ConOps: The Cryptex to Operational System Mission Success

Alan C. Jost
Raytheon

As engineering firms start to design any number of systems for a variety of customers and end users, the number and variety of system documentation can be overwhelming. Among this pile of documentation, the Concept of Operations (ConOps) stands out as a critically important engineering document that should be created at the beginning of the system development and maintained throughout the engineering life cycle. This article discusses the ConOps and if it truly is necessary in addition to all of the other documentation available.

Of course we need a ConOps! Admittedly, system development programs can be overwhelmed by the number and variety of required documentation. There are system specifications, subsystem specifications, discipline requirements specifications (hardware, software), discipline design documents (hardware, software), interface requirement specifications (IRS), interface control documents (ICD), test plans, procedures, reports, and a multitude other documents that capture what system is to be built, how the system is built, and how the system is tested. There is, however, a critically important engineering document that should be the key document developed at the beginning of the system development and maintained throughout the engineering life cycle. What is this critically important engineering document, the ConOps? It is key to successfully developing an operational system. This article covers ConOps – what it is, what it is not, and its contents. I will also show the importance of the ConOps in three situations dealing with four Air Force programs: the Over-the-Horizon Backscatter (OTH-B) Radar, the Seek Score Radar Bomb Scoring System, and the PAVE[1] Phased Array Warning System (PAWS) Ballistic Missile Early Warning System (BMEWS).

The ConOps is a descriptive document usually created by the future operational users of the system. It details what the system is going to be used for, what other systems it will be used with and communicate with, what kind of data and information it requires and supplies, how it is going to be used by the operational user, who is going to be the operational user, how it is going to get to where it is going to be used, and how it is going to be maintained. In the past, the document has been called a variety of titles (e.g., ConOps, CONOPS, and Mission Needs Statement); regardless, in this article it will be called the ConOps. In my many years in the Air Force and in industry, I have found the ConOps to be one of the most difficult engineering documents to write. Why? The ConOps is a description on how the system is going to be used. It is not an engineering document that details system requirements or describes the desired design of the system. The ConOps should be written devoid of system requirement statements and engineering design. This is the cornerstone document that drives the follow-on engineering documents, where requirements flow out of the ConOps into the system-level requirements specifications during the system requirements analysis life-cycle phase. The difficulty is keeping the requirements and design from creeping into the ConOps. The ConOps should not constrain the engineering process and its creativity in solving the operational needs of the operational users. At times, however, writing the ConOps creates a dilemma for the operational command who wish to dictate not only how the system is to be used, but how the system should be built. Who should write this critically important document, the ConOps?

Who Should Author the ConOps?

Naturally, the optimum authors of the system should be the operational users of the system. This poses a problem for a number of reasons. In some instances with the extensively long procurement and system development life-cycle time frames, the operational users of the system may not even be old enough to be in the military or in the employment pool when the ConOps needs to be initially written. Then you have the current operational users who are extremely busy performing their operational duties and have very little time to devote to writing a detailed ConOps. Some operational commands have organizations within the command to generate ConOps and future systems requirements to meet their command's operational missions. These organizations usually have a staff mix of recent operational users and engineers. In my experience, these folks are usually very passionate over improving their organization's capability in the field. They want to make it much, much easier for their future operational users – their comrades in arms. It is, however, difficult to write the ConOps without trying to drive the requirements or design of the system. Operational commands and product divisions that procure the systems usually hire engineering organizations like MITRE or system engineering firms to assist in developing the required engineering documents, and yes, even a ConOps. Again in my experience the ConOps generated by these organizations include system requirements and engineering design influences. But the operational users will contend that if they don't have the system design, it is difficult for them to write the ConOps. Likewise, not having the ConOps is a constraining factor in coming up with the engineering requirements and design – the classic chicken or the egg dilemma.

What Is a ConOps?

