Configuration Management Issues in the Development of the Consolidated Air Mobility Planning System (CAMPS)

J. Charles Davis (Oak Ridge National Laboratory)
Charles Dearth (Unisys)
Ursula Seastrand (US Air Force)
Glen Harrison (Oak Ridge National Laboratory)

Background

Less than a year after the Air Force was established as a separate military service in 1947, it faced a tremendous challenge—sustaining Berlin during the Soviet blockade of the city during World War II. That task was an overwhelming success as the result of massive airlift operations by the USAF and the British allies. Out of the Military Air Transport Service (MATS), the Military Airlift Command (MAC) was born on 1 January 1966, and was responsible for worldwide airlift operations for over 26 years. On 1 June 1992, as part of a major reorganization of the Air Force, the Military Airlift Command was inactivated and the new Air Mobility Command (AMC) began operations to provide "Global Reach" for America.

In the mid-1980s, the Military Airlift Command began a complete overhaul of how airlift was planned, scheduled, and executed. Many of the changes had as their foundation the "MAC Command And Control (C^2) Upgrade Architecture" study, which was accomplished by the Air Force Systems Command with extensive support from MAC personnel—from the Commander-in-Chief, MAC (CINCMAC), all the way down to individuals at the unit level. This paper addresses the area of upgrading the air mobility planning and scheduling system and the associated processes. The computer system that brought about the revolution in this part of MAC (and later AMC) C^2 was the "Airlift Deployment Analysis System" (ADANS). It was later expanded and designated the "AMC Deployment Analysis System" (ADANS), since the responsibility of AMC extended beyond just airlift; it also encompassed air refueling. The system, with additional functionality, was later designated the "Consolidated Air Mobility Planning System" (CAMPS), as the scheduling components of the system were given a new user interface and moved from Sun workstations to Windows-based PCs. To avoid confusion, the term "CAMPS" will be used throughout the rest of this paper to refer to this system, including during the time it was known as ADANS. The term "AMC" will be used to designate the Air Force organization, even though prior to mid-1992 the organization was "MAC" and after that time it was "AMC".

Development Goals

There were several high-level goals that defined what CAMPS was to become. First, it was clear that there must be a central, integrated database that served as the center of the system, and with which all the scheduling tools communicated. This went a long way toward rectifying the "stove pipe" nature of AMC's scheduling processes—where systems generally addressed only a single type of planning or scheduling. Before CAMPS there was no interaction between the various scheduling tasks and no overall, total visibility of the scheduled missions and their required aircraft assets. Second, there must be a consistent user interface. The older systems had evolved over a number of years, and without any particular standard for how the user would interact with the computer. This made it very difficult for personnel to move from one type of scheduling task to another, without spending a great deal of time learning how to make the new computer do what was desired—instead of concentrating on how to solve the scheduling challenge itself. CAMPS provided a consistent user interface across all the scheduling components. This paid particular dividends during Operation DESERT STORM when manpower had to be moved from some of the peacetime scheduling functions into contingency scheduling operations. The third goal was to develop a system that planners could use in a consistent way, whether working in peacetime or wartime. In other words, allow for consistent screens and methods across the spectrum of operations—allowing a smoother transition to fast-paced, crisis operations. And lastly, there was a significant need to upgrade the processes and technology used to schedule missions. The software being used was inefficient and at times made assumptions that were completely unreasonable. For example, the software used to schedule exercise and contingency missions would sometimes schedule crew rests or crew changes at an ***air*** refueling point. (That's a schedule that crewmembers are not thrilled to see.) The scheduling process was slow and did not take into consideration many of the critical items that were limiting constraints. Basically, the computer system could not produce a *flyable* schedule. In an era when contingencies were occurring more frequently, AMC was in need of a completely new system that addressed the challenges outlined above.

