ENGINEERING COMPUTER SYSTEMS FOR SIMULATORS

CHARLES F. SUMMER
Naval Training Equipment Center

INTRODUCTION

Computer system engineering in the context of this paper includes both hardware and software. Real-time systems controlled by computers are inherently different from conventional scientific and business ADP environments. There are special system problems and special considerations of timing and program organization to achieve effective real-time processing and control. This paper addresses problems and procedures for engineering computer systems for real-time trainers and simulators.

In analyzing previous computer systems engineering efforts by trainer contractors, a number of problem areas have been identified. Frequently, the real-time simulation problem was molded to a previously selected computer. It was decided to use a specific computer according to some rationale and the contractor would "fit" the simulation problem to the computer instead of initially sizing the computer to the problem. This one item is a key consideration for achieving a good real-time system for trainers. Invariably, under such a philosophy a machine is selected which is either oversized, thereby wasting the extra capacity, or the problem must be significantly modified such that the fidelity of simulation suffers.

The real-time software that should complement the hardware organization frequently has been very inflexible and difficult to modify. It exhibits attributes of an activity that gave little forethought as to how the real-time problem was to be solved. There are a number of instances where it appears that individuals just started programming and coding without any systems work being performed beforehand. For a large sophisticated trainer a good software design can only result from a good systems design.

Inadequate or poor software documentation has been an outstanding problem area. Such documentation has made it difficult and costly to perform even routine software configuration management. Too frequently the trainer has been noncompliant, particularly the computer system, at the time of acceptance testing. In the interest and need for early training, specification waivers had to be granted.

Investigations have been conducted over the last several years to identify these problems and to seek proper solution in a cost-effective manner. Obtaining cost-effective trainers is becoming even more significant with the emphasis on simulation, and the constraints being placed on the Department of Defense because of reduced fiscal resources provided by the Congress. Even though trainers can simulate operational systems and thus reduce the use of operational equipment for training, if trainers are not cost effective, they'll not achieve the intended purpose. The procedures discussed in this paper offer solutions to the spectrum of problems outlined above.

COMPUTER SYSTEM PERFORMANCE REQUIREMENTS

The key to performance requirements is the NAVTRAEQUIPCEN technical specification. The specification provides the minimum complexity desired of the trainer to meet the training requirements. This includes both hardware and software. The specification also delineates any performance and software/hardware overkill desired by NAVTRAEQUIPCEN for future or unforeseen contingencies. However, it should be obvious that the simplest system architecture which meets the performance requirements is usually the most mature design within the state of the art. The design should provide a system which is easy to maintain and is least complex. These are all relative factors that must be weighed during system design and analysis including hardware and software. Software which has been organized on a system basis is usually the most reliable and easiest to maintain. It is easiest to document and to understand.

The complexity of the average trainer or simulator has been increasing. As users realize more of what simulation can provide in the way of adequate training, they are demanding that trainers provide more simulation. Since the computer is the key simulation element in the system in conjunction with the software, as the complexity of the functional requirements increase, the complexity of the software obviously must increase. This in turn gives rise to increased costs. Earlier, when trainers were relatively simple there was a reasonable balance between computer hardware and software. Hardware was the pacing cost item. The software was relatively simple and straightforward in its organization. Within the past three years the cost of computer hardware was decreased exponentially. The performance characteristics of the newer trainers have increased dramatically. The cost of the software has risen not necessarily exponentially, but at least with a very steep slope. Available statistics show that the cost of the computer software can be three to five times the cost of the computer hardware. Approximately 40 to 50 percent of the software cost is incurred during the hardware-software integration phase of trainer checkout. A significant objective is to optimize the software and hardware and reduce the cost of trainers through a system engineering approach to its design. This is to include not only the initial software costs, but also the cost of software configuration management over the life of the trainer. Procedures will be discussed which have significant potential to overcome the problems cited and achieve a cost-effective design.

