Cost Evaluation of a DSN High Level Real-Time Language

M. McKenzie
TDA Planning

The hypothesis that the implementation of a DSN High Level Real-Time Language will reduce real-time software expenditures is explored. The High Level Real-Time Language is found to be both affordable and cost-effective.

I. Introduction

The Deep Space Network (DSN) spends approximately \$3 million a year on real-time software. It has been hypothesized that the implementation of a DSN High Level Real-Time Language (HLRTL) would reduce this expenditure substantially by increasing programmer efficiency and program maintainability. To explore this hypothesis, the HLRTL costs and DSN real-time software expenditures were evaluated; a Cost/Savings model was applied to the expenditures. The results confirmed the hypothesis – over ten years the DSN will save from \$6 million to \$16 million above costs with the implementation of a High Level Real-Time Language.

II. Cost

To calculate the cost of the HLRTL, the following assumptions were made:

(1) There will be a single HLRTL machine-independent design, with all necessary manuals.

(2) The HLRTL machine-independent design will be implemented on five machines. The implementations will occur during years 1, 2, 3, 6, and 9, each requiring a one-year period.

(3) The first implementation will affect 50% of DSN Real-time Software. With the second implementation, 80% will be affected; with the third, 100%. The remaining implementations are replacements of machines on which the HLRTL has already been implemented.

(4) The time period of the HLRTL, the life-cycle assumed by this study, is ten years. This was found to be more applicable than the usual implementation-plus-ten-year-maintenance cycle, due to the implementation of the HLRTL on different machines at different times.

(5) The HLRTL will be of sufficiently high quality to ensure no increase in hardware resource requirements. Although high level languages generate programs with less than optimal computer efficiency, with an optimizing compiler the machine code generated can still be at least as efficient as that produced by a low level language (for most programmers).

With these assumptions, and component costs as listed in Appendix A-1, the projected life-cycle cost of the DSN High Level Real-Time Language is \$2620K in constant dollars (Appendix A-2).

III. Affordability

From an economic viewpoint, the DSN can afford the HLRTL if its implementation will save at least enough money to cover its costs. It will be shown that the DSN High Level Real-Time Language is highly affordable, the breakeven point probably occurring within 3 years or less of its implementation.

IV. Expenditures on Real-Time Software

Based on a search through the 77-1 Work Authorization Document (WAD) and the Format A job description of each DSN work unit, it was estimated that the DSN will spend $14,880K on real-time software over the period 1977-1982 (Appendix B-1). An independent estimate validated this approach (Appendix B-2).

The DSN ten-year real-time software cost is assumed to be twice this five-year expenditure. This is a low estimate, if anything, since Fiscal Year 77 is at the end of the large Mark III Data System implementation and thus does not include major funding of the previous two years. Hence, the ten-year DSN real-time software expenditures are conservatively estimated at $29,760K (constant dollars). With reference to assumption 3, implementation of the HLRTL will affect $27,677K of the estimated expenditures.

V. Real-Time Software Cost-Savings Model

As seen in the Cost-Savings Model, the effect of a High Level Real-Time Language on DSN real-time software production will vary across size and phase of projects. For instance, the larger the project, the more likely that a HLRTL will save effort in the testing phase. So using industrial and DSN data, the applicable DSN ten-year real-time software expenditures are first distributed by size and phase (Appendix C-1).

Within each phase of a project, one expects an increase in cost-effectiveness with the High Level Real-Time Language. Specifically, the HLRTL will allow savings in each phase as shown in Table 1.

The increase in efficiency, by phase and size, is difficult to ascertain. However, estimates were derived from data found in the software industry and literature (Appendix C-2). The distributed real-time software costs were multiplied by these efficiency factors (measures of increased efficiency under the HLRTL system). The result of the multiplication is the cost per element when using a High Level Real-Time Language. Summing these elemental costs gives the total real-time software cost when using the HLRTL.

