DEFSMAC

The Future Is Now

SHAUN D. STINE

1. INTRODUCTION (U)

~~(C)~~ DEFSMAC (Defense Special Missile and Astronautics Center) went into operation as a joint NSA/DIA element in June 1964. It was created to provide "a management for (a) control of DoD missile and space intelligence collection and processing activities directed against foreign missile and space activities and (b) current analysis and reporting of foreign missile and space events."[1] Its charter further specified DEFSMAC responsible for "(a) twenty-four-hour surveillance of foreign missile and space activity; (b) tasking and technical control of DoD collection activities against foreign missile and space activity; (c) providing technical support, including tip-off, to all DoD missile and space collection activities; and (d) current analysis and reporting of foreign missile and space events based on data collected and received by DEFSMAC up to seventy-two hours after an event."[2]

~~(FOUO)~~ As one can imagine, this mandate causes DEFSMAC to deal with a vast array of field sites and other Headquarters organizations to perform its mission. This leads to an enormous amount of data and a confusing mix of computer hardware, software, and communication circuits, as well as the different communications protocols that each system "speaks," to handle the data. Figure 1 shows, as of June 1991, the approximate layout of systems in DEFSMAC.

~~(FOUO)~~ The Data Systems Division, responsible for all DEFSMAC computer and communications systems, began laying plans for a DEFSMAC-wide Local Area Network (LAN) in conjunction with the other divisions of DEFSMAC in 1987. The LAN, part of W1's NETHERLAND effort, would serve all branches of DEFSMAC – Systems, Intelligence and Operations – allowing users to share data and run multiple tasks from the same workstation.

~~(FOUO)~~ The workstations and servers of choice for this configuration were Sun Microsystems products – the High Performance Workstations (HPWs). Most networks, even up to the potential size of the DEFSMAC LAN, use one continuous length of Ethernet cable. Planners instead decided to use Network Systems routers. The reasoning behind this was to allow for a measure of redundancy in the event of network problems. Using two routers, up to sixteen strands could be implemented; two of these were meant for the

1. (U) U.S. Department of Defense, Directive S-5100.43: Defense Special Missile and Astronautics Center (Defense/SMAC), 27 April 1964.

2. (U) Ibid.

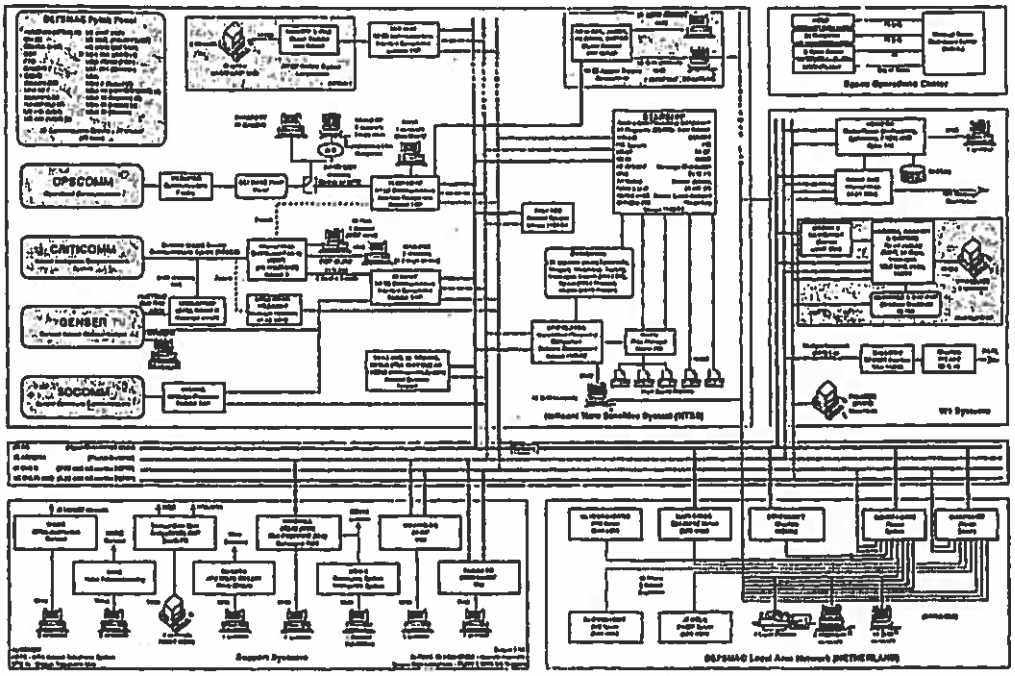


Fig. 1. (S) DEFSMAC and its data flow

as-yet nonexistent CLOVER network. This layout, shown in figure 2a, also separated the various parts of DEFSMAC along division lines. Future plans then called for many more workstations, the aforementioned CLOVER connectivity, and eventually, fiber-optic connections (figure 2b shows a more current router set-up, while figure 2c shows a map of machine and printer locations).