The Institute of Electrical and Electronics Engineers (IEEE) Std. 1362-1998 Guide for Information Technology – System Definition – ConOps Document Description provides user organizations a way to describe their missions and organizational objectives to contractors from an integrated systems point of view. The document abstract reads as follows:

> The format and contents of a concept of operations (ConOps) document are described. A ConOps is a user-oriented document that describes system characteristics for a proposed system from the users' viewpoint. The ConOps document is used to communicate overall quantitative and qualitative system characteristics to the user, buyer, developer, and other organizational elements (for example, training, facilities, staffing, and maintenance). It is used to describe the user organization(s), mission(s),

and organizational objectives from an integrated systems point of view. [1]

The purpose of the ConOps is to provide the user community a vehicle for describing their operational needs that must be satisfied by the system under development.

The ConOps approach provides an analysis activity and a document that bridges the gap between the user's needs and visions and the developer's technical specifications. In addition, the ConOps document provides the following:

- A means of describing a user's operational needs without becoming bogged down in detailed technical issues that shall be addressed during the systems analysis activity.
- A mechanism for documenting a system's characteristics and the user's operational needs in a manner that can be verified by the user without requiring any technical knowledge beyond that required to perform normal job functions.
- A place for users to state their desires, visions, and expectations without requiring the provision of quantified, testable specifications. For example, the users could express their need for a *highly reliable* system and their reasons for that need without having to produce a testable reliability requirement. (In this case, the user's need for *high reliability* might be stated in quantitative terms by the buyer prior to issuing a request for proposal [RFP], or it might be quantified by the developer during requirements analysis. In any case, it is the job of the buyer and/or the developer to quantify users' needs [and not the responsibility of the user even though they are usually very anxious to provide the ole 0.99999 reliability number instead of *highly reliable*.])
- A mechanism for users and buyer(s) to express thoughts and concerns on possible solution strategies. In some cases, design constraints dictate particular approaches. In other cases, there may be a variety of acceptable solution strategies. The ConOps document allows users and buyer(s) to record design constraints and the rationale for those constraints as well as indicate the range of acceptable solution strategies.[1]

Structure of the ConOps

By examining the IEEE's suggested ConOps structure, you can see how it is oriented around the operational user's needs. It is not a simple document to write and complete without having system requirements and design creep into the document; try to describe something as common as your next dream car without including system requirements or design in your dream car ConOps [1]. As you can see, the contents of an IEEE compliant ConOps document is defined in Section 4 of the IEEE Standard 1362-1998 – go ahead try to write one for your new dream car.

- Section 1: Scope.
- Section 2: References.
- Section 3: Definitions.
- Section 4: Elements of a ConOps document.
  - 4.1 Scope (Clause 1 of the ConOps document).
  - 4.2 Referenced documents (Clause 2 of the ConOps document).
  - 4.3 Current system or situation (Clause 3 of the ConOps document) .
  - 4.4 Justification for and nature of changes (Clause 4 of the ConOps document).
  - 4.5 Concepts for the proposed system (Clause 5 of the ConOps document).
  - 4.6 Operational scenarios (Clause 6 of the ConOps document).
  - 4.7 Summary of impacts (Clause 7 of the ConOps document).
  - 4.8 Analysis of the proposed system (Clause 8 of the ConOps document).
  - 4.9 Notes (Clause 9 on the ConOps document).
  - 4.10 Appendices (Appendices of the ConOps document).
  - 4.11 Glossary (Glossary of the ConOps document) [1].

Joint Authorship of the ConOps

While potentially creating blasphemy, I suggest the critically important ConOps be drafted by an operational command with as much operational detail as possible and included in the RFP during the initial program phase. The initial ConOps may have sections 4.1-4.4, 4.6-4.7, and 4.9-4.11. The contractor can input section 4.5 and 4.8 in their proposal. But once the contract is awarded, all parties should finalize the initial ConOps for the system under development. If it is impossible to eliminate any requirements or design content from the ConOps – at least put them in the context of suggestions. The initial effort in the engineering life cycle should be for the contractor, product division, support contractors, and operational command to jointly update and finalize the draft ConOps found in the RFP. The result is to move any system requirements from the ConOps into the associated system specification and move further detailed requirements in lower-level specifications and design documents. Also, while you want the ConOps to remain relatively stable and unchanging, the reality of the engineering life cycle is that it does take a long time to engineer and develop these systems. In the meantime, the operational mission may change and, therefore, the ConOps should be updated to reflect the current operational mission. This should involve all participants in the engineering and development of the system so that at the end of the day, not only did the contractor build the system right (i.e. met all the system requirements), but also built the right system (i.e. met all the operational user's needs). As a matter of fact, these are the three process areas in the Capability Maturity Model® Integration (CMMI®) Maturity Level 3 process areas of Verification (build the system right) and Validation (build the right system). More to the point, the Technical Solution process area has a specific practice for *evolving operational concepts and scenarios* [2].

