth's InOut module (and others) is provided in source form. The source for the Definition modules of all the libraries is provided.

2.3 TopSpeed Modula-2, Version 2

This system is an "enhanced" version of the system discussed above. It incorporates the interactive debugger and lists for US$199 including the debugger. The interface has been redesigned to mesh with other compiler products produced by Jensen & Partners. Most of the revisions are not applicable in our intended use.

The documentation has been revised and consists of a User's Manual and Language Tutorial.

This compiler was received only very recently and was not subjected to the full set of tests.

3. EVALUATION

Evaluation was done in two phases. The first was an informal evaluation of the usability of the system. This was done by entering, debugging and running a total of 4 programs and noting problems, features, etc. The second was

a performance test of the systems installed on a JANET network.

3.1 Usability

The four programs used were taken from Tremblay and DeDourek[1]. Their characteristics are tabulated in Table 3.1.1.

Table 3.1.1
Test Program Characteristics

	Program Number 1	2	3	4
Name:	FactN	ClassStat4	SkiRental	Sequential Update
No.of Modules:	1	1	3	6
No.of Lines:	24	95	500	395
Library Modules Used:	InOut	InOut	ASCII, InOut, Storage, Strings	InOut, System, FileSystem
No. of Source Characters:	700	3400	13900	12600
Comments:	Trivial	Simple Arrays	Complex, pointers, many procedures	Sequential file I/O, Moderate Complexity
Ref. Page:	83	173	411	486

Programs 1 and 3 were liberally seeded with errors: typing, deliberate syntax, undeliberate logic. The utility of the systems in diagnosing them was assessed. TopSpeed Modula-2, Version 2 was not given a full test, but it appears to be essentially the same as Version 1 for our purposes.

---

[1] Tremblay, J.P., DeDourek, J.M., Programming in Modula-2, McGraw-Hill, New York, 1989.

TopSpeed generally has very good syntax error messages. QuickMod has even better ones. Both systems place the user back in the editor with the cursor positioned at the first error at the end of a compile which fails. Subsequent errors are located by pressing the PF8 key. TopSpeed allows you to skip backwards as well as forwards, QuickMod does not.

Both systems use "Wordstar-like" editors. I don't particularly like either, but both are adequate for the job. The TopSpeed editor has more flexibility with respect to windows - up to 4, arbitrarily sized and placed. QuickMod will retain as many files as will fit in RAM. The windows can be switched to display any file in memory very quickly, in a manner similar to the WATCOM editor.

The TopSpeed system logs screen I/O so that it can be reviewed from the editor while errors are being corrected. This is an advantage over the Stony Brook system. Students can use the DOS screen capture facility for this purpose.

Both compilers have options for generating code to check for subscripts out of range, use of nil pointer, etc. These are specified using menu selection and, for TopSpeed, by using "magic" comments. TopSpeed required a magic comment in each module to turn on pointer checking. This was disaster prone.

My overall impression, gained while debugging the test programs, was that QuickMod provided a more robust working environment. I crashed the machine several times, requiring a power-off re-boot, while using TopSpeed; never while using QuickMod. I attribute this in part to the better pointer checking in QuickMod.

My overall impression was that the TopSpeed menu and editing interface is somewhat better than the QuickMod interface. The difference is not large, there are good and bad points to each.

3.2 Performance

The primary benchmark used for performance evaluation was the following sequence of actions applied to a single module program (program 2 in Table 3.1.1).

1. Load system and program (Timed)
2. Modify the program with a null-effect modification (Not timed)
3. Run the program ("make", compile, link and execute) (Timed to start of execution)
4. Run the program again (Timed to start of execution).
5. Edit the program to induce errors. ("VAR" changed to "VAOR") (Not timed)

6. Run the resulting program. (Timed to placement of cursor in file for error correction).
7. Exit system (Timed)

This benchmark was run on these configurations:

1. A true IBM PC-XT
2. A diskless IBM-PC (8088 processor) served by the JANET E-cluster file server.
3. An IBM PS-2 Model 25 (8086 processor) served by the JANET E-cluster file server.

For the tests of configurations 2 and 3 the network was otherwise idle. The file server is an AT-class machine with 90 megabyte disk drives.

Each compiler was setup to include the most protection, diagnostic, and safety features. Optimization was inhibited for the TopSpeed compilers.

The results of this test are summarized in Table 3.2.1

Table 3.2.1

Compilation Benchmarks

	Load System	Run	Run Again	Compile time error	Exit System
PC-XT					
QuickMod	10	6.3	1.0	5.6	NM
TopSpeed V1.14	9.4	31	1.2	14	NM
8088/JANET					
QuickMod	18	19	1.0	6.6	13.5
TopSpeed V1.14	18	82	2.5	8	8
TopSpeed V2	41	200	10	NM	14
8086/JANET					
QuickMod	14.5	15	0.5	5.5	11
TopSpeed V1.14	13.5	73	4.0	46	6
TopSpeed V2	28	139	9	28	8

Notes: 1. All times in seconds
2. NM - Not Measured

Step 3 of the benchmark was also executed simultaneously on 3 8086 processor workstations to observe the effect of

network and server load. The times recorded are for the latest starting workstation.