AMC Mission and Scheduling Processes

AMC's primary mission is rapid, global mobility and sustainment for America's armed forces. U.S. forces must be able to provide a rapid, tailored response with a capability to intervene against a well-equipped foe, hit hard, and terminate quickly. Rapid global mobility lies at the heart of U.S. strategy. As U.S. forces stationed overseas continue to decline and global interests remain, rapid airlift deployment provided by AMC is even more important. AMC provides the capability to deploy U.S. armed forces anywhere in the world, and help sustain them in a conflict. AMC air refueling capabilities enable U.S. and allied aircraft to have increased range, payloads, and flexibility. The command also plays a crucial role in providing humanitarian support and disaster relief for hurricane, flood, and earthquake victims in the U.S. and around the world. In recent years, command tankers and airlifters have supported peacekeeping and humanitarian efforts in Bosnia, Iraq, Cambodia, Somalia, Rwanda, Haiti, and Afghanistan.

AMC's Tanker Airlift Control Center (TACC), at Headquarters AMC, Scott Air Force Base, IL, is the command's hub for planning and directing tanker and transport aircraft operations around the world. The TACC was created to centralize command and control responsibilities previously located in the numbered air forces (NAFs) and airlift divisions. AMC has ten stateside bases, six AMC units at non-AMC bases, and 39 Air Reserve Component units.

AMC's strategic aircraft includes the C-5 Galaxy, C-9A Nightingale, KC-10 Extender, C-17 Globemaster III, KC-135 Stratotanker, and the C-141 Starlifter. The stateside based C-130 Hercules is AMC's tactical airlifter. AMC's active airlift assets include 252 strategic airlift aircraft, 257 air refueling aircraft, and 88 additional aircraft. In addition, the Air Guard and Air Reserve have 106 strategic aircraft and 296 air refueling aircraft. AMC has over 150,000 staff on active, civilian, Guard, and Reserve duty. They fly over 300 missions each day.

Specific types of missions flown by AMC include:

- Channel missions that are regularly flown on scheduled routes at specified intervals (i.e., daily, weekly, or monthly basis).
- Special Assignment Airlift Missions (SAAMs) that are single purpose missions usually flown for the delivery of specific cargo or troops.
- Contingency and Exercise missions that are flown in support of military forces involved in specific operations, such as Enduring Freedom in Afghanistan.
- Air Refueling missions for strategic and tactical aircraft.

In addition, AMC conducts Deliberate Planning to determine the feasibility of supporting potential contingency operations and to plan for the resources needed to support those operations.

Implementation Approach

Review of operations. At the start of the project, the Oak Ridge National Laboratory (ORNL) development team spent time with each of the scheduling components at AMC headquarters and the numbered air forces to gain an understanding of the scheduling processes. The team also reviewed the software that was being used to determine its functionality.

Requirements analysis. The development team worked with AMC staff to develop a set of requirements that needed to be met to support the Deliberate Planning process, as well as, airlift scheduling for Channel, Special Assignment Airlift, Contingency, Exercise, and Air Refueling missions.

System Design. The development team worked very closely with AMC staff to design a database and software system to support air mobility scheduling. The focus of the design was on a common normalized database, efficient scheduling algorithms, and similar user interfaces. The design of the integrated database required substantial analysis of existing data structures and an understanding of scheduling processes. The collaboration of different AMC scheduling groups to agree on naming conventions and configurations for the data tables was an essential part of this process.

Development. The development stage involved writing the software for the database, algorithms, and user interfaces. A team of programmers was assembled for each type of mission. The leader of each team coordinated with other teams to ensure compatibility between the different modules of the CAMPS software. A database development team was responsible for the overall implementation of the database that interfaced with each of the different scheduling algorithms. User reviews were integrated into the development process to ensure that the users had input into the software development as it progressed. After each user review, the team did an assessment of any required changes and the resulting changes in the design, software, schedule, and cost.

Independent Validation and Verification (IV&V). Before any software was delivered to AMC, an independent IV&V contractor tested it to ensure it met the requirements and specifications in the design. Software problem reports were recorded and corrections to the problems were made. The software was then retested before it was sent to AMC for installation. In addition, if there were problems with the software after it was installed at AMC, software problem reports were used to document the problems and resolve them.