System Requirements Continue to Reflect the ConOps

So, one of the important aspects of the engineering process is to make sure that not only are you meeting the system requirements, but that those system requirements actually reflect the operational mission of the end user. This is reflected in the CMMI model which was collaboratively written by some very smart folks in industry and government [2]. Taken as industry best practices and extensive lessons learned, one can conclude it is extremely important to keep the ConOps and system requirements relatively in sync throughout the engineering life cycle. Naturally, one would expect that the mission operational needs would match the system requirements maintained in the system specification. However, with extended procurement schedules and restricted budgets, the need to field a system that meets some or most of the requirements sometimes takes over the procurement process. If the operational mission changes, this usually results in changes to the requirements. Of course, depending on when in the development life cycle these changes occur, the cost of the resulting Engineering Change Proposal (ECP) can be very expensive

® Capability Maturity Model and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

and, therefore, the decision is made that the contract specification is *not* updated and the ECPs are not made to the contract to match the operational requirements. If many of these mission changes are not incorporated into the contract via ECP, you will have the situation where the system being built is starting to drift away from being the system needed by the operational user. Once the ConOps is developed, it is critical to make sure that any changes made to the operational mission or needs are translated into updated system requirements. These changes need to be incorporated into the systems contract. Let us examine an example of where the system's operational mission changed rather dramatically from the original ConOps, resulting in an operational test that was less than satisfactory. In addition, let us examine how creative use of new ConOps can lead to renewed use of systems whose original ConOps were no longer valid but were successfully reused after readjusting the systems' original ConOps.

OTH-B Radar

The Army-Navy Fixed Radar Search 118 (AN/FPS-118) OTH-B was produced for the Electronic Systems Division of Air Force Systems Command to fill a vital need for long-range air surveillance for North America. Designed by General Electric in the 1980s, the AN/FPS-118 would provide detection and tracking of airborne threats at ranges up to 1,800 nautical miles regardless of altitude. The Air Force's OTH-B air defense radar system is, by several criteria, the largest radar system in the world. Six OTH-B radars see far beyond the range of conventional microwave radars by bouncing their radar waves off the ionosphere, an ionized layer about 200 km above the earth. It was developed over 25 years at a cost of $1.5 billion to warn against Soviet bomber attacks when the planes were still hundreds of miles from U.S. airspace [3].

With the end of the Cold War, just months after their deployment, the three OTH radars on the West Coast were mothballed, the Central and the incomplete Alaskan Systems were cancelled, but the three radars in Maine were redirected to counter-narcotics surveillance. In 1994, Congress directed the Air Force to continue operating the East Coast OTH-B radar. The East Coast formally ceased OTH-B operations in October 1997 [3].

Here was this new radar system that used the novel idea of using the ionosphere as part of a radar system that could literally see over the horizon to get an advanced warning of incoming Soviet Union aircraft that carried cruise missiles. Because of the novel use of the ionosphere as a component of the radar, it took a long time (25 years) to develop the radar system and its software. So long in fact, that in the meantime the Soviet Union collapsed, and the threat of the Soviet aircraft with cruise missile disappeared; this occurred literally months before the OTH-B radar system was to undergo its operational testing for its operational mission. The operational command that originally contracted for the OTH-B Radar system no longer had an operational need for the system since the original threat had disappeared.