DEFSMAC Time Sensitive Operations Network

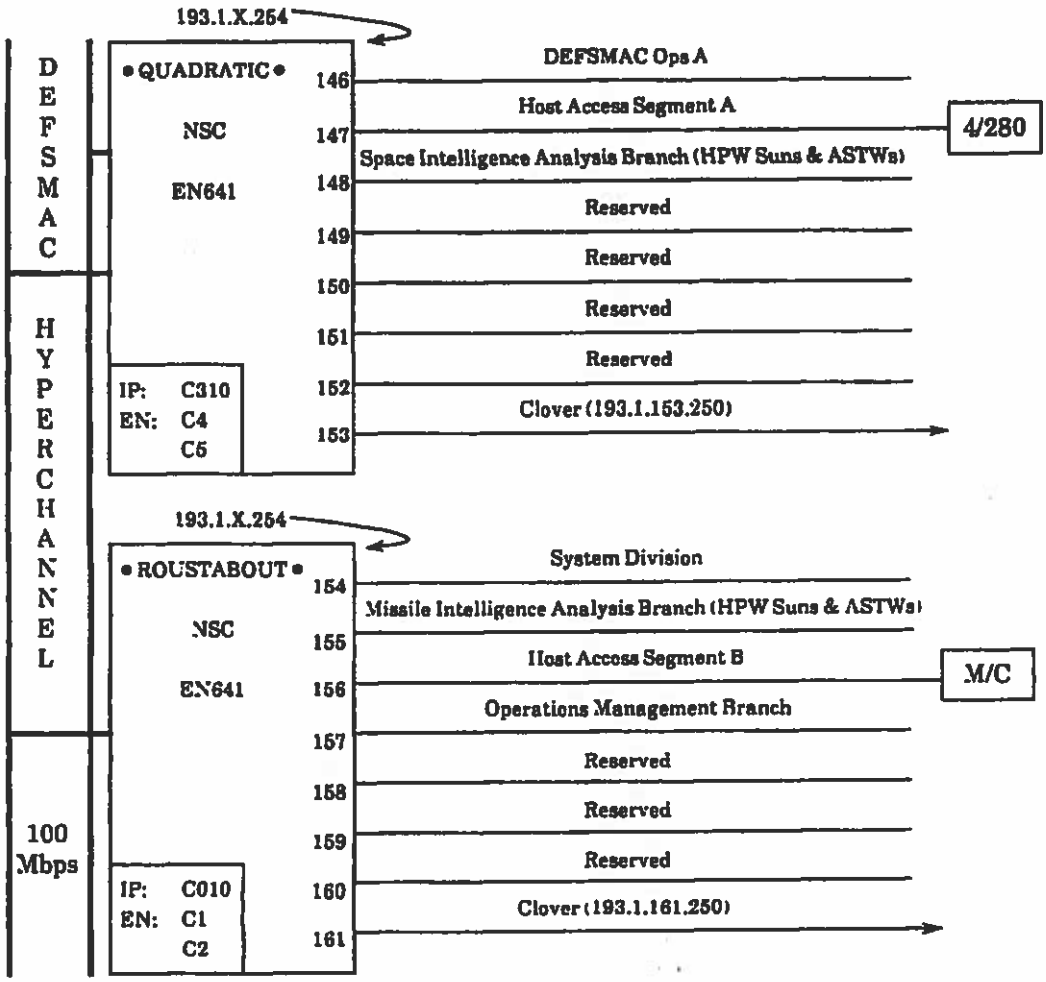


Fig. 2a. (C) Proposed router and network layout, October 1988

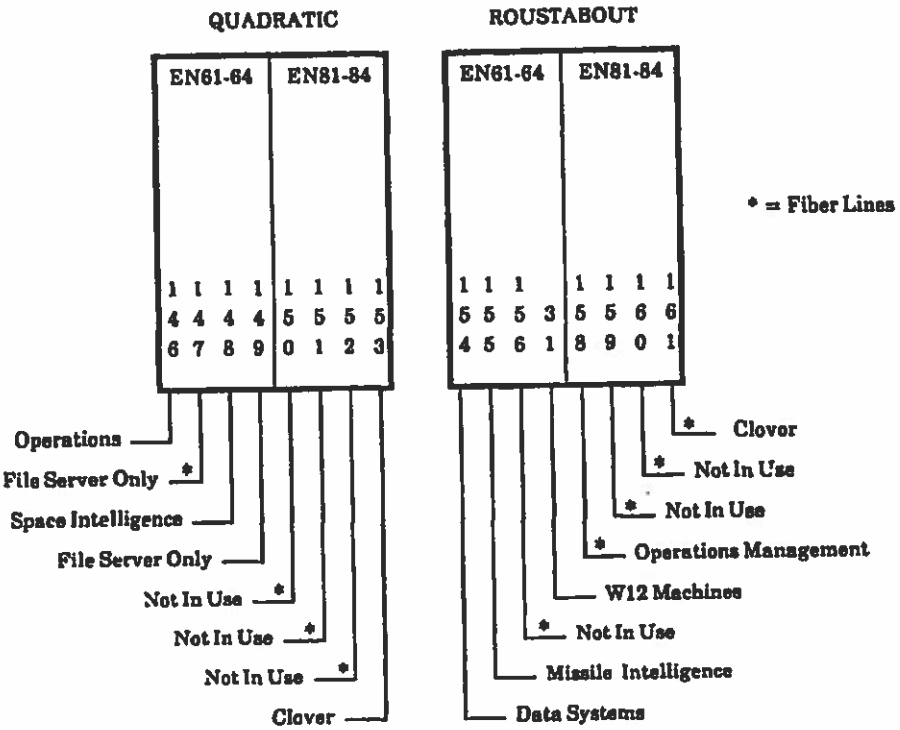


Fig. 2b. ~~(S)~~ Router and network layout, June 1991

~~(S)~~ ~~(FOUO)~~ In September 1988, the first Sun systems arrived (two 3/160s, with two 3/150s as their clients) and were simply connected to each other by a single Ethernet cable. In the two and a half years since, the number of systems has grown to about forty, approximately 110 users have accounts on those systems, and the routers and some of the fiber-optic cable are in place. Have we made progress? Yes. Is DEFSMAC fully up to date? Hardly. The rest of this paper documents the successes, failures, trials and tribulations that have been encountered to this point. Since many organizations are also going down the same "garden path," it will also offer some observations and advice on what others can do to repeat the successes and avoid the pitfalls.

2. PERSONNEL ISSUES (U)

(U) People, being less consistent than machines, are the hardest to plan for when designing and implementing a new architecture. Employees come and go, management loses interest and finds other directions to turn, and morale problems arise over learning curves – many things can sidetrack or kill a project. This section deals with the important issues involving personnel – management's role, user acceptance, and employee training.

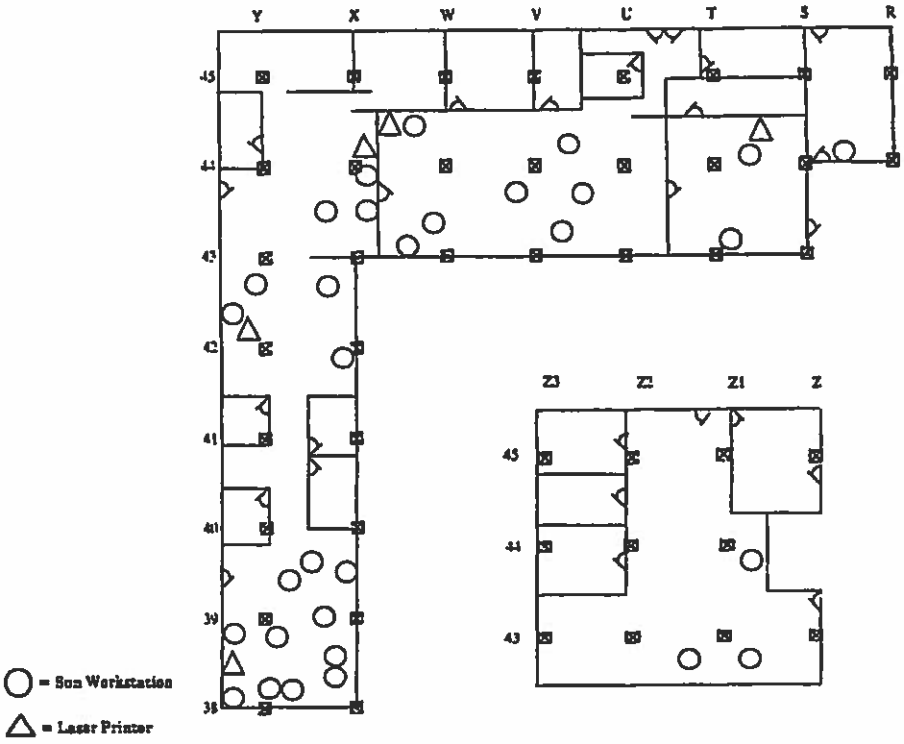


Fig. 2c. (U) DEFSMAC workstation layout, June 1991

2.1. Management (U)

(U) Regardless of who initiates the idea for an upgrade or modernization, management is the key to the life of that project. If management loses interest, resources for the upgrade get cut. Either financial support for the project is diverted, or staffing for the project is shifted to other places and projects. Even if resource levels are maintained, the tone set by the various levels of management influence the work force. If mid-level managers are open and willing to use a new system, acceptance by the employees will come fairly easily. However, if they are reluctant or openly rebellious, the employees tend to be the same.

~~(FOUO)~~ In DEFSMAC, the latter was the case. Operations management was, for the most part, against the new Suns. In turn, this feeling filtered down to Mission Directors, and Suns were slow to be placed on the Operations floor. Unfortunately, this attitude was unproductive in two ways: the entire effort to upgrade was held up while feet were dragged, and most Operations personnel were actually interested in the new systems but were being denied the opportunity. Other divisions in DEFSMAC, while leery, were not strongly opposed, and their users were more open to change.