Initial Operating Capability. Each module of the CAMPS software had to be certified as reaching an initial operating capability by the staff at AMC. Once this had been done, the software was used for scheduling operational air mobility missions.

System Configuration

There are a number of areas that define the system configuration of CAMPS. We will first describe the system during the development years—starting in 1986 and continuing through 1996. Later in this paper, we will discuss the changes to the system during its operational years. We will describe the hardware, software applications, database, and user interface.

When the CAMPS development began, AMC had just purchased a large, fairly powerful, mainframe computer, with the intent of using it to enhance airlift scheduling capabilities. The original thought was to do software coding in FORTRAN and use PCs (AT vintage 286s) as the user "workstations". Fortunately, when the AMC and ORNL development team began laying out the roadmap for the new planning and scheduling system, they quickly realized that the mainframe with PCs as terminals was not the appropriate road to travel, and a modified architecture was developed. Initially, plans were made to keep the mainframe connected to the system, but to use it only for demanding algorithm work. The rest of the system would be composed of desktop workstations to host the user interface and some of the applications, and desk-side workstations to provide database and file server support. As development continued, it became apparent that the cost and complexity of interfacing with the mainframe added little benefit—especially since the power of the desktop workstations was rapidly increasing each year, and these workstations also provided a scalable architecture that could grow and expand as necessary. The idea to include the mainframe was eventually dropped, and no CAMPS code was ever written to run on it. The Sun Microsystems family of workstations was chosen to support the system, both as the desktop workstation and to meet the system's server needs.

A very robust hardware configuration was developed, primarily by the AMC part of the development team, while the ORNL part of the team concentrated on the software needs of the project. Computing tasks were split between multiple servers, to provide maximum computing power and system redundancy. For example, the database was placed on one Sun server while a different machine handled the other file server functions. However, each of these servers "backed up" the other one. A backup database was running on the primary file server, and the file server backed up its information on the primary database server. Though it happened only once, when one of the servers failed, all the processing functionality from that machine was rapidly switched to the other server, and the system remained operational with some slight (almost negligible) degradation in performance. When the inoperative server came back up, the two servers were resynchronized and processing was again divided between the two machines.

At the time CAMPS was initially brought on line operationally (early 1990 for the first component), AMC had a NAF on each coast—21AF at McGuire AFB, NJ and 22AF at Travis AFB, CA. Many of the scheduling functions were performed at the NAFs, so the system architecture had to reach out to those locations, as well. Connectivity to workstations at the NAFs was provided via leased telephone lines. Though the primary database remained on servers at Scott AFB, backup copies of the scheduling data were transferred several times a day to the NAFs, to provide the capability to operate from those locations should something catastrophic occur that would preclude continued operations at Scott.

The system architecture was further complicated by the fact that the planning and scheduling functions were accomplished at various security levels. The majority of the peacetime scheduling was done at the Unclassified level, while most of the exercise and contingency planning was done at the SECRET level. In addition, there was some planning done at the TOP SECRET level. This required three separate systems, which in itself is not a particular problem, but it also required the capability to frequently and easily transfer data [aircraft schedules] from the SECRET system to the Unclassified system. This was implemented using a floppy disk to transfer the data, with a number of special programs to ensure that classified data were not inadvertently transferred to the Unclassified system.

The database selected to support AMC was Sybase. There were not many products that provided the required capabilities when the selection was made, and few of them ran on the Sun-based platform. Sybase was clearly in front of the other possible candidates at that time. As mentioned earlier, the database server ran on a workstation that was separate from the file server. The database could take advantage of only a single processor, even after Sun multiprocessor workstations became available. As time passed, Sybase was enhanced to take better advantage of multiple processors on the host machine. The database consists of over 250 tables, and makes extensive use of "stored procedures", which are groups of Structured Query Language (SQL) statements that are stored within the database and therefore optimized for performance.