VI. Savings

DSN savings are equal to the ten-year DSN real-time software costs using a low level language minus those using the HLRTL. The savings were calculated assuming the efficiency data derived from software industry and literature. The DSN real-time software savings stream, for ten years, was also calculated using this assumption (Appendix D-1). By comparing these yearly savings to the HLRTL cost stream, it was calculated that the HLRTL is affordable and would save $14,196K over costs (constant dollars). The breakeven point, at the beginning of year three, is shown in Appendix D-2.

The efficiency factors are derived from industrial and literature data representing a variety of projects, languages, methodologies, and years, so they cannot be taken as absolute predictors of DSN software costs using a High Level Real-Time Language. In recent work for NASA and the U.S. Air Force, E. N. Dodson of General Research Corporation reports indications that real-time "assembly language programs are approximately two to five times more expensive to develop *per object instruction*," and two to five times more expensive to maintain, than real-time high level programs. Therefore, to check sensitivity of the model, savings were recalculated assuming this range for the efficiency factors (Appendix C-2). The HLRTL is still affordable and saves from $5,921K to $16,013K over costs (in constant dollars).

VII. Conclusion

The DSN High Level Real-Time Language is clearly affordable – savings cover costs. Furthermore, additional savings can be expected to run from $6 million to $16 million over a ten-year period. In fact, in view of the reinforcement produced by the software standard practices employed by the DSN, it is probable that savings will be on the high side of this range near the $14 million value. The proposed High Level Real-Time Language is affordable and cost-effective.

Table 1. Source of savings by software phase

Phase	HLRTL Attribute	Result
Design	Emphasis on top-down design. Fits well with modular development.	Fewer design errors. More efficient designing.
Coding	Flexible language designed for the programmer.	Fewer coding errors. Faster coding.
Testing	High readability of code. Allows modular testing. Allows top-down testing.	Fewer errors. Easier correction of errors. Higher management visibility.
Transfer to Ops	Lends itself to clear documentation.	More efficient testing. More efficient transfer to Ops.
Sustaining	Enhanced readability of code. Documentation is tied to code.	Fewer errors. Errors easier to correct. More efficient sustaining.

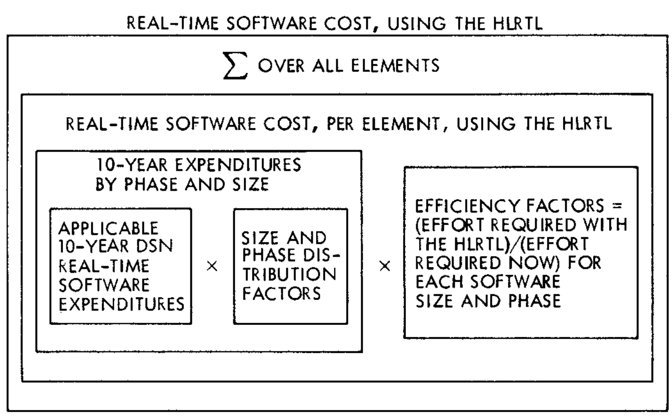

Fig. 1. Cost-savings model

Appendix A-1
HLRTL Life-Cycle Component Costs

Implementation Costs[a]	
One-time machine-independent design (5 MY at $60K/MY)	$ 300K
User Manuals (4 contractor MY at $30K/MY)	$ 120K
Five HLRTL implementations (4.5 MY each at $60K/MY)	$1350K
Sustaining Costs[b]	
Machine-independent design and manuals ($30K/year for 9 years)	$ 270K
HLRTL on five machines ($20K/year per machine)	$ 580K

[a]Derived from information supplied by TDA Engineering using a Work Authorization Document baseline (4/77) and the JPL System for Resource Management (12/76).

[b]This is a high estimate, if anything, based on the sustaining costs of the DSN High Level Non-Real Time MBASIC[tm] Language.