(U) The solution to this problem is actually very simple; its execution is the difficult part. Upper-level management must set a firm tone that the transition to the new systems will be followed and not allow parts of the organization to slow the process. Also, when – not if – there is conflict between the ADP personnel and user groups, the upper management, while being sensitive to the users' needs, must strongly back the ADP group, as long as they are following the overall "plan." Unfortunately, tacit backing will not do : visible support sends the message to users that they will not be able to stonewall new systems just for the sake of stonewalling. It also gives ADP people a feeling of support, knowing they will not be left to fight their battles alone, and provides a tangible help to productivity by raising morale.

2.2. Training (U)

(U) Once the machines arrived, almost every member of DEFSMAC was at the same level of ability. Unfortunately, that level was not very high, even in Systems. Most people had never worked with UNIX, and those who had, had not necessarily used a full, multi-user version, much less a windowing environment like Sun.[3] Our first challenge, straight out of the blocks, was to get the analysts prepared to use the machines and the "computer types" ready to create applications and "troubleshoot" for users.

(U) Three methods of training were used, each with its own set of advantages and disadvantages. The first course chosen was to contract directly with Sun Microsystems for class time, rather than wait for the National Cryptologic School (NCS)-sponsored slots. The chief advantage was the sheer number of people that could be taught at one time; Sun classes ran from twelve to fifteen people per session. That way, everyone started from the same point at the same time.

(FOUO) However, the drawbacks tend to outweigh that advantage. Obviously, budget determines if an organization can afford to go this route. DEFSMAC itself was able to do this only twice, in fact. Another problem arises directly from the advantage – while a contracted class allows many people to be taught, what if there aren't enough people to fulfill space requirements? There may not be quite enough people to satisfy minimum enrollment. Also, once the Agency pays for a class, it is charged the same amount whether the minimum or maximum attendance is attained. (In both classes contracted by DEFSMAC, W12 personnel attended along with members of Systems to fill the class.)

(U) There are two more serious problems, however. Some people in the classes lacked experience and had no background training. In effect, they were "jumping ahead" of themselves trying to learn basics and the given subject matter at the same time – and running the risk of slower learning and low retention. For some topics, like system administration, this is not too serious – the class is designed for a slower pace, so students can learn "what to do if" – but in a programming class, it can mean trouble for the entire class. For example, if a person knows little about an editor, much less how to compile and

3. (U) PC/IX and AUX, being on personal computers, often do not include full UNIX capability and commands, and still allow for only one "logged on" user at a time.

link a program under a given operating system, how can that person actually learn the programming techniques being taught?

(U) The other problem was that to fill the classes, people were sent who should not have been there. Learning ability was not the issue; everyone there was capable of learning the subject matter. However, certain people, either through actual job description or by the reality of their day-to-day work, were not going to use what they learned in the office. Of course, some may dislike a subject once they see it and not want to work further with it, but that is always a risk with any course. In reality, though, how many System Administrators does an organization really need? Aside from sounding like a bad riddle, this is a question that must be planned around. An organization like DEFSMAC should get by with two System Administrators.

~~(FOUO)~~ The second method involves using regular Agency channels and acquiring classes through NCS-sponsored slots. The advantages here concern money. First, there is no extra expenditure by the Agency – rather than send people to a separately purchased class, they are put into "prepaid," prescheduled slots. Also, for the individual organization, no money has to be set aside to send personnel – the slots are already budgeted to the "alpha" organization by E Group. The disadvantage is that the competition for some classes is so keen that one individual organization may take months or years to get a significant number of people through the classes they need. People who have taken a class may attempt to help those who haven't when they return, but that is a poor substitute for actual classroom learning. DEFSMAC had purchased its equipment early enough that demand at the time for training was light, and Systems personnel were able to attend more courses than expected.

(U) The last method is to perform internal training of some sort. If an individual or group is well versed in a subject, they can be tasked to teach the remaining members. However, experience shows, for whatever reasons, that form of training is not as effective as others, most likely because the day-to-day job is hovering just in the background, not allowing for total concentration on the subject at hand. Another way is to have contractors train in-house, usually as part of the contract. As part of Project VARIATOR, Booz-Allen employees offered elementary training on Suns, McIntoshes, and FrameMaker to DEFSMAC personnel. However, before the job can interrupt the training, you have to get to the training session! If sessions are strictly voluntary, there will be plenty of empty slots in the trainer's calendar, and even more no-shows. In addition to being inefficient for the trainer, this system leaves many people untrained and unprepared for the new systems.

(U) Having read the previous paragraphs, one may think there is no method of training without dire consequences. Unfortunately, this is true to a point. The key is not really the methods, but rather the planning beforehand. If people are sent to classes simply because money appears or open spaces exist, the wrong people get in the wrong classes, some people get no training to speak of, and morale suffers from the perception of

an "arbitrary" management method. These problems have plagued DEFSMAC at one time or another.

(U) What is the solution? As with most problems, some advance planning of needs and requirements will help immensely. One way is to create a roadmap of sorts, dividing members of the organization into various types – programmers, end users, administrators, etc. Planners can decide what combinations of classes offer the most training value. At first, everyone starts with the same courses; then, systems people can branch to programming and administration, while users move to classes that allow them to use the machine more efficiently. Rather than go into excruciating detail, figure 3 shows a map DEFSMAC could use.

(U) Once the map has been planned, it is relatively easy to place individuals in priority order, depending on their job in the organization. Availability of classes, addition of new personnel, individuals' schedules, and mission essential activity will force some changes in the list of priorities, but the overall scheme should be flexible to allow for these changes. As is the case with most things, if a little thought is put in early, there is much less work later.

2.3. User Acceptance (U)

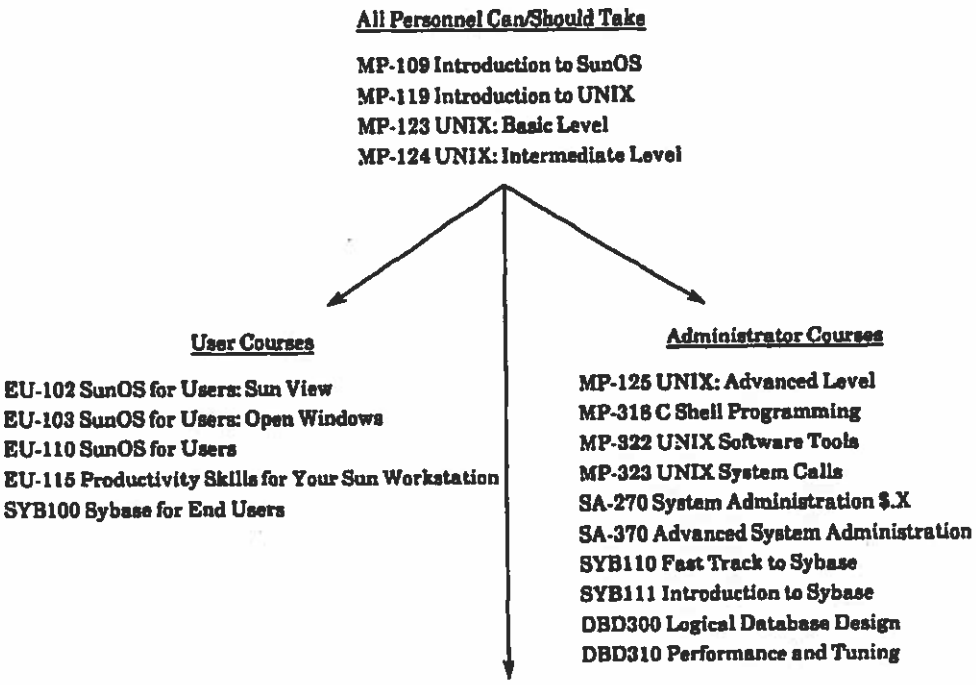
(~~FOUO~~) With management backing and a training scheme in place, convincing the user population to "make the switch" is easy, right? Wrong. Human nature being what is, when confronted with something new, the first instinct is to cling to the old ways. This happens even when the old methods are hated, as was the case in DEFSMAC. Users did not like the turnaround time of STARSHIP,[1] and they wanted newer and better machines to work with than the IBM PCs they used. However, once new systems began arriving, the old ways became good enough, and people were "too busy" to learn anything new.