The ConOps for the OTH-B Radar changed dramatically as the operational usage of the radar system shifted to a Drug Enforcement Agency (DEA) use of the radar to monitor potential drug trafficking off the East Coast of the United States. The Air Force Operational Test and Evaluation Center (AFOTEC) were responsible for conducting the operational tests of the radar, and were using the current DEA operational mission ConOps as the guidance for developing the operational test procedures. The dramatic shift of the operational mission from the Air Force to the DEA occurred almost at the end of the development life cycle with minimal chance to change the ConOps and then go through the ECP process to change the system specification along with all the associated changes that would be needed in the system software. It is important to note, the contractor with the product division conducts a series of tests to prove that the system was built right, i.e. it passes all the system requirements or the *shalls* in the system specification, which it did with flying colors. However, the mission of the operational testing organization is to test the system against the current ConOps to make sure the system that was built was the right system to support its current operational mission. Naturally, the operational system test did not go so well since the ConOps changes could not generate adjustments to requirements at the very end of the program. This was not the contractor's fault whatsoever, but it shows an example in the extreme of what could happen when the changes in the ConOps are not reflected in the system requirements on contract. This is a dramatic example that is atypical in a discussion on the importance of a current ConOps in sync with the contract documents and system requirements. Let us examine two systems where their original ConOps were adjusted to allow the systems renewed and different missions for the operational command.

Radar Bomb Scoring System

Before sophisticated laser guided bombs, we used ground directed radar guided bombing, where the radar system would direct the flight path of the bomber aircraft to a drop point in the sky. With computer algorithms using aircraft location, meteorological inputs, and flight characteristics of the weapon, the operators of the radar system could indicate to the pilots and weapons controllers when and where to release the weapon to hit the selected targets. The operational command generated a ConOps that was flexible enough to allow the reuse of the Ground Directed Bombing System (GDBS) to be converted into a Radar Bomb Scoring System, known as SEEK SCORE. The AN/TPQ-43 SEEK SCORE is an automatic tracking radar system. This system replaced the antiquated AN/MSQ-46 and AN/MSQ-77 Bomb Directing Central systems used during the Vietnam conflict to guide bombers to their target. The SEEK SCORE AN/TPQ-43 can automatically *score* accuracy of simulated bomb releases electronically. Using computer targeting coordinates the SEEK SCORE computer performs a complete ballistics computation on any type of simulated weapon release from where the tracked aircraft is at the release point to where the *target* is. This computation provides an accurate miss distance score. The radar system can also perform a comparison of aircraft position in relation to a target to *score* the navigation and timing accuracy of an aircrew. The computerized scoring capabilities of the SEEK SCORE enhances USAF training because an aircrew can practice flying over any type of terrain at any altitude and practice bomb drops or navigation without ever dropping bombs. This clever reuse of the existing radar system capability was directly related to a ConOps that was void of design and requirements, allowing for the reuse of the GDBS system to support the new operational use of the system as a Radar Bomb Scoring System [4].