SYSTEMS ENGINEERING AND DESIGN PROCEDURES

Previously, it was stated that a majority of the more modern trainers have computer equipment which is used to duplicate or simulate either a real or hypothetical system. The computer can simulate system behavior only in the sense that it can manipulate mathematical quantities representing system responses when it is presented with other mathematical quantities representing system stimuli. To this is added, if required, such things as visual, aural and tactile effects and other equipments to enable the computer to effect a transformation between numerical data and receptors and displays to provide a transfer of training. Real-time simulation or the computing of system responses must be accomplished as rapidly as the actual system itself responds. Aircraft flight trainers are examples of the more sophisticated simulators. They must realistically reproduce the performance of the aircraft in response to a trainee who is manipulating the controls. However, the techniques and procedures to be discussed are applicable to simulators and trainers of all complexities, and can be applied to any level of function definition.

Some people tend to view with indifference the introduction of the classical systems engineering procedure[1][4] which has been fostered for the last decade. Others have considered it as a tool that can enable them to do a better design job. The systems engineering approach, regardless of the level it is initiated, is one of the best methods of achieving an adequate design of a complex system. If this is coupled with other design requirements, a more cost-effective design can be achieved.

Good computer system design for simulators and trainers should involve three basic procedures, as indicated in Figure 1. Procedure I encompasses System Engineering Procedures which include analyses and system design. Procedure II involves the Programming and Coding which includes program debugging, assembly and verification independent of the trainer per se. Procedure III covers hardware-software integration with the full trainer hardware. The final steps that are not specifically related to the computer system design are the formal tests and acceptance, the documentation requirements, and delivery, installation and acceptance for training at a designated site.

The input to Procedure I is a contract document, usually the technical specifications. Between the systems engineering activity as shown here and Procedure II, Programming and Coding, there are design reviews and the approval of design documents required by SECNAVINST 3560.1 [2][3]. In parallel with this is the remainder of the design of the simulator system, the fabrication of components and their assembly. Between Procedure step II and III, as far as computer system design is concerned, there will be additional design reviews so that cognizance can be maintained of the programming effort and the problems that may arise. The final procedure involving the computer system design is the hardware-software integration effort of Step III. At that time the program will be verified by the full trainer performance.

The System Engineering Procedures of Block I [4] are further subdivided as shown on Table I. The function analysis is achieved by way of function flow block diagrams. Sequential system functions are identified and become a functional model to satisfy the total simulation requirements set forth in the specifications. A basis is developed for establishing the function interfaces as well as identifying system and other relationships of the trainer. Function analysis also involves function analysis sheet preparation. This is the initial assignment of functions to be accomplished by computer programs and those to be accomplished by special purpose hardware. Analysis Reports are internal documents the contractor should accumulate and maintain for his own use in his system engineering process.

