An integrated support software network using NSW technology

by RICHARD A. ROBINSON and EMILY A. KRZYSIAK

Rome Air Development Center
Griffiss AFB, New York

INTRODUCTION

The problems with support software and the management structure employed in acquiring major weapon systems within the Air Force have been documented in an excellent fashion by General McCarthy, and his remarks are used to introduce the subject of this paper. In addition, plans are currently under way to tie together the many organizations involved in major system acquisitions, and to utilize ARPANET and National Software Works (NSW) technology as the framework for a series of technology demonstration(s). Finally, the reader should note that substantial efforts are currently under way within the Air Force to standardize and control programming languages and compilers.

"Program managers are faced with a major challenge brought about by the rapid expansion in use of digital computers in our modern weapon systems. It is estimated that the Air Force spends in excess of one billion dollars annually to make required changes to existing embedded computer programs. This cost will continue to grow as we bring into the inventory an increasing number of more complex digital systems. Given potential snowballing support costs, there is the real possibility that the Air Force will not be able to afford the required support posture for future weapon system embedded computers unless we change our current management philosophy.

It appears that major software systems and their attendant management structure have grown without an overall master plan or long range goal to guide the developer, maintainer, and user organizations. Today there are at least 37 different regulations and policy letters on Air Force management of software systems. Some of these regulations and policies provide conflicting guidance.

In the late 1960s and early 1970s, a typical weapon system had less than 100,000 words of embedded software, usually handling only one or two fairly straightforward functions. Current estimates place the operational software for the Joint Tactical Information Distribution System (JTIDS) in excess of 400,000 words. The operational software for the E-3A is over 500,000 words implementing 275,000 instructions. This represents only 9 percent of the total E-3A weapon system software.

This volume of embedded software is a major factor in the high support costs. Another is the expense of changing mature software. A representative cost to develop a line of embedded software would be $40-$75, depending on complexity of the program and development tools available. Once the weapon system has been fielded and the configuration baseline has been established, it may cost upwards of $4000 to change that same line of software.

With software's inherent flexibility we can add functions or integrate systems to provide rapid and effective response to threat changes or to advances in technology. We stand to lose this inherent flexibility and seriously impair our mission readiness posture unless we develop a strong support capability whether it be organic or contractor furnished.

WHAT IS SUPPORT SOFTWARE?

For purposes of this paper, support software will be defined "as the set of programs you need to develop the software you want." A representative set of programming aids (or tools) is as follows:

- Communications Aids
- Compilers
- Linkers
- Mgmt Info Systems
- Online Doc Aids
- Standards Enforcers
- V & V Aids
- Requirements Analysis Tools
- Assemblers
- Online Editors
- Data Base Mgmt Systems
- Code Auditors
- Automated Testing Systems
- Simulators

HOW IS SUPPORT SOFTWARE ACQUIRED?

In order to appreciate why support software is so costly, it might be best to review in more detail how support software gets acquired on a major system acquisition. Typically,
a user organization (USER) identifies an operational need. A subsequent study effort is awarded to further describe the operational requirements and/or develop functional specifications. A System Project Office (SPO) is created for coordinating system acquisition efforts. Procurement actions are initiated, a prime contractor (DEVELOPER) is selected, components of the job are identified, digital systems are integrated, etc. Note that little if any attention is paid to support software. The basic support software is found or developed by the prime contractor, and the criteria used is that “if something is once proved adequate, it will be used again because of schedule, monetary, and other constraints.”

The support software that gets developed in this fashion is developed out of desperation rather than by design. It is generally crude and inefficient because it has not been developed by specialists, nor has it been developed for use by others. Neither does it take advantage of current technology. Definite tradeoffs occur between the sophistication of the support software, and “getting on” with the job. For example, a program manager may decide it is not cost effective to use an on-line editor on his project because it is currently not available, it is buggy, or his staff has no experience with it. Therefore he uses punched cards, etc. As a result, the support software ends up being very expensive, inefficient, and unusable during subsequent phases of the system life cycle, or on other programs.