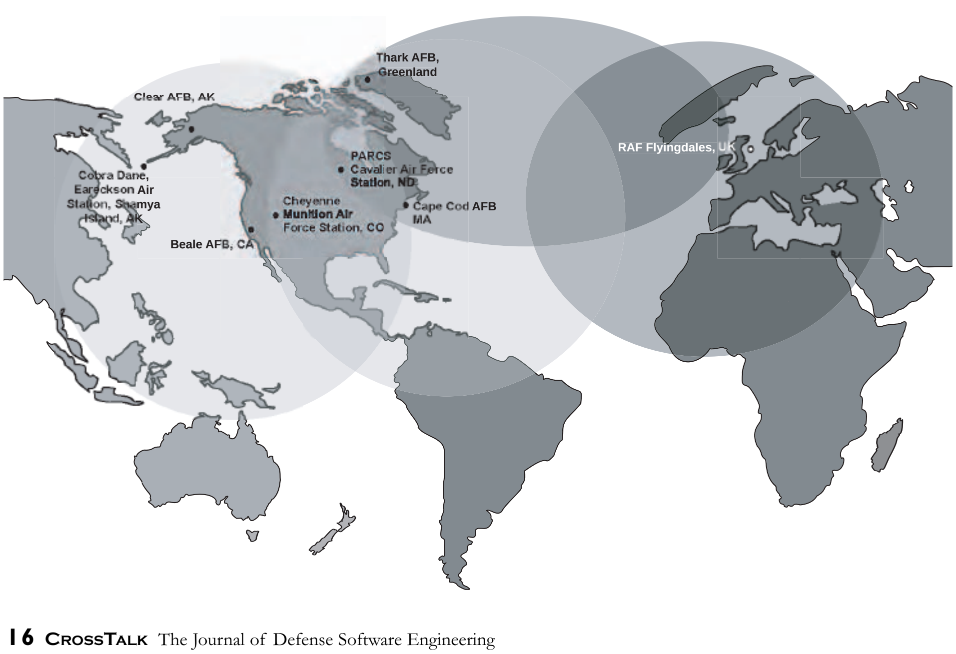
PAVE PAWS to BMEWS Site II

At height of the cold war, the Air Force built six extremely large phased array radar systems (see Figure 1, page 16) [5]. One of the systems known as PAVE (PAVE PAWS) initially had large phased array radars located in four locations in the continental United States, the first of which was located at Otis Air Force Base on Cape Cod Massachusetts. The design of the Cape Cod system had a dual phased array radar face providing radar coverage eastward over the Atlantic Ocean. In parallel, the same operational command also needed to upgrade the existing BMEWS that was located in three places: Thule Greenland; Fylingdales, United Kingdom; and Clear Air Force Base, Alaska. The PAVE PAWS mission was to provide warning of a Sea-Launched Ballistic Missile (SLBM) attack against the United States, while the BMEWS system was focused on Inter-Continental Ballistic Missile attack against North America and United Kingdom. The original BMEWS system built in the early 1960s really needed to be refurbished and upgraded in the 1980s and 1990s. However, since the operational command stated its mission needs in an operational context rather than specific design and requirements, the contractor building PAVE PAWS phased array radars offered to upgrade the current BMEWS by replacing them with a PAVE PAWS-like, two-faced, phased array radar built on the existing BMEWS buildings at the Thule Site with adjusted tracking and reporting software. Known as the BMEWS Radar Upgrade Site I (BMEWS I), the new radar system provided a dramatically enhanced capability. The next BMEWS site upgraded was the Fylingdales, United Kingdom site, but with pyramid-shaped, three-faced phased array radar system. Again with the ConOps based on mission needs, the Fylingdales system could extend the two-faced, phased array system installed at Thule to handle the mission at Fylingdales. To take this concept to the extreme, the Clear AFB, Alaska BMEWS Site II (BMEWS II) was recently upgraded by actually dismantling the PAVE PAWS site located at Warner-Robbins Air Force Base and reinstalling the radar system at the Clear location. Naturally, the processing equipment and software also needed upgrading since the system was going from the SLBM mission to its current mission, but the same level of flexibility in the ConOps allowed for extensive reuse of existing systems for new operational missions.

Summary

The purpose of this article is to give a viewpoint on how extremely important the ConOps document is in the system engineering and development life cycle of the system. Not only is the document important in the beginning of the life cycle, but it needs to be revisited during the major lifecycle phase points of the program to ensure the program remains on track to build the right system. It is also important to make sure it is updated to make sure the ever explosive growth of technology is continually examined to see how technology insertion can be accomplished at the most economical point of the engineering life cycle. While the OTH-B situation is in the extreme, it goes to the point that there are two aspects of the system life cycle: Did the contractor build the system right (all *shalls* passed) and did the contractor build the right system (meets the operational missions)? We saw two significant systems that, by having a mission-oriented ConOps, allowed the contractors to bring creativity, flexibility, and cost-savings reuse of existing systems for new missions, such as PAVE PAWS for BMEWS and the new SEEK SCORE radar bomb scoring system from the old, existing GDBS.◆

Figure 1: *PAVE PAWS and BMEWS Coverage Map*



References

1