The original user interface for the system dates back to the days before "Windows" was common on PCs; a version of the interface tool even ran on DOS based computers. The primary user interface was based on Sybase's product called Application Productivity Tool (APT). It was a text based user interface. It managed screen menus, and it had provisions for doing extensive processing within the interface, before sending the data to the database. It was quite advanced for its time, but the original system development concentrated on the addition of new capabilities, and therefore the user interface was not updated until after the initial development phase was completed. In addition to the APT screens, graphical maps and analysis displays were developed using X-windows techniques.

The application software was written primarily in two languages. During the development phase, the military “mandated” use of Ada as *the* programming language. To comply with the spirit of the law, two components were written in Ada: the PrePlanning component for Flow Planning and the Special Assignment Airlift Mission (SAAM) component. That approach, with the combination of 4GL software tools, kept the auditors at bay, while implementing the most important component of CAMPS—the Flow Planning component—in “C”. Some early algorithm prototype development was done in FORTRAN, but the flow planning algorithms that remain at the heart of AMC’s war planning capability are written in “C” code. A “Report Writer” package, SQR, provided the basis for most of the reports and file generation capability.

Configuration Management

Configuration management (CM) is the process of managing product, facilities, and processes by managing their requirements; including changes, and assuring that results conform in each case. The best configuration management process is that which can best (1) accommodate change, (2) accommodate the reuse of standards and best practices, (3) assure that all requirements remain clear, concise, and valid, (4) communicate (1), (2), and (3) promptly and precisely, and (5) assure conformance in each case. (The Institute of Configuration Management, 1988).

Configuration management played a key role in structuring the development process for CAMPS. The development process was based on a set of carefully defined measurable requirements. The design of the system focused on satisfying these requirements. The software that was developed was based on the design and rigorously tested to ensure it satisfied the requirements and met the design specifications. When problems were found in the software, they were documented, returned to the programmer for review and resolution, and retested. Each release of the software was carefully planed and implemented. Users were trained and provided with user manuals and online help.

Configuration control was a key element in keeping the system working smoothly and operational 24 hours a day, 7 days a week. A Configuration Control Board (CCB) that consisted of various members of the project development team managed the operational system configuration. Though the ORNL team members provided valuable input to the system configuration, the AMC members of the team did the detailed system analysis and management. Before changes were made to the operational system, the CCB would evaluate the impact of the proposed changes, and decide on the path to follow. A test system was put in place that replicated the operational system in great detail. Use of the test system allowed the development team to uncover potential problems before changes were made that could have impacted the system’s operational use. For quite a few years, some of the external systems with which CAMPS interfaced did not have similar test, so it was not easy to test the external interfaces in a realistic manner. This shortcoming has since been addressed by AMC.

Though hardware configuration control was essential, the real challenge in the development of CAMPS was software configuration management. The software CCB consisted of members of the ORNL and AMC development teams, plus the Independent Verification and Validation (IV&V) contractor. Though some look at CM as a method of preventing or controlling change to the system, the CAMPS development team looked at CM as the opportunity to *keep track* of change. The system was developed during an era where change was inevitable, sometimes arriving on the scene unexpectedly. For example, the Channels component was made operational in May 1990; it was only four months later that General H.T. Johnson, CINCMAC, explained to the AMC/ORNL development team that we needed to accelerate the development of the Flow Planning component by about 9 months—so he could use it to schedule what was to become Operation DESERT SHIELD/DESERT STORM. The Flow Planning component was made operational in October 1990, though it was very immature at that time. The development team had not enjoyed the "luxury" of adequate testing and refinement of the software. CM was a real challenge, with software releases being installed once a week, and sometimes more often than that. The whole development team was reorganized to accommodate changes required for this mode of operation. But the CM procedures worked well—for during the 10-year development period, never once did a software release have to be "backed out," and there was never a loss of operational data.

Configuration management of the database was a joint responsibility between the ORNL development team and the AMC team. The ORNL DBA had the lead on the database configuration, while the AMC DBA had the lead on data management. Similarly to the software environment described above, there were several test databases—both at ORNL and at Scott AFB—to ensure that changes made did not damage any existing data and were compatible with the software being used. Detailed records on the database changes were maintained, and a suite of database and data management tools was developed to help manage the database and its configuration.