(U) The best way to entice users to make a transition is to at least equal their current capability. This sounds obvious, but systems integrators and applications programmers have been known to leave out options that weren't "sexy" enough to include but were very important to the user in his everyday job. We encountered both the good and bad sides of this when giving users some form of PC compatibility.

(S) For example, most of our users use COASTLINE software on a PC to communicate with STARSHIP via the NTSS.[2] There are other packages and internally written programs running in DOS, but they are usually little programs that automate repetitive tasks that can probably be rewritten for the UNIX machines. COASTLINE, however, is universal throughout DEFSMAC divisions, and it is life-or-death for the analyst population to retrieve data and release product. To provide compatibility in the Sun-3 line, there is a

1. (~~FOUO~~) A Unisys 1100 that performs DEFSMAC processing, located in the basement of Ops 1.

FOUO 2. (U) NTSS – National Time Sensitive System. COASTLINE – Analyst Support Component for the NTSS that provides distributed processing.



Note: Designators with MP-xxx are NCS courses; other two-letter designators are Sun courses, while three-letter designators are Sybase courses

Fig. 3. (U) Training map

combination of software emulation/hardware card; in the Sun-4 family, the solution is software emulation only. This was the good side – these solutions worked well with COASTLINE, and users could save their message traffic under their UNIX accounts, if they wished.

(S) The down side? Sun was moving away from that software and entering into a partnership with a company whose product was called SoftPC. This product, while having various new "bells and whistles," like EGA and VGA monitor emulation, did not do what we needed it to do – run COASTLINE – thus rendering our solution useless. Fortunately, Sun has agreed to support, if not sell, the original emulation package for a period of five years past the termination of the project, and they are currently looking into the problems with COASTLINE. Hopefully, at the end of the five years, there will be some program that will be successful, or the XMI project, currently run by J34, will be ready to take the place of the DOS emulation.

(U) Another way to encourage users is to provide them with things to "play" with on the new systems. This is not meant in a derogatory way; it is normal for people to want to work with things they can enjoy using. Giving users a windowing environment was definitely a plus for the Suns. Even if users did not know what they were, there were plenty of different tools and utilities that come with operating system with which they could experiment and still do no damage to the computer. The level of confidence users have in their Systems people also plays a role in their willingness to change; if they feel the support they currently get is not adequate, then they will not be as confident in the newer systems and vice versa.

(U) Finally, the previous two subsections, *Management* and *Training*, alluded to the impact each has on the user population. Management reluctance is reflected by the users, and the transition becomes a political struggle. The absence of a good training plan, or training of any kind, discourages users and makes them feel as if they have been "thrown to the wolves" – forced to learn machines on their own.

(U) The seemingly best solution is to present a unified front for the user population. Start by gaining management's complete support. Then, three different paths need to be worked simultaneously. Of course, one of these is the technical aspect – the pieces of equipment to buy, how they hook together, and what protocol they speak to each other. The second path is the training plan mentioned in section 2.2; the course work required should be researched and mapped out as soon as the different types of hardware and software have been chosen.

(U) The last path, mentioned in this section, is the application path – the software that users will "play" with and/or use in their everyday jobs; this includes both in-house and off-the-shelf software. The beginning stages of this path must be started with the other segments. Actual coding, of course, will not start until the hardware and any software tools arrive, but the first steps of the true software engineering lifecycle, such as requirements gathering, as well as the identification of which programs to port and which to get rewritten, can be done in conjunction with the two previous paths.

(U) If these three paths are followed and carefully planned, the new systems and architectures become more palatable to the users. There will always be jitters and a little grumbling, to be sure, but users will have a higher level of confidence, seeing that most contingencies are planned for. Surprises will still occur, but users can see the framework is in place to deal with troubles without throwing everything in disarray. Also, morale will rise for systems people as well, knowing they are well prepared; since users have more confidence in them, there will be less contention between the groups.

3. HARDWARE ISSUES (U)

(U) Of course, once the planning stages are complete, then specific hardware decisions must be made. First, do you go with solely one manufacturer, or do you buy primarily from one vendor, with other companies filling gaps as needed? The "open systems" concept has made the idea of using multiple vendors easier than before, but some people still prefer to lean on just one company. As far as UNIX machines are concerned, DEFSMAC has gone strictly with Suns and are fairly satisfied with the choice.

(U) Then, more decisions must be made concerning individual machines. Do you buy diskless or diskfull, and if you get disks for each machine, how many disks and how much total storage per machine? How much RAM? Color or monochrome screens? Finally, the "noncomputer" equipment must be considered. What peripherals do you buy, and to what machines are they attached? What kind of network-related equipment do you get – gateways, routers, bridges, or a combination of the three? And what about fiber?

(~~FOUO~~) Unfortunately, the speed with which technology moves works heavily against making solid decisions; the best plan can easily be outdone by a company's product announcement or termination. As if making hardware choices isn't hard enough, those who plan must also take into account changes in technology. For example, in the three years since the first Sun arrived in DEFSMAC, Sun Microsystems introduced and expanded its line of SPARC machines and abandoned the 68XXX Motorola chip. In addition, the physical space required to store the same amount of data decreased. Because of those changes, a person can get 16 or 32MB RAM, 208MB of disk space, and a 3.5" floppy disk drive – about the same or better than the Sun 3/150s that first arrived in DEFSMAC – and it can fit on a desk or even a SpaceSaver, like a PC, thanks to the "pizza box" – a unit about 16 inches square and 4 inches deep.

(U) Aside from growing pains from time to time, DEFSMAC Systems people have been relatively satisfied with the choices made on hardware. The following subsections discuss the different types of equipment and the decisions and outcomes relating to them.

3.1. Workstations and Fileservers (U)

(U) The first lesson we learned was not to purchase machines that are diskless workstations. The first four Suns, mentioned in section 1, were two servers and two

clients. From a technical point of view, this did not present a major problem; in fact, some offices have nothing but diskless machines for their users. We simply felt that having disk space is a benefit, because the operating system, windowing software, and other packages can be stored on each machine as needed, reducing the amount serviced by fileservers. Every machine purchased since then has its own disk space, and the original two clients have disks as well. Most of our workstations have about 200MB; some have as much as 327MB, while the file servers have in the range of 688MB to 1.5GB.

(U) Partitioning a disk in a UNIX system makes software and machine management easier. Usually, disks are partitioned into user space, space for the operating system and other software, and sometimes space for varying length administrative files. Each partition corresponds to a directory in the filesystem hierarchy, i.e., typically / (root), /home, /var, and /usr. A partition cannot be split across two physical devices, so choices had to be made as to how to partition machine's disks and still have enough room in each partition to be of use to the workstation. Figure 4 shows some of the ways DEFSMAC machines have been partitioned.