Once the system has been developed and tested, it is transitioned to the logistics command (MAINTAINER) for support on behalf of the operational command (USER). The accompanying support software is frequently viewed as inadequate or unusable because there is no documentation, they are not familiar with it, they have not been trained to properly support it, and frequently are not able to take over and maintain the system using it. As a result, the logistics command is forced into assuming development contractor responsibilities rather than concentrate on maintenance (or resource management) functions.

WHY DO WE HAVE PROBLEMS WITH SUPPORT SOFTWARE?

In view of the above remarks, let’s try and summarize why we have problems with support software so that we can get a better handle on what to do about it. According to Sof-tech’s analysis of the problem: (1) it is expensive to develop and funds may not be available; (2) it takes time to develop and time may not be available; (3) it is not planned that others may want to use it; (4) it is not developed nor maintained by specialists; and (5) user access (via networks) has not been provided nor is it conveniently available.

Even if it is available, support software is generally not suitable during subsequent phases, or on other programs because: (1) it is developed as a “one-time” application specific package; (2) it is operating system and machine dependent; (3) it is crude and inefficient, and is poorly documented; and (4) it requires such extensive modification that re-development may be easier, more timely and less expensive.

WHAT CAN WE DO ABOUT IT?

It is proposed that the following steps be taken to address the support software problem:

- Establish an Integrated Support Software Network — to provide DMU personnel with convenient and timely access to a repository of proven, high quality tools.
- Provide a staff of software specialists and consultants — to assist DMU personnel in selecting tools from the repository, and in assessing the utility of the selected tools.
- Standardize, control, and distribute the tools that have been determined to be most useful and cost effective.
- Maintain the tools as required by Air Force (or DoD) policy, and keep abreast of new technology developments.
- Demonstrate feasibility and practicality of the ISSN concept.

With the support of ISSN staff personnel, DMU personnel would log directly into the ISSN and request an inventory list of available tools. They would then select from that list those of most interest, and try those they find to be most useful for their needs. Once the decision is made to procure the tools, they would be delivered via the network to their respective machines. Supporting documentation would be provided on-line, and in hard copy form. If this were not feasible because of machine non-compatibility, or possibly machine non-availability, DMU personnel would simply use the requested tool kit or remotely located ISSN machine(s) until a dedicated, project-owned machine became available. The tool kit, or selected tools would then be transferred to their host environment for dedicated project use.

HOW DO WE DO IT?

As illustrated in Figure 1, the overall ISSN concept can be broken down into the following elements: (1) a computer network; (2) the NSW system; (3) a “core” facility; (4) a number of general purpose/experimental machines; (5) a number of project-specific (native) machines; and (6) a professional staff of software specialists, dedicated to supporting the needs of DMU personnel, including maintenance, training and documentation.

The network

A computer network, e.g. the ARPANET, would be used to tie together the major machines involved in the acquisition of a major weapons system. It would also be used to provide real-time communications between the many organizations involved in formulating requirements and specifications, and during design, development, and subsequent maintenance of the system. The network is designed to provide efficient communications between heterogeneous computers so that hardware, software, and data resources can be conveniently
and economically shared by a wide community of users. The ARPANET currently links a wide variety of computers at Defense Advanced Research Projects Agency (DARPA) sponsored research centers and other DoD and non-DoD activities in CONUS, Hawaii, Norway, and England.

The NSW system

Once a network connection had been established, the National Software Works (NSW) system would be used to provide DMU personnel with convenient access to the distributed computer resources, including those available within the core facility (described below). The NSW is a distributed software system which resides upon the ARPANET host machines, and provides a user with single point access to resources (or tools) on those machines. It also obviates the need for DMU personnel to know host operating system or file system details.

The core facility

The major element within the ISSN is the "core" facility or tool repository, where DMU personnel can get the latest information on tool technology, and where they can assess the suitability of and possibly acquire high quality tools for use on their respective projects. The core facility might consist of three types of machines - DEC 20 (TOPS 20), IBM 360 (OS), and DEC 11/70 (UNIX/PWB), all suitably hosted on the network described above. Extensions can readily be made to include other mainframe computers, e.g. UNIVAC.
1110 (EXEC 8). Geographic location of the machines is not important because of the network connection. It is important, however, that the core facility be professionally staffed and run by a responsible organization.