Table 3.2.2

Loaded JANET System Benchmark Times

	One Workstation	Three Workstations
QuickMod	14.5	42
TopSpeed 1.14	73	135
TopSpeed 2.0	139	NM

Notes: 1. Times in seconds
2. 8086 processors
3. NM - Not Measured

Three simultaneous compiles were chosen to be representative of the network load presented by 25 active workstations.

A secondary benchmark was run to determine program execution speeds. This consisted of timing the execution of a selection sort procedure sorting from 100 to 500 random long integers. The measured portion of this benchmark does no I/O.

The results are summarized in table 3.2.3. Some measurements of WATFOR77 executing the same algorithm coded in Fortran are also shown for comparison. In each case full diagnostics, no optimization was specified.

Table 3.2.3

Execution Speed Benchmark Times (seconds)

Number of elements sorted	Top Speed V1.14	QuickMod	WATFOR77
100	0.7	0.9	2.5
200	1.8	2.4	8.5
250	2.9	3.8	NM
400	7.1	9.2	NM
500	10.8	14.2	NM

The TopSpeed compiler is much slower at compiling than the QuickMod compiler. Most of the difference can be attributed to disk activity. This disk activity will put a severe strain on the JANET network and file server. Table 3.2.2 illustrates the kind of performance the students can expect. While the TopSpeed compiler has less of a

percentage degradation as the load is increased, its absolute numbers are far worse. Furthermore, Version 2 of TopSpeed does even more disk I/O and is even slower than Version 1 (see Table 3.2.1).

As Table 3.2.3 shows, the compiled code from TopSpeed is about 30% faster than QuickMod. In our application this difference is not nearly as significant as the compile times.

Both of these systems will require more disk space per student on JANET than similar size WATFOR77 programs would require. For example QuickMod used 61K bytes of disk storage for program 2 (Table 3.1.1), TopSpeed used 53K for the same program.

Fortunately most of this space is one-time overhead. As programs grow larger the growth is equal to twice the size of the source text files. Twice because the systems keep a backup version of each source file.

Unfortunately, in the case of QuickMod the "recommended" approach is to setup each assignment as an independent module (i.e. 61K bytes per assignment). Obviously students will have to do considerable housekeeping to live within the available disk space.

3.3 Summary

Table 3.3.1 attempts to quantify and summarize my impressions of the two systems. Version 2 of Top Speed is not considered because it is noticeably poorer in performance than its cheaper predecessor. The scale is 0-10, high numbers are better.

Table 3.3.1

Summary Comparison

	Top Speed V1.14	QuickMod
editor	7	7
menus, selection sequences	8	6
libraries	6	9
error messages	8	10
robustness	5	8
debugger	8	7
"making" speed	3	8
hardware resources required	5	7
per student disk space	7	5
Total (90 max)	57	67

4.0 BUSINESS CONSIDERATIONS

TopSpeed Modula-2 pricing is as follows (US$):
V1.17 (incl. debugger) $60 direct to students, incl. shipping
Site license: $500 first 10
$30 each additional

100 stations $3200.00

V2.0 $130 direct to students including shipping

Site license: $995 first 10
$50 each additional

100 stations $5495.00

Stony Brook QuickMod - Pricing (US$):

$40 each on orders of 20 or more for students

Site license: $ 250 up to 10
$1000 up to 50
$2500 unlimited (on university's machines)
(10 copies of documentation)

100 stations $2500.00

5.0 RECOMMENDATIONS

UNB should obtain an unlimited site license for Stony Brook QuickMod. The Faculty should obtain an evaluation copy or two of the Stony Brook Professional Modula 2 compiler. Arrangements should be made for selling copies of QuickMod to students.

Unless the site license is unexpectably generous, the Faculty of Computer Science (possibly jointly with EE) should purchase a "20 pack" of individual sets to provide faculty with documentation and to lend to faculty for use on personal computers off-campus. (In quantity 20, the set is $40, documentation only is $25).

An "Introduction to QuickMod" handout be prepared for September for use by students.

JANET planning should consider 128K bytes per student as a minimum working disk size.

APPENDIX

Supplier Addresses

QuickMod:

Stony Brook Software
187 E. Wilbur Rd., Suite 9
Thousand Oaks, CA 91360
U.S.A.
(805) 496-5837

TopSpeed:

Jenson & Partners International
1101 San Antonio Road, Ste. 301
Mountain View, CA 94043
U.S.A.
USA 1-800-543-5205
Canada 1-800-543-8452
Europe (01) 253-4333