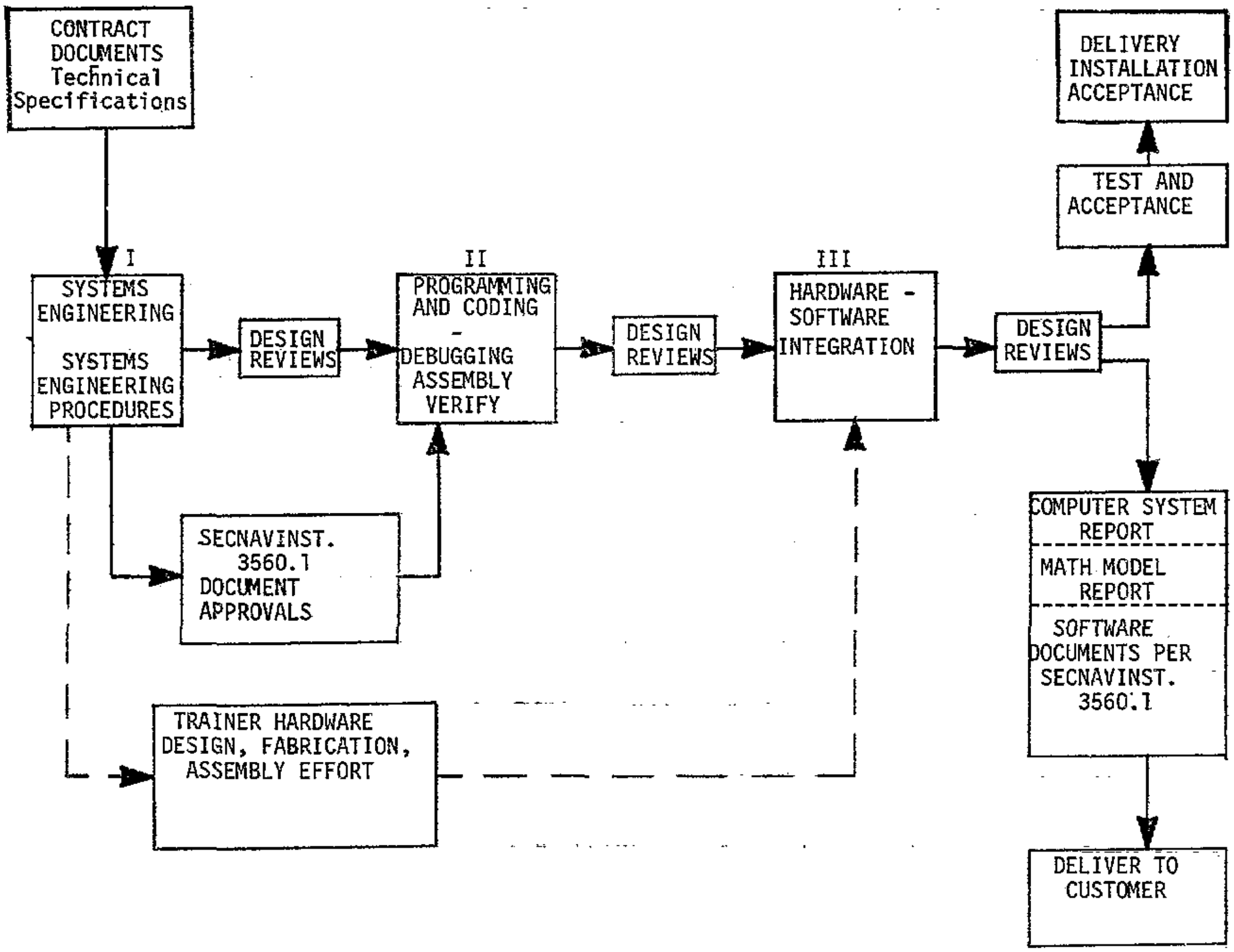

Figure 1. Computer System Design Steps

They document the results of engineering analysis required in the development of the function model and system synthesis. These analyses may include such things as system performance, weight, volume, hydraulic, pneumatic, electrical, heating and cooling loads, stress, fatigue, flutter, vibration, time dependent events, EMI, computer processing requirements and the like.

For system synthesis, functions are assigned to the various methods of implementation in the system. This is accomplished by schematic block diagrams which show interrelationships of the various functions as hardware and software implementation schemes. System elements such as end item equipments (e.g., computers, memory, CPU's, motion system, cockpit, subsystems, components, computer programs, etc.) are identified and the basic information flow and interchange between the system elements is defined. All system interfaces must be described in detail. The interfaces are very important considerations in the entire system design process and procedure. All interfaces and requirements between all major elements and all interfaces between all other identifiable system elements and personnel are documented. The system design is technically described in terms of its physical construction, its performance, its application, personnel functions, etc. Functions which are specifically allocated to computer programs are delineated under System Synthesis. A Work Breakdown Structure (WBS) of the total system should be prepared. It defines in an indentured form every end item, assembly, component, part and gives quantities required in the assembly of the trainer. The WBS also pertains to the total computer software and this will be discussed later.