(U) As far as memory was concerned, not much thought was given -- whatever standard amount the factory put in was the RAM we received. In the earlier machines, 8MB was standard, but 16MB has become the normal amount now. In fact, if a workstation is going to use Open Windows, or any X windowing system, 16MB keeps the amount of "dead" time to a minimum. Lastly, we have bought color monitors with all our Suns; most people prefer a color screen no matter what kind of system is being discussed, but an organization pressed for procurement money may be better served by monochrome (They may also save enough to buy additional machines.)

(U) Depending on the approach to the network layout, if the idea of centralized files and data is used, then larger machines will be needed as fileservers. While one server per network sounds logical enough, DEFSMAC currently has six active strands, and only four machines have sufficiently large storage areas to be considered fileservers (and just three are for users). The cost for us, or any organization that uses this idea, is too high. The plan we used was to mount users (discussed later) on one of the larger machines and to divide user accounts evenly among the machines. The fourth machine serves data, executables, and the manual pages.

(U) What's the ideal machine? Again, new hardware makes that a tough decision; in fact, in the time it took to complete this paper from the first day of writing, a new workstation may come along that will be the state of the art. However, at the risk of being considered biased, the author believes the Sun family of products has served DEFSMAC and other organizations well, and it is well thought of in the outside world. As long as the machine meets minimum memory and storage requirements, includes a windowing environment (specifically X Windows, as that is the direction of "the world"), and comes from a company that appears to stay at the head of the industry, any workstation will be fine.

~~U//FOUO~~

Partition	Start_Cyl	Blocks	Size	Mount Pt
a	0	32562	16	/
b	18	63315	32	(swap)
c	0	1344087	688	(the entire disk)
d	0	0	0	(unused)
e	53	195372	100	(used by Sybase)
f	161	293058	150	/usr
g	323	732645	375	/home
h	728	27135	13	/var

This is an example of a fileserver partitioning. Sybase is more efficient when using a 'raw' UNIX partition.

Partition	Start_Cyl	Blocks	Size	Mount Pt
a	0	16170	8	/
b	77	48930	25	(swap)
c	0	204540	104	(the entire disk)
d	0	0	0	(unused)
e	0	0	0	(unused)
f	0	0	0	(unused)
g	310	115920	59	/home
h	862	23520	12	/var

This is an example of the first disk on a two-disk SPARCstation 1.

Partition	Start_Cyl	Blocks	Size	Mount Pt
a	0	0	0	(unused)
b	0	0	0	(unused)
c	0	204540	104	(the entire disk)
d	0	0	0	(unused)
e	0	0	0	(unused)
f	0	0	0	(unised)
g	0	204540	104	/usr
h	0	0	0	(unused)

This is the second disk for the SPARCstation 1.

Note: According to the Sun world, partition 'c' is always the size of the entire disk.

Fig. 4. (U) Sample partitioning

3.2. *Peripherals (U)*

(U) The most important peripheral to the user community is a printer. Despite assurances that mankind is moving to a paperless society, the reality is that society is as paperless as much as it is cashless. Sooner or later, everyone needs a printout of a file, either source code, data, message traffic, or program output.

(~~FOUO~~) As opposed to fileservers, it is easier and less expensive to have a one-to-one ratio of printers to strands. For DEFSMAC, there are five printers in the area. Most of the machines are set up to reach any printer in the area, using the /etc/printcap file. In case there is a problem with a printer or the host to which it is connected, a user can redirect to another machine; FrameMaker and other software usually have a way to direct to other printers, and *lp* and *enscript* have a flag to send print jobs anywhere. If the network in question consists of one large network, rather than small strands, then a ratio of one printer to every ten workstations should suffice, even during heavy periods. An example printcap file is in figure 5.

```
# The following is an example of a printcap file. The entry is for a printer connected
# directly to the machine containing this file. This line holds the names the printer is known
# by to the printing daemon. 'lp' stands for the default; if a user were to send out a print job,
# it will go to the printer containing 'lp.' To send another printer, most UNIX commands
# use the option '-P[printer]' where [printer] would be a name from this first line.
lp|ps|lw|dps|engineering:\
# These lines are the options associated with this printer. The backlash lets the daemon
# know the entry is continuing. The next two lines tell the operating system where the device
# is and where to write out accounting and audit-type information. The third line tells the
# technical information, like baud rate, masks, and the like.
    :lp=/dev/lw:sd=/usr/spool/lw:\
    :lf=/usr/spool/lw/lw-log:af=/usr/adm/lw.acct:\
    :br#9600:rw:fc#0000374:fs#0000003:xc#0:xs#0040040:mx#0:sf:sb:\
# The following lines are put in here by Transcript software. They let the daemon know what
# filters to use on the output. In this specific case, the filters are for PostScript.
    :if=/usr/local/lib/adobe/ps/psif:\
    :of=/usr/local/lib/adobe/ps/psof:gf=/usr/local/adobe/ps/psgf:\
    :nf=/usr/local/lib/adobe/ps/psnf:tf=/usr/local/lib/adobe/ps/pstf:\
    :rf=/usr/local/lib/adobe/ps/psrf:vf=/usr/local/lib/adobe/ps/psvf:\
    :cf=/usr/local/lib/adobe/ps/pscf:df=/usr/local/lib/adobe/ps/psdf:sh:
# This is a brand new entry. It tells the daemon about a remote machine on the network.
# 'rm=exxon' means use the machine known as exxon on the network, while 'rp=lp' says
# use the default printer on that machine. The third option is required.
pps|accounting:\
    :lp=:rp=lp:rm=exxon:
# This is another printer entry – this time, output goes to the default printer on shell.
mps|personnel:\
    :lp=:rp=lp:rm=shell:
```

Fig. 5. (U) Example printcap file

(U) The only other requirement is that each printer be capable of printing PostScript. Most software packages, both Agency and external, send their output to a printer in PostScript form. This makes a print job larger and slower, but it also allows for higher quality and graphics; of course, more memory for the printer results in faster throughput.

3.3. Network-Related Hardware (U)

(U) The simplest piece of network equipment is the Ethernet cable. Tap your machines into a length of it, and an "instant network" is created. There are limits: machines can only be tapped into the cable every 2.5 meters, and the maximum cable length for coax is 500m. (The suggested maximum number of nodes is actually in the range of forty to eighty, depending on the systems, to avoid load problems.) When these limits are reached, other pieces of equipment are needed, such as a repeater, bridge, or router.

(U) A repeater takes the packet signal and boosts it, so it can be transmitted over longer distances. A bridge connects two pieces of cable and allows them to be considered part of the same logical address area. These pieces of equipment allow the use of one logical network. To have multiple networks, and maintain internetwork communication, a router is necessary.

(U) One thing a router can do is take multiple wires and work as a bridge, treating them either as part of the same address area or as separate areas. The other is to take the strands and convert their protocol to communicate with outside hosts. The DEFSMAC routers perform both of these tasks; each one takes eight lines of cable, each line representing a different logical network, and allows machines on each network to communicate with others through an Ethernet engine. At the same time, another engine in the router allows machines attached to it to talk to other routers and hosts through HYPERchannel.

(U) The idea behind DEFSMAC's routers was to divide machines along office designator, and, as stated in the introduction, provide redundancy to protect twenty-four operations. The redundancy comes from two Ethernet ports on the back of each workstation and switching from one line to the other in case of failure. Unfortunately, this idea assumes that a failure would be in the line; in reality, the problem would usually be the router (since a cable cannot "fail," unless it was physically abused), which means the second network would also be isolated and thus no help. The other problems are that the machines in the DEFSMAC network are spread too thinly – machines in two branches or divisions could be consolidated – and that having two routers adds one more point of failure for inter-workstation communications.