Changes In The Operational System

Change is inevitable in a computer system that is being used operationally, and CAMPS is no exception. First, the hardware must change to take advantage of the faster speed and improved capabilities of new generations of computers. Particularly as more complex software is added to a system, higher CPU speeds and more memory must be put in place just to "keep up." As the functional capability that the system supports is expanded, new and even faster hardware is essential. Since the CAMPS development team selected leading edge technology at the start of the development cycle, the upgrades were fairly straightforward—just replacing older, slower workstations with more capable ones.

Another huge change that arose during the development of CAMPS was the transition from MAC to AMC in June 1992. Not only did the mission of the operating command change, but also the method of planning and scheduling changed significantly. The first apparent change that impacted CAMPS development was the additional requirement to plan and schedule air refueling missions, in addition to the airlift missions that the system had originally been designed to accommodate. At the direction of General Ronald Fogleman, CINCTRANS and the commander of AMC, an additional component was developed and added to the CAMPS suite of sophisticated scheduling tools: the Air Refueling (AR) component. To add to the challenge, AMC also wanted to develop a PC version of the AR component that could be deployed to individual tanker units. This was intended to be an interim system until AMC's Command and Control Information Processing System (C2IPS) could be deployed to the tanker units. The PC version was developed to "look and feel" like the Sun/Sybase/APT version of the AR component as much as possible. Keeping both versions in sync brought with it additional CM challenges. As a means of addressing that need, the development team implemented methods of remotely and automatically installing new software releases on the deployed PCs. Remote configuration management was a real challenge and something to avoid if possible.

The other change that the formation of AMC brought to the table was the consolidation of scheduling at a centralized location. AMC had operated in a decentralized scheduling mode for most types of missions. Deliberate Planning (war planning) was done at the headquarters. Channel missions were initially planned at the headquarters, and Contingency planning was done at headquarters, to some extent. But, the real operational scheduling expertise was located at the NAFs and the operational units. Even the Channel missions were revised and managed at the NAFs, as were the exercise missions. The "Barrelmaster", the person who meshes the aircraft schedule requirements with the available aircraft and aircrew assets, sat at the NAF, as well. The individual airlift unit did scheduling of SAAMs, with oversight by the NAF. The "stand up" of AMC and its Tanker Airlift Control Center (TACC) at Scott AFB changed all that—with the help of CAMPS. It had long been the dream of Colonel Darryl Bottjer, the AMC Director of Current Operations and one of the AMC Crisis Action Team Directors, to centralize the scheduling functions. With the new scheduling system, that capability became a reality. "I couldn't do that until we had ADANS," Col Bottjer frequently remarked. As AMC got its feet on the ground, the scheduling functions began to move to the TACC at AMC. Eventually, even the SAAM scheduling responsibility was transferred from the operational units to the TACC. Centralized scheduling, accomplished by a single organization that has visibility over the whole of the command's mobility assets, had finally become a reality.

As the original system development period drew to a close, primary responsibility for maintenance of the mobility planning and scheduling software was transferred to a maintenance contractor selected by AMC—Logicon, which is now part of Northrop Grumman. There was a period of overlap that allowed the contractor to come up-to-speed on the system for which they were assuming responsibility. The contractor had the challenge of giving CAMPS a new user interface, and moving most of the scheduling components from Sun workstations to PCs running Windows on the desktop. This involved replacing the Sybase forms and the processing that was embedded within them, and converting some code that was developed to run in a Unix environment to code that would operate in a Windows/PC environment. It also involved making the user interface look more like "Microsoft Windows" office" tools. The migration from the original

Sybase database to a corporate Oracle database has begun, but a significant amount of work remains to be accomplished in that area. In addition, the complete Flow Planning Component, as well as, the Quick Course of Action (QCOA) Toolkit, remains to be migrated from Suns to PCs and from Sybase to ORACLE. That will be a very significant challenge, since the Flow Planning component is as large (based on amount of code) and at least as complex as all the other components of CAMPS put together. That task began in April 2002.