TABLE I

PROCEDURE STEPS

I	SYSTEM ENGINEERING PROCEDURES
	A. Function Analysis 1. Identify and sequence system functions 2. Establish system interfaces 3. Function analysis sheets 4. Analysis Reports B. System Synthesis 1. Identify system elements 2. Interfaces requirements document 3. System description 4. Computer program function allocation 5. Work Breakdown Structures C. System Analysis 1. I/O data exchange with computer and other elements 2. Establish data formats and data rates 3. Derive mathematical techniques, equations, numerical methods 4. Determine computation accuracy requirements 5. Determine numerical accuracy, integration intervals, etc. 6. Establish coordinate systems 7. Determine computer performance and processing requirements a. Establish tentative program module list b. Establish program processing and execution requirements c. Derive average instruction time required of computer d. Establish tentative memory and storage requirements e. Determine weighted average instruction execution times for candidate computer systems 8. Make time slot assignment of tentative module list 9. Make initial selection of computer system hardware 10. Initiate program design procedures per SECNAVINST 3560.1 a. Prepare Program Performance Specification (PPS) b. Prepare Program Design Specification (PDS) c. Prepare Program Description Document (PDD) d. Prepare Data Base Design Document (DBDD) e. Perform more detailed design analysis after all program modules are identified f. Iterate procedures 7 through 9, as necessary, to conform to specifications g. Prepare flow diagrams for each program module
II	PROGRAMMING AND CODING
	A. Debugging B. Assembly C. Verification D. Linking of all modules
III	HARDWARE - SOFTWARE INTEGRATION

A major item under System Engineering Procedures is System Analysis. In some cases the system analysis work will be done prior to system synthesis. This is left to the individual contractor as to which is more appropriate for the specific training simulator being designed. This effort is documented by way of Analysis Reports. They are internal documents that may be design notes, working papers, memorandums, etc., but they should be complete in every detail since they will be the basis for delivered documentation (DD 1423's). They establish the data formats and data rate requirements for all I/O functions between computer and linkage, between linkages and cockpit, linkages and instructor's station and all other subsystems. They define and derive the mathematical techniques, simulation equations, control equations, coordinate system and numerical methods of solution required by computer programs. Numerical accuracy requirements for integration intervals and data resolution requirements of the specification must be determined.

A tentative list of program modules that have been generated is established. They are based on the simulation and control functions previously identified. Computer program processing and execution requirements are determined and the average instruction execution time required of the program is derived. Up to this point a computer system has not been selected although there may be several candidate computers under consideration. Tentative memory and storage requirements can now be established in accordance with the specification.

A weighted average instruction execution time for several candidate computer systems is computed. To do this the contractor is required to establish a percent usage of the various classes of instructions so that he can weight the average execution time for instructions for the computer systems under consideration to determine their relative processing speed capability. These average times will then be compared with the program execution requirements.

Under Analysis, all program modules are identified in a chart format for assignment for execution in each time slot or execution cycle according to the system requirements. The execution times are computed for each of these modules and they are tabulated and summed for each time slot. The worst case time required for execution in each time slot is determined and spare processing time per time slot is derived.

The initial selection of a computer is now possible from several candidates. It is also necessary that an iterative process be carried out to equate the specific characteristics of the selected computer to the programming and system analyses previously assumed. This iteration will modify the initial computer analysis and programming analysis previously completed. The tabulation of program modules now constitutes a software WBS. In accordance with SECNAVINST 3560.1 the program design and documentation requirements and procedures are conducted in the following sequence.

Under The System Analysis step, a Program Performance Specification (PPS) is prepared. This document will describe the performance required of the computer system. It specifies criteria in terms of operational, functional, system architecture and mathematical language. It should provide the basic information for the Computer System report later on. Some of this information will have been determined in the previous analyses, but it is now to be specifically related to the exact program organization. SECNAVINST 3560.1 describes how this information is to be documented. Next, the Program Design Specification (PDS) is prepared. The PDS specifies the design details and organization of the complete computer software required by the trainer specification and the system analysis effort. Program Description Documents (PDD) are required and must be prepared at this point in the design sequence. The PDD specifies the design details, programming and coding requirements for each program module identified in the software WBS previously mentioned. It should be emphasized that these documents are to be prepared and approved by NTEC before any coding is initiated. They are the design documents from which a coder (or programmer) will code each module of the complete real-time software. The document is a detail delineation of the function, interface, mathematical, programming, timing, solution rate, logic and language requirements related to that particular module.