(U) To further complicate matters, several workstations are, and plans say the entire network will be, hooked up to fiber optic cable. Of course, this adds more hardware to the picture. First, since the Suns and the routers have nine-pin cable ports, they need conversion devices to change the Ethernet signal into light and vice versa. Also, as opposed to a cable into which machines tap, each workstation connects through a bulkhead

to a device that is equivalent to an Ethernet cable – it sends packets to each host, and the appropriate addressee responds. These boxes may also be "daisy-chained" to allow for more hosts. The major problem with fiber is its fragility; otherwise, it has worked no better or worse in day-to-day use.

(U) The best solution, based on the experience here in DEFSMAC, would be to leave all the machines on a single logical network if at all possible and use as few separate wires as plausible. As more machines are added, use more wires, but keep them hooked together in the single logical network with bridges. Only move to a multiple network layout with a router when either the IP addresses are used up, or when the load on the wire(s) becomes too much for efficient network use. This setup minimizes failure points, makes certain administrative functions easier, and is easier for both the user and system personnel to understand.

4. SOFTWARE ISSUES (U)

(U) Deciding what software to use is at least as hard as, if not harder than, choosing hardware. Once the platform and operating system have been selected, there are still myriad software packages that run under each platform; in addition, under each type of software, there are several companies that produce a package, each claiming to be the best there ever was. Finally, as if the decisions aren't hard enough, there are each individual's biases. In any office of twenty people, you will find at least a half dozen preferences for any type of software amongst them. Add software developed by the local systems people and tbe numerous packages created by Agency personnel, and this creates quite a logjam of choices.

4.1. Purchased Software (U)

(U) Most general applications can be solved by purchasing software from vendors, just like in the personal computer world. These applications include desktop publishing, database management systems, spreadsheet, and the like. Unless there is a very specific need to create a special application, depending on things like the type of data used or the format or use of the information being manipulated, the time and effort needed to create that kind of software are not justified.

(U) The best way to buy software is to attempt to get demonstration copies from the vendor. Many vendors provide legitimate versions of their programs that allow use of the software in a limited manner. Some, like FrameMaker, just disable the save capability, while others have a temporary license that expires after sixty or ninety days. That way, users and/or the systems people can use the software for a specified time and compare to other packages. However, some companies may not want to loan their software out in this manner. Many, though, can be persuaded by the potential customer base, if their software

gains support in just one organization. Once software has had sufficient usage, then a consensus can be reached.

~~(FOUO)~~ If demonstration copies cannot be acquired, another way is to "ask around" the building, to see what other organizations have purchased and their opinions of it. Some organizations, such as the A Group Center for Applied Technology (ACAT) or certain R Group centers, can provide some guidance as to what is in the marketplace and make recommendations on the better software. Sometimes, formal reports or even technical papers will have documented comparisons of different packages.

(U) As is the case when purchasing hardware, the vendors being examined should be somewhat established, so that software will be supported into the lifetime of the package. Granted, all companies start somewhere, but it is better to have software from a company that will still be in existence, able to provide bug fixes, upgrades, and other forms of support. In our case, Sun Microsystems puts out the Catalyst Catalog, a listing of software vendors that sell software for Sun computers. These vendors have entered into agreements with Sun to be a part of the Catalyst program, and in a sense, can be considered "trusted" vendors.

(U) Site licenses should be bought when possible so as many copies as needed may be used. Another issue is how copies of the software are disseminated. FrameMaker, for example, uses floating licenses, something to which other software vendors are moving. DEFSMAC purchased twenty licenses, but received one tape from Frame Technology. After running a script and entering a password, FrameMaker keeps track of how many users are running the program and shuts out any user after the twentieth. We mounted the software from a fileserver to any workstation that would use it, and it is used on first-come, first-served basis. Thus, we need only one set of executables to service twenty people.

(U) The bottom line is that software buying, like hardware, can be a hit-or-miss idea. The most important thing to remember is set a standard early, and stay with it. Some argue against standards, saying they stunt creativity; however, in the application software field, it makes things much easier for everyone. Systems people do not have to worry about maintaining separate packages for each type of application, and data files are much easier to pass between users, as there are no worries about converting formats.

4.2. In-house Software (U)

~~(C)~~ Of course, not everything lends itself to purchases straight off the shelf. DEFSMAC, for example, has a program known as SMACPOST that communicates with various tracking sites, relaying data and tasking information. This program, written by a DEFSMAC programmer along with some "borrowed" code from POSTMASTER, has been used in DEFSMAC since about 1984, making it a fairly robust program for the PC world. Currently being ported to Suns and renamed HEIRSHIP, SMACPOST was not the kind of program that could have been readily bought from a vendor, except as contracted software.

(FOUO) So far, in the new DEFSMAC LAN, there is little code that has been newly written and/or ported from older systems. Many of the applications that reside on STARSHIP will be ported as the DEFSMAC Local Database design of STARGLORY[6] continues. Most of the current support Systems provides is help with the new machines, helping users with a new operating system, and setting up some of the purchased applications.

(U) However, there are lessons to be learned regarding software and the Agency, not only from DEFSMAC experience, but from general observation. Note that the rest of this section, even more so than the rest of this paper, is strictly personal opinion, and should not be construed as the official policy or opinion of DEFSMAC.

(FOUO) One of the biggest current problems is the duplication of effort. In DEFSMAC, there is an operational need for mapping software, both to track space and missile events and for informational purposes. Fortunately, we turned to programs written by other internal organizations. Unfortunately, there are many organizations that have written mapping software. In fact, at least three mapping programs – SUNSHINE, GALACTAN, and IMPRINT – are currently in use somewhere in DEFSMAC spaces.

(U) There is nothing wrong with writing several variations on the same theme, such as the mapping example, since there is always a certain amount of fine tuning or feature adding that each version provides. The problem is that every major effort is written basically from the ground up. The computer science catch phrase for this is "information sharing," or the lack thereof. When a difficult programming task in encountered, the wheel is reinvented.

(U) The best way to combat this problem is to set up some form of repository of software modules and routines, documented in some manner. Depending on their needs, programmers could then research and find code relevant to the tasks they are programming. This central library would not enforce standards – that is up to the management of each division. In other words, if a group wants to continue programming applications in the SunView windowing system, or if they want the MOTIF "look and feel" for their windows, as opposed to OpenLook, they can stay with that choice.

(U) Could this work? Different platforms used at the Agency mean object code would prove difficult to keep. Either executable code could be stored after compilation, or users could download source only, then perform compilation on their own machines. The library office would also not be responsible for code correctness – that would have to be certified by the organization that submitted it.

(FOUO) The biggest problem would be creating the library organization. Previous attempts at similar ideas – W Group briefly had a software engineering working group designed to let each other know what software existed – seemed to work only because of personal interest by a few people. When those people change offices, the group dies. An Agency-wide group would need to be in an individual's job description, rather than

6. (C) STARGLORY is the joint DEFSMAC-J2 project to upgrade DEFSMAC's ADP effort.

something he does "on the side." To be fair, the size of the Agency makes it tough to implement, but an Agency-wide commitment to the idea could make it possible.