The tool repository contains a variety of tools (or toolkit). The fundamental notion is that the same toolkit would be available to DMU personnel throughout the system life cycle. Individual tools would be simply added to or removed from the toolkit from time to time, depending upon DMU needs. A basic toolkit might consist of an on-line editor (TECO), a compiler (JOVIAL), a program support library (PSL), and an automated testing tool (JAVS). Additional support software would be obtained from the tool repository, as required. The decision on which tools to include in the toolkit would depend on the particular acquisition phase, and the functions to be performed.

The core facility can also be thought of as a repository where prospective tool vendors can install proprietary or unproven tools so they can be properly evaluated by core facility/DMU personnel. Of particular interest is the issue of whether the tools properly communicate with each other. The “do it yourself” syndrome invariably leads to the development of stand alone tools that do not communicate with each other, e.g. the many dialects of JOVIAL that currently exist, and of the many problems that occur because of the prevalence of divergent interfaces. More rapid assessment and evaluation of this aspect of tool technology is possible through the use of NSW technology. When DMU personnel become convinced of the applicability of a tool, or a tool kit, they will import the selected tool(s) (with supporting documentation) onto their “native” machines.

General purpose/experimental machines

These are machines that currently exist within the NSW System as tool bearing hosts (TBH) and would be utilized by DMU personnel for such things as electronic mail, training, off-loading when dedicated (native) resources are saturated or are not available, or when specific tools (or tool kits) are not available on the native machine(s). Machines available for this purpose are TENEX (DEC-10) and TOPS-20 (DEC-20) machines at USC-1SI, an IBM 3033 at UCLA, a TOPS-20 (DEC-20) at RADC, a MULTICS (H6180) at RADAC, an EXEC 8 (UNIVAC 1110) (location to be determined), and a UNIX (DEC 11/70) system. Note that the “core” machines identified above are also on this list. Actual details will depend upon DMU needs.

It should be noted that general purpose/experimental machines can be used for supporting specific phases of the acquisition process. For example, dedicated SREM or CAD-SAT machines can be used to support requirements definition, a PDL machine for design support, etc. It is not unrealistic to suggest that a machine be totally dedicated to a specific function, e.g. requirements analysis, and would service multiple DMU organizations. Access to these resources would be obtained (and controlled) at each DMU site.

Project-specific (or native) machines

These are usually owned and operated by contractor personnel or government agencies, and are located on-site and under the control of contractor or government personnel. For the ISSN concept to be demonstrable, it is necessary that the operating system on the native machine be compatible with one of the core facility machines. Exceptions to this rule should generally be discouraged or forbidden because of the obvious and frequently detrimental impact on associated support software. Specific tools (or toolkit) to support development of the weapon system can then be imported from the core facility using the network connection. These tools will consist mostly of standardized, high quality tools that have been registered with the Federal Software Exchange and are maintained and serviced by a professional staff of software specialists. If native or dedicated program resources are not available, or are otherwise saturated during peak load conditions, the developer will also have the option of using a remotely located general purpose/experimental machine.

It is required that the native machine(s) be on or have convenient access to the ARPANET, and be machine/operating system compatible with the core facility machine/operating system, i.e. a TOPS-20, OS 370, UNIX PWB. This is necessary primarily because of the cost (and time) of re-hosting the support software obtained from the core facility; also because of the requirement for core facility staff to maintain the rehosted tools. Upon completion of development and testing, the DEVELOPER turns over project owned machines to the logistics command (within the Air Force) and they are installed in a support center specifically set up to maintain each major weapon system. These facilities are located within the individual air logistic centers (ALCs). In order to complete the scenario, it is necessary that these support centers also be on or have convenient access to the ARPANET to ensure continued use of the same tools used during earlier phases.