The Data Base Design Document (DBDD) provides a complete description of all data and numerical constants, including I/O parameters and operands required by the full real-time software. It identifies all common data base items according to type, purpose, size, use, range of values, formats, scalings, mnemonic representations, and provides a cross-reference to the individual elements and memory locations, and where they are used in all program modules. Since a common area in high speed memory is now specified by NTEC in which all data, all constants and I/O items which are globally referenced throughout the program are stored. No data items are to be included in the body of instructions comprising each module. They are to be referenced from this "common buffer". This document should describe how the data base will be set up, how it will be accessed, how it will be maintained and include any other information necessary to organize and access the data.

A second and more detailed analysis must be made after all subprogram modules have been identified. The revised program figures are to be determined and a revised time slot scheduling of each module accomplished to rederive a worst case time usage per time slot. To a certain extent this must still be an estimate, but now it is a more educated estimate. The analysis procedures involved are repeated as necessary to maintain worst case execution time within simulator specifications. If a cycle time exceeds specification requirements, timing must be adjusted by reallocation or re-scheduling of modules, or a re-definition of elements to bring the execution time within the timing constraints. If this becomes impossible to achieve, selection of a more capable computer must be made. If a single computer can not be selected because of the processing workload and speed requirements of the program, then the software must be divided and reallocated among two or more computers, CPU's or processors as the case may be. If this latter is required, then the procedures just outlined must be reiterated and an additional analysis performed for each processor.

Program flow charts are to be prepared for each module and they are to be included as an item in the PDD prior to NTEC approval. The flow chart must show all processing and program logic necessary for that module. All interfaces with other modules or hardware must be shown in detail.

All analyses and all Systems Engineering Procedures carried out as indicated above comprise an activity that will require at least three to four months to complete after contract award. But by doing first things first, benefits will accrue in the form of complete design documentation, system performance and schedule performance.

Design reviews must be held at appropriate times as shown on Figure 1. A design review may be the vehicle by which the required design documents are approved before formally submitting them. This must be worked out contractually for each case. After approval of the program design approach as represented by the documentation of SECNAVINST 3560.1, then and only then will actual programming and coding be started for Procedure Step II of Table I. Within the total design procedure, after each module is programmed, each must be debugged and verified by using appropriate test data. Each module should be tested with the range of variables it is to accept. The initial testing should be accomplished independent of the special trainer hardware and documented as such. Each time a change is required or considered for the PDD, the change or modification must receive approval before the change is introduced in the respective PDD. The design and implementation of the computer system (hardware and software) will be controlled by means of these media.

When the programming and coding have been accomplished, and the initial debugging. assembly and verification of all modules have been completed, and after the various modules have been progressively assembled and linked to form the complete real-time program, the entire program must be exercised with representative inputs to get representative outputs. The running time per time slot must be measured. A final design review should cover the complete program, the verification data and outputs, the measured running time, and all required design documentation for the complete computer system.

The main purpose of the design review at this time in the schedule is to preclude any surprises at full trainer test and acceptance or at the time of hardware-software integration. If modifications were made to the contract for Engineering Change Proposals (ECP's), and these inadvertently were not incorporated into the software, they should be addressed at this point in the design procedure and not at final hardware-software integration or final test and acceptance. At this design review the software test plan should also be reviewed. This document is the software test procedures and plan submitted in accordance with SECNAVINST 3560.1. If it is determined that the programming and coding are complete, then final integration, test and acceptance as scheduled can proceed.