Conclusions and Lessons Learned

The development of CAMPS began in 1986, first became operational in 1990, was used to schedule Operation DESERT STORM, and made the consolidation of air mobility scheduling (and thus the formation of the AMC TACC) possible. Dramatic changes took place during the life cycle of the program thus far, and more are probably in store for the future. One thing that is inevitable in any system is change; attempting to prevent change is futile and counterproductive. However, managing and keeping track of change, dealing with what those changes mean to the overall system, and *planning* for change is essential. Here are some lessons that were learned during the exciting development of this critical system.

Plan for system upgrades, both hardware and software, as well as system maintenance. This means not only to accept the fact that they will be required, but budget for them and plan how the changes will be accomplished and who will be responsible for them—while they are still in the future. In the early phases of the CAMPS development, the project team planned for the transition of software maintenance responsibility to a commercial contractor when the full expertise of ORNL would no longer be needed on a continual basis. Maintenance of the sophisticated scheduling algorithms, however, was not well suited to a commercial contractor, so budgets were planned to maintain a continued expertise to modify and enhance the algorithms. The project team knew that hardware would need to be upgraded and replaced periodically, so budget requests were submitted to address those known expenditures, as well.

Plan for system training. During the early development of CAMPS, the ORNL development team members spent considerable amounts of time at AMC prior to and/or after each software release. They prepared the users for the upcoming changes to the system's capabilities, and assisted with questions and problems after the new release was available for the functional expert to use. In addition, the AMC project staff had individuals who were experts in both the functional and system aspects for each component. They were in the same building, and for a large portion of the development, were in the same organization. This provided a very good environment in which to implement the new system and changes to it. In some areas, particularly late in the development period, the training tasks were not accomplished as well as they could have been. Routine military reassignments exasperated the overall problem associated with training. As a result, some extremely capable tools are used less effectively than they should be. Examples of this problem include the Quick Course of Action (QCOA) Toolkit, the User Based Scheduler, the Graphically Editable Interactive Rainbow, and Station Monitors. The lesson learned here is to recognize that continuous training of staff and available expert support are essential, if maximum benefit is to be derived for the development dollars spent.

Plan for change. Expect change and welcome change—but be sure to capture and record what has changed. Whether it is frequent software releases, such as those that this project experienced during DESERT STORM, emergency fixes to software that “broke” after it has been installed operationally, or the day-to-day changes that are made in a very deliberate manner during development; change needs to be managed—not prevented. In creating an extremely complex system such as CAMPS, it is also essential to have both developer and user involvement. Changes that may make perfect sense to a developer may not be practical from an operational standpoint—and vice versa. A team *partnership*, rather than a sponsor/contractor relationship, proved to be very beneficial for the CAMPS development effort. It made dealing with the required changes much easier than would have otherwise been the case.

Maintain a high level of functional involvement. This means two things. First, the functional end users must be heavily involved in the development of the system. Not only must they help establish the requirements, but they must also be involved all along the way if user acceptance is expected. Second, it is very beneficial to have some involvement from a *high level* within the functional staff. Particularly with development for the military community, functional experts come and go—sometimes before a component of the new system is completed and ready to use. The “new guy” didn’t necessarily have input into defining what the new system should be and what it should do, and he may not even have a vision for the new way of doing business that the new system makes possible. One of the reasons the development of CAMPS was so successful was because it had support and *involvement* on a regular basis, up through the two-star level (except during DESERT STORM, when we had regular *four-star* involvement). The strategic vision was set (and maintained) at that level, and the staff at the worker level helped define the details and make it work. As the staff members came and went, they quickly understood the *strategic* goals—for their boss understood them, and conveyed them to his staff. For CAMPS, this made a huge difference in how well the end product worked and was accepted.

This paper has used the development of the CAMPS airlift and air refueling software system to review some of the important configuration management issues that must be considered in any large-scale software development process. A key to the success of any project like this is the adherence to configuration management principles in the development and maintenance stages of the project. This is the only way to promote the continuation of a quality system that is functional for those people who use it.