(U) The other problem is more morale than technical. In many cases, software is contracted to outside employees, while Agency employees are left with maintenance or contractor management. Granted, there are large amounts of software that need to be written, but the Agency hires or trains large numbers of computer analysts. There have been several divisions with problems keeping employees because plum assignments have been contracted out. From a technical standpoint, problems arise when the contract ends and Agency employees must maintain code they may not even have had a hand in designing or coding. DEFSMAC has not contracted out any work on the Sun network, as J2 personnel are the "contractors" working with us, but these types of problems have arisen before in other contracts.

(U) From what software that has been written on the new systems in DEFSMAC, there are a few final recommendations. If the machines on the network are UNIX machines, the languages of choice should be C or C++, since they are the traditional languages used with UNIX, and they will allow for portability. Also, development software that builds windowing code is a definite help to the programmer, as well as debugging tools that go beyond vanilla UNIX. Lastly, set standards immediately for user and programmer documentation, and enforce it early and often. That sounds trite, but DEFSMAC has programs on its old system whose documentation has not been updated in four or more years, and the code has been repeatedly altered, sometimes beyond recognition. Aside from that, there are enough ways to approach development that each office must make its own decisions.

5. ADMINISTRATION (U)

5.1. User Accounts (U)

(U) Before, when the machine of choice was an IBM-PC/XT/AT, there was a ratio of about one machine per person. Everyone kept his data on his "own" PC, and since DOS is a single-user operating system, there was no need for account information. However, UNIX is for multiple users, and the ratio of Suns per person in most DEFSMAC divisions is less than 1:1. This means users may not be able to use the same machine every time. How, then, do users keep data current on different machines, and how does the System Administrator keep accounts current on dozens of different machines?

(U) DEFSMAC solved the problem by using a combination of Sun's Network File System (NFS) and Network Information System (NIS). NFS is a "machine independent network service that allows application transparent sharing of file systems or directories among nodes on a network."[7] Our use of this, simply put, was to "mount" directories from

7. (U) Sun Microsystems, System Administration 4.X Student Guide, Mountain View, California, June 1990, 10-2.

fileservers; mounting attaches a filesystem from a remote machine to the local filesystem hierarchy, causing the mounted files to appear as if they are on the local machine.

(U) Specifically, a machine was chosen for each strand to store user files, usually one of the bigger Suns. Normally, convention puts user files in the directory path/ home/machine/username. What we did was create a directory from /home that was the server name, rather than the local name, and mount the file server on that directory; this was done on every machine on the strand. It is cleaner to leave the machine name out of the path altogether, but the users do not notice the difference. To them, their files are on every machine in the area, and changes are kept in synch.

(U) There is another way of providing this service, first introduced in SunOS version 4.0, known as automounter. This service is the same as regular mounting, yet it is performed automatically by the operating system only when a request is made for those files. After a certain amount of "dead" time, the machine unmounts the filesystem. This mounting and unmounting, which is controlled by a daemon, is invisible to the user. DEFSMAC has not implemented the automounter at this time.

(U//FOUO) (U) NIS is a distributed network lookup service that allows a Systems Administrator to update administrative files on one machine, rather than each workstation. Certain files, such as /etc/passwd (account information) and /etc/hosts (host "tables"), contain information that a machine needs to accept users and communicate with other machines and networks. Within a group of machines, called a domain, there is a master server where all changes to the files are made. They are then transformed into maps and propagated to slave servers (if any are declared). The remaining machines, known as clients, can then make requests for information from these files, and any server can service that request; this provides redundancy and reliability to the process. If all the machines are in a single logical network, they can still be divided by using multiple domains.

(U) To this point, NIS has saved much work for DEFSMAC System Administrators, especially as the number of systems has grown. The only problem encountered has been with the routers – NIS maps and client requests will not work across different networks. The way the routers work, and the way NIS sends its requests and services, each wire must be a separate domain, even if multiple wires are part of a single logical network. This means six different strands must be maintained in the current implementation, but most users do not need an account on every strand so the work is not overwhelming. One solution is to maintain one overall set of ASCII files, then mail maps to every master server when a change is made – but no major effort has been undertaken.

(U//FOUO) (U) This combination of NIS and NFS is one of the best ways to maintain systems and still keep the file-sharing feel of a network. As mentioned earlier, an even better solution is to remove the machine names from the user path, leaving just /home/userid. However, our current implementation works well enough that effort to make that change would not be justified.

5.2. Backups (U)

(U) The first question that should be asked about a network, from an administrative point of view, is what is the backup scheme? Of course, that is one of the least interesting aspects of a network and thus is largely ignored or avoided. But as the network and the reliance upon it grows, the need for reliable backups becomes evident. Unfortunately, and dangerously, DEFSMAC currently has no set backup scheme. On the positive side, user files are backed up (albeit aperiodically), and we possess the equipment to make compact backups – EXEBYTE 8mm videotape drives. However, DEFSMAC users are like users everywhere – they assume that someone in Systems keeps constant backups of their files, and thus they make no attempt to back up even their own data.

(U) Actually, a schedule for DEFSMAC would be fairly easy to create. Since we currently have tapes for SunOS 4.0.3 and 4.1 for all architectures, there is no need to dump those parts of storage to tape. Most of the software aside from the operating system has been bought, and tapes from the manufacturer still exist. Of course, any customizing or licensing information would be lost, but for the most part, recovery would be minimal and easy. Any Agency-created software is also stored on tape.

(U) That leaves two important areas – user files and the upcoming DEFSMAC local database. Since individual user accounts are grouped together on file servers, it is easy to dump these files onto tape. These files could be dumped twice a week – say Tuesday and Friday – and if users feel their data are too important or are updated too often for that scheme, then they can dump them more often on their own.

(U) Databases are a different story. Since most of DEFSMAC's data gets aged off and/or archived to historical databases within seven to ten days of receipt, it is important to dump it every day. In fact, it may be wise to dump databases twice a day, to further minimize loss. Daytime dumps can be done by an administrator, while night dumps can be handled by the UNIX cron command. Fortunately, Sybase – the product used by DEFSMAC – allows for dumps at any time, even while the database is in use, so applications may still run, even during backups; the impact on twenty-four-hour operations, then, is minimal. Some individual databases may be too large for cartridge tapes, so the 8mm tape could also be used for this purpose.

(U) As a further precaution against loss of data, the databases can be mirrored. Mirroring is where two parts of the disk (or better still, two separate disk devices) are used, and changes to data are written to both parts. This makes write-heavy applications slower, as two writes are performed, but read-heavy applications are slightly faster, since either part can be read. In case of disk failure, the second device will be used with no interruption. The device can then run while the other is being repaired. Mirroring will cost more in disk space, but will make the databases more reliable.

(U) What is the real "solution" for backing up the network? Remember, in DEFSMAC at this point, the transition of users and their data is not complete, nor are the databases

on-line;[8] so the solutions given here are more like ideas, and other organizations may choose their own path that will meet their needs. The real answer is, once a scheme has been devised, to follow it faithfully (no matter how tedious), even if someone has a full-time job performing backup tasks. That way, when the inevitable disaster hits, the loss to the entire center will be minimal.

6. CONCLUSIONS (U)

(U) Without question, more and more organizations on the inside and outside are moving to networked computing, either for centralizing resources and data or just to pass small amounts of specialized data among small numbers of machines. Moving away from dumb terminal attachments to a workstation/fileserver arrangement allows for more computing power for the user, since each workstation has its own computing power to run against data received from a fileserver. Also, storing commonly used resources in a central location, such as the Unix manual pages or FrameMaker, allows each machine to be used more efficiently. Thus, networking represents a better environment for efficient office computing.