There are two approaches to hardware-software integration, Procedure Step III of Table I. It can be accomplished on an incremental basis in which there is a gradual build-up of the total system. Or, the total program can be completed, all hardware components assembled and then integration is initiated. The procedure to be followed should be dictated by the particular or specific technical and contract requirements for a given trainer. If the programming and coding, and the debugging of individual routines (modules) are complete, and they are gradually linked to form the complete program, and all interfaces have been reviewed and determined to be proper, then perhaps software can be finally assembled and integrated with the system hardware. This method has several advantages. At least two major system areas have been independently checked-out and any significant internal problems discovered and corrected. These two areas are the trainer hardware and the complete trainer program. Only minor problems should manifest themselves if the individual checkouts were properly accomplished. The software can now be integrated with relative ease on a very abbreviated time schedule. If such a procedure is not followed, there will be hardware problems, software problems and a combination of hardware-software problems that will occur. Their solution usually requires extensive manpower and extends the integration and check-out schedule. The procedure for design as delineated in this paper should measurably assist in overcoming this problem area. A better trainer design should result with little or no additional costs and with a more confident schedule.

It is my opinion that the system engineering and design procedures described in this paper will achieve a better and more cost-effective trainer. It is recognized that additional system engineering effort and additional analyses are required to carry-out these procedures. However, a large portion of the dollars normally assigned to an extensive hardware-software integration effort can now be shifted to the system engineering effort. A significant number of technical problems associated with the computer system will be alleviated or be significantly reduced. The programming and coding should become a very rote activity under the system engineering procedures outlined above. The programming effort should be simple and can be assigned to more junior personnel and still result in an excellent output. Obviously, more experienced personnel must review their work. But the more experienced and technically mature engineers should perform the systems engineering. They should control all facets of the effort with systems engineering being their biggest job. Hardware design and selection is also an important part of the total trainer design, but the design of a totally responsive trainer should be the overwhelming system consideration. As far as the computer system is concerned the system engineering personnel are the key individuals in the program design and implementation. It is just not possible for a single computer engineer to properly coordinate or supervise all facets of the total computer system.

CONCLUSIONS

This discussion is somewhat abbreviated in areas, but documents have been referenced that cover the essentials in considerably more detail. Procedural steps have been outlined that will result in a better system definition and a better computer system that will meet all specifications. Such a system should be more cost-effective overall because it will be a design that has been thoroughly analyzed to be more cost effective before implementation. The procedures as outlined will overcome marginal system performance previously due to marginal computer hardware and software. Math model deficiencies will also be overcome since suitable analyses will have been accomplished to define minimum requirements. The procedures outlined will assure the computer will be matched to a system design that has been adequately defined in advance of implementation.

Proper program documentation will be achieved inherently when the requirements of SECNAVINST 3560.1 are met during the system engineering phase. The data will be accumulated in a form which is readily deliverable as a DD 1423 item of the contract. The cost of this effort will have been largely assumed in the system engineering and system design phases of the project.

REFERENCES

[1] Air Force Systems Command Manual 375-5 Series, "Systems Engineering Management Procedures"

[2] Department of the Navy, SECNAVINST.3560.1, "Tactical Digital Systems Documentation Standards."

[3] Naval Training Equipment Center, Director of Engineering Memorandum 3900.3, 4 April 1975 (NAVTRAEQUIPCEN Internal Document)

[4] Technical Report: NAVTRAEQUIPCEN 1089-1, Revision 1, "Simulation System Programming Design Manual," C. F. Summer, April 1973, Section II, Section III.

ABOUT THE AUTHOR

MR. CHARLES F. SUMMER has been a Project Engineer in the Computer Laboratory of the Research and Technology Department of the Naval Training Equipment Center since 1970. He holds a B.S. degree in Electrical Engineering from the University of South Carolina. At the Center, Mr. Summer has been involved in applied research in simulation computing techniques for trainers and simulators. He has also acted as a consultant in computer technology for various Center departments and other Navy and Army agencies. He has more than 20 years experience in the design of digital computers and their application in complex systems. Prior to joining the Naval Training Equipment Center, Mr. Summer was associated with various firms such as RCA, Harris-Intertype Corporation, and Martin-Marietta Corporation. He is a senior member of the Institute of Electrical and Electronic Engineers and IEEE Computer Society. Mr. Summer is also a member of Tau Beta Pi and a Registered Professional Engineer in Florida.