Appendix A-2

HLRTL Cost Stream

Year	1	2	3	4	5	6	7	8	9	10
Implementation ($K)										
Machine-independent design	300									
Five implementations	270	270	270			270			270	
User manuals	120									
Sustaining ($K)										
Machine-independent design and manuals		30	30	30	30	30	30	30	30	30
HLRTL – Machine 1		20	20	20	20	20	20	20	20	20
HLRTL – Machine 2			20	20	20	20	20	20	20	20
HLRTL – Machine 3				20	20	20	20	20	20	20
HLRTL – Machine 4							20	20	20	20
HLRTL – Machine 5										20
Yearly Totals	690	320	340	90	90	360	110	110	380	130
Total for 10 years : $2620K (constant dollars)										

Appendix B-1

DSN Real-Time Software Expenditures (in $K) Derived From The 77-1 Work Authorization Document and Format A's

Work Unit 311-03	1977 Impl[a]	1977 Sus[b]	1978 Impl	1978 Sus	1979 Impl	1979 Sus	1980 Impl	1980 Sus	1981 Impl	1981 Sus
13-24-01	110	110	117	117	122	122	128	128	133	133
13-24-02	90	90	77	77	82	82	87	87	92	92
14-12-01		81		154		164		172		181
14-12-02		161		131		139		146		155
14-12-92		34		32		32		33		33
14-20-02	58	58	67	67	71	71	75	75	80	80
23-10-03	44		47				23		24	
41-01-01	127		98							
41-01-04			27		52		82		35	
41-01-16					28		21			
41-05-10					16					
41-01-09			12		52		55		29	
42-49-07	6									
42-53-04	153		59							
42-53-05	72		24							
42-54-06	115		43							
42-54-22			60		85		15			
42-60-01	151		97							
42-60-02	106		15							
42-60-03	13		17							
42-60-11	214		248							
42-60-16	23		24							
42-60-18	23									
42-61-13	58									
43-10-04	192		222		234		248		262	
43-10-06	66									
43-20-01	60		67		74		80		80	
43-20-04	57		41		42		45		49	
44-12-01		41		62		96		144		119
44-12-04		82		90		83		51		88
44-12-05		60		80		80		89		94
44-12-06		81		48		50		56		62
44-12-07		78		86		51		54		58
44-21-02	62		24		26		27		29	
44-21-03	68		104		108		60		64	
44-23-01	104		76		82		88		94	
44-30-01	350		346		364		429		476	
311-06:										
20-00-72		93		97		150		234		296
30-00-62	36		44		57					
40-00-59	87		65		57					
40-00-60	96		66							
Subtotals	2541	969	2087	1041	1552	1120	1463	1269	1447	1391
Yearly totals	$3510K		$3128K		$2672K		$2732K		$2838K	

Category totals:		
	Implementation costs	$9090K (61%)
	Sustaining costs	$5790K (39%)
	5-year costs	$14880K (constant dollars)

[a]Implementation Cost
[b]Sustaining Cost

Appendix B-2
Accuracy Evaluation

To check the accuracy of the estimate shown in Appendix B-1, a second method was used to estimate FY77 expenditures on real-time software. It was hypothesized that the DSN Data Systems Section comprises half of DSN real-time software budget. This section invested $1.8 million in real-time software in FY77. Thus, the two methods show DSN FY77 real-time software expenditures as:

WAD estimate	$3510K
Independent Estimate	$3600K

This high degree of agreement only means that the two independent estimates derived similar results. Yet the second estimate does lend credence to the five-year estimate of Appendix B-1.

Appendix C-1

Cost Distribution of the DSN Ten-Year Real-Time Software Expenditures[a]

Distribution Element	Distribution Factor, %	Source of Factor
Ten-year costs	100	
Implementation phase	60	1
Large software projects	76	1
Design phase	55	2
Code phase	11	2
Test phase	22	2
Transfer to Ops phase	12	2
Small software projects	24	1
Design phase	55	2
Code phase	11	2
Test phase	22	2
Transfer to Ops phase	12	2
Sustaining phase	40	1
Large software projects	76	1
Small software projects	24	1

[a]It is assumed that over ten years, the distribution of the DSN real-time expenditures will approach that of a software project.

Sources:

1. *Work Authorization Document,* 77-1, Deep Space Network, and corresponding Format A's.
2. Edmund B. Daly, "Management of Software Development," *IEEE Transactions on Software Engineering,* May 1977, p. 232, and Thomas J. Devenny, "An Exploratory Study of Software Cost Estimating at the Electronic Systems Division," NASA Report #N77-18791, Air Force Institute of Technology, p. 29.