A professional staff of software specialists

For the ISSN concept to be demonstrably successful, the staff located within the “core” facility must provide timely service to DMU personnel, and must effectively maintain and control the tools which reside within the repository. This includes the servicing of software trouble reports (STR) submitted by DMU personnel on tool problems.

RECENT EVENTS ARE BEGINNING TO LOOK PROMISING!

Although recent efforts in language standardization and control look promising, much remains to be done to change the basic procurement/management practices that are used to acquire and support major systems. Some of the changes that are currently under way that support the ISSN concept

From the collection of the Computer History Museum (www.computerhistory.org)
are as follows:

**MIL-STD-1589A**
- A single language for Command & Control
- Standard interface is likely
- LCF standardization & control possible

**MIL-STD-1750**
- One instruction set
- Standard linker possible
- Standard assembler possible
- Standard debugger(s) possible

**MIL-STD-1533B**
- Standard I/O protocols likely
- Environmental models are easier to adapt
- Aids V&V tool standardization

**DAIS Tech Demo**
- Flexible avionics executive
- Flexible structured digital systems

**AFLC Tech Demo**
- Establish network/NSW applications
- Install network connections
- Install standard support facilities (PWB)
- Provide NSW resources

**WHAT CAN BE ACCOMPLISHED?**

Now that we have described the elements of an integrated support software network, what can be expected to happen if it were suitably staffed and demonstrated? It should be possible to demonstrate:

- The support software development burden can be removed from the system program offices (SPO). Support software requirements, specifications and high quality tools can be provided to the SPO's, and contractor and SPO personnel can conveniently assess the quality of the support software before buying or committing it!
- The support software maintenance and enhancement burden can be transitioned to the logistics command, where the distribution and use of tools employed during system acquisition can be used to responsibly maintain major weapon systems.
- The developer, maintainer and user efforts can be focused on improving techniques for advancing weapon system technology rather than building up an inadequate support software capability on each program and for each phase.
- The quality of weapon systems can be improved.
- Life cycle costs can be substantially reduced.
- Support software can be made immediately available for subsequent programs.
- Most importantly, it will be easier to establish a corporate memory from phase to phase (within a project) and across projects, and effectively apply new advances in tool technology.

**WHAT IMPROVEMENTS CAN BE EXPECTED?**

Some of the improvements that can be expected if proven, high quality support software is utilized by DMU personnel are:

**Requirements analysis**
- Life Cycle Costing Possible
- Less Costly Maintenance
- Distribution Problems Resolved
- Training Problems Simplified
- More Timely Development
- Responsible R&D Support

**Documentation**
- Standardized and Controlled
- Distribution On-Line
- Under Configuration Control

**Integrated tool kit**
- Library of Tools (selectable)
- High Quality Tools
- Reliability Demonstrated
- Effectiveness Demonstrated
- Interoperability Demonstrated

**Availability (of ISSN)**
- For Training Support
- For Tool Assessment (planning)
- For Tool Evaluation (project)
- For Tool Utilization
- For On-line Assistance

**Adaptability (of ISSN)**
- For Special Problems
- Readily Available
- Maintenance & Engineering
**Support**

- Adaptation Assistance
- Requirements Analysis Assistance
- For Local Site Support

**SUMMARY**

This paper has described an integrated support software network (ISSN) which can be used by the DEVELOPER, MAINTAINER and USER organizations in acquiring and maintaining large, computer-based systems. It focuses on the problems of technology transfer (it is virtually non-existent) within the software engineering business and how an "integrated" support software network could be used to alleviate or solve many of these problems. Networking and NSW technology are proposed as the vehicle for tying together the various machines, resources and organizations involved in major system acquisitions. Application of this technology provides a means for changing acquisition management practices within the Air Force, and should result in substantial cost savings and more timely delivery of major weapon systems.

**BIBLIOGRAPHY**

1 McCarthy, General James, Brig Gen, U.S. Air Force, AFALD/AQ letter, entitled "Embedded Computer Software Management Concept."
7 SOFTECH seminar presentation entitled, "Solving JOVIAL Support Problems for Embedded Computers; A Controlled Approach."