(U) That much we know; until "fuzzy logic" and other topics came along, networking was the industry rage, so the advantages are well known and documented. What we didn't know was the level of patience needed to make it work. Users are nervous, political wars break out, money runs out, and sometimes we've simply made bad decisions. At times, it feels like the network never will become fully functional, and that making even a small amount of progress takes a large amount of time. To be sure, there will be pockets of resistance, and some things we do, and data we keep, will not lend themselves to an open environment.

(U) However, most of our users are slowly coming around to the new way of doing things – they seem to realize that to get some of the "gee-whiz" things they see, they need to go beyond the PC world, and move along with technology. Also, the pace at which they've been eased into the new machinery has been slow enough – sometimes too slow – that the "culture shock" should not even be a factor. In addition, they are finally starting to get new and better applications and capabilities at their workstation, instead of small adjustments and improvements made by a Systems that are virtually invisible to the user community. Finally, anyone who uses computers in his or her daily job has to realize that change is always inevitable – in the early days of DEFSMAC, punched cards were the norm, and now, computers are starting to take advantage of CD technology.

(U) The real key for any organization to make this transition is planning. There are several things to consider – equipment, training, compatibility with other systems outside your own – but relatively small amounts of thought in advance saves an enormous amount of grief in the future. In addition, there has to be continual examination of the current

8. (FOUO) The design and building of the DEFSMAC Local Database is budgeted through FY 1994.

situation and future developments to ensure that the network can expand with time and not become a dinosaur needing future replacement.

(U) The problem with the DEFSMAC LAN was in the planning. Frankly, there simply was not enough of it. What we had was a network architecture laid out by one individual who did not easily spread the knowledge out and then left at a critical point in the network's creation. In addition, not enough serious thought was given to nonhardware issues like training and a suite of software for use once the machines were in place. Also, in all honesty, there was too much advance notice to the user community. I feel that user acceptance and Systems morale suffered more because the anticipation started too soon, and eventually disappointment set in. Users need to be told what is happening, so they can be prepared, but they cannot be promised "the moon and the stars," and they need to be aware that the actual timetable of events will probably not fit the timetable they have in mind, so they will not press.

(U) The other problem is in personnel. The Systems division of DEFSMAC just has not been staffed with enough people to maintain an old system, and create a new one, without time slippages. Some know mostly the old system, while others know mostly the new. Also, the planning function for the new network, when the task has been performed, has only one or two people performing it, which the division really cannot spare. It has been shown before that increasing the number of people on a project does not necessarily equally increase the amount of work that is accomplished, but it would help in this case.

(U) What does an organization need to do? As stated earlier; it needs to make sure there is a certain amount of management commitment, both from a financial and personnel point of view, to insure the project's survival. Once that has been established, plan all the aspects of the network, not just the technical parts. Even things that seem strange, such as the users' feeling towards the effort, and making them feel part of it, should be examined to a certain extent. Attempt to be aware of anything that may affect the network, including maintaining connections to machines outside your office's jurisdiction. Also, make sure that everyone in the organization has enough training and exposure to the machines so they can efficiently operate the machines for their job. Finally, a separate group needs to be created that examines technology developments, current capabilities, and future needs, and points the network in the proper direction.

(U) Moving into some form of networked environment is definitely worth the effort in the long run for an organization, and it will ultimately benefit DEFSMAC in the near future. It will be a struggle, to be sure, especially since there is usually an old system that must be maintained while the new one is created, but eventually things work themselves out one way or another. By heeding some of the potential problems brought out in this paper, being flexible and ready for any roadblock in the approach, and maintaining a sense of humor about the effort, your organization can survive the process intact and relatively pain free.

Acknowledgments

(U) UNIX is a registered trademark, and OpenLook is a trademark of AT&T. Sun Microsystems and SunView are registered trademarks and Sun-3, Sun-4, SPARC, NFS,SunOS, and OpenWindows are trademarks of Sun Microsystems, Inc. Sybase is a registered trademark of Sybase, Inc. IBM-PC, XT, and AT are trademarks of International Business Machines Corporation. SoftPC is a registered trademark of Insignia Solutions, Inc. PostScript is a trademark of Adobe Systems, Inc. FrameMaker is a registered trademark of Frame Technology. The X Window System is a trademark of the Massachusetts Institute of Technology. MOTIF is a trademark of the Open Software Foundation. HYPERchannel is a trademark of Network Systems Corporation.

(~~FOUO~~) The author would also like to acknowledge the following current or former DEFSMAC employees for their contributions of diagrams used at the end of this document: Kimberly Beam (fig. 2b), Howard Gay (fig. 2a), Mark Marden (fig. 2c), John Olson (fig. 1), and Patti Posa (fig.2c).

(~~FOUO~~) Mr. Stine has been a computer scientist in the Software Support Facility (SSF) at U.S. Army Field Station Kunia (F4062) since July 1992. Previously, he served in DEFSMAC Data Systems (W431) after joining the Agency in 1988. He has a B.S. in computer science from Virginia Tech and is currently completing work for an M.S. from George Washington University. He is a member of CISI and is professionalized as a Computer Systems Analyst.

BIBLIOGRAPHY

Brooks, Jr., Frederick P.: *The Mythical Man-Month*, Reading, Massachusetts: Addison-Wesley, 1975.

Callaghan, Brent, and Tom Lyon: "The Automounter," *Sun386i Administration Cookbook*, Sun Microsystems, Mountain View, California.

Coastline User Manual, U.S Department of Defense (T3413), Ft. Meade, Maryland, January 1991.

DEFSMAC Concept of Operations, U.S. Department of Defense (W11), Ft. Meade, Maryland, July 1990.

Directive S-5100.43: Defense Special Missile and Astronautics Center (Defense/SMAC), U.S. Department of Defense, Ft. Meade, Maryland, April 1964.

System & Network Administration, Mountain View, California: Sun Microsystems, March 1990.

System Administration 4.X Student Guide, Mountain View, California: Sun Microsystems, June 1990.

System Administration Guide, Emeryville, California: Sybase, October 1989.

13, 1 Spring 1994.

Distribution

Cryptologic Quarterly is published four times a year by the Center for Cryptologic History, NSA. The publication is designated as a working aid and is not subject to receipt, control, or accountability. Distribution is made through the division level; further dissemination is the responsibility of each division. Extra copies or those for which there is no further need should be returned to the Editor for disposition.

Contributions

Contributions to *Cryptologic Quarterly* should be sent to

(b)(3)-P.L. 86-36

[redacted]
Managing Editor, *Cryptologic Quarterly*
E324, Airport Square 13
Phone: 968-6158

Manuscripts should be accompanied by an abstract and should be typed double-spaced with generous margins. Two copies are required; a third copy should be retained by the author. Illustrations should be submitted with the manuscript and should be adequately identified. If your manuscript was prepared on a desktop computer/word processor, please submit a disk with the following information: the type of equipment, the operating system, and the word-processing software that you used. All material used in the publication of an article is destroyed when no longer needed unless the author requests that it be returned.

Reproduction and Dissemination

Contents of any issue of *Cryptologic Quarterly* may not be reproduced, or further disseminated outside the National Security Agency, without the permission of the Director, NSA/CSS. Inquiries regarding reproduction and dissemination should be directed to the Editor, E324, Airport Square 13, 968-6158.

All opinions expressed in *Cryptologic Quarterly* are those of the authors. They do not necessarily represent the official views of the National Security Agency.