Appendix C-2

Efficiency Factors Measured in (Effort Required with the HLRTL)/(Effort Required Now) as Derived from Software Literature and Industry Data

Element	Efficiency Factor	Maximum Value	Minimum Value	Source of Factor
Ten-year costs				
Implementation phase				
Large software projects				
Design phase	0.31	1.00	0.30	1
Code phase	0.20	0.30	0.20	2
Test phase	0.39	0.50	0.20	3
Transfer to Ops phase	0.39	0.50	0.20	4
Small software projects				
Design phase	0.46	1.00	0.45	5
Code phase	0.25	0.45	0.25	6
Test phase	0.54	0.50	0.20	7
Transfer to Ops phase	0.54	0.50	0.20	8
Sustaining Phase				
Large Software Projects	0.27	0.50	0.20	9
Small Software Projects	0.23	0.50	0.20	10

Sources:

1. B. C. Nichols, "SAFEGUARD Data-Processing System: Structured Programming and Program Production Librarians," in L. A. Howard et al. (eds.), *The Bell System Technical Journal Special Supplement,* American Telephone and Telegraph Company, U.S.A., 1975, p. S215.

   and

   F. Terry Baker, "Structured Programming in a Production Programming Environment," *IEEE Transactions on Software Engineering,* Vol. SE-1, No. 2, June 1975, p. 251.

   The increase in efficiency for the design phase is due to the use of structured programming techniques, more easily applied in a high level language environment. These techniques are already included in the DSN software methodology, however, so the maximum value of the efficiency factor was chosen as 1.0, corresponding to no increase in efficiency.

2. B. C. Nichols, *loc. cit.*

   and

   F. Terry Baker, *loc. cit.*

   These sources determined the maximum value of the efficiency factor. To determine the actual and minimum value, it is assumed that a source statement can be coded as quickly in assembly language as a high level language, but that one high level statement is equivalent to five assembly statements. These assumptions were strongly supported at the Goddard Space Flight Center Software Conference, September 19, 1977.

3. B. C. Nichols, *loc. cit.*

   and

   Montgomery Phister, Jr., *Data Processing Technology and Economics,* Santa Monica Publishing Co., Santa Monica, California, 1974, p. 217.

   and

   E. N. Dodson, *Resource Analysis for Data-Processing Software,* General Research Corporation RM-2117, Santa Barbara, California, August 1977, p. 19.

4. B. C. Nichols, *loc. cit.*

   and

   Montgomery Phister, Jr., *loc. cit.*

   and

   E. N. Dodson, *loc. cit.*

5. B. C. Nichols, *loc. cit.*

   Again, the maximum value was assumed to be 1.0.

6. B. C. Nichols, *loc. cit.*

   The minimum and actual values were again chosen with reference to the Goddard Space Flight Center Software Conference information. However, since small programs tend to be less complicated than larger ones, it was assumed here that one high level statement is equivalent to four assembly language statements.

7. B. C. Nichols, *loc. cit.*

   and

   Montgomery Phister, Jr., *loc. cit.*

   and

   E. N. Dodson, *loc. cit.*

8. B. C. Nichols, *loc. cit.*

   and

   Montgomery Phister, Jr., *loc. cit.*

   and

   E. N. Dodson, *loc. cit.*

9 E. N. Dodson, pp. 17 and 19.

10. E. N. Dodson, pp. 17 and 19.

Appendix D-1

Cost and Savings Streams (in $K) Assuming the Efficiency Factors Listed in Appendix C-2

Year	1	2	3	4	5	6	7	8	9	10
HLRTL cost stream	690	320	340	90	90	360	110	110	380	130
Savings, assuming the efficiency factors listed in Appendix C-2	0	1013	1621	2026	2026	2026	2026	2026	2026	2026
Affordability stream = HLRTL cost – savings (in constant dollars)	-690	+693	+1281	+1936	+1936	+1666	+1916	+1916	+1646	+1896

Ten-year Totals: HLRTL $ 2620K (constant dollars)
Savings $16816K (constant dollars)
Affordability $14196K (constant dollars)

Appendix D-2

Cumulative Savings (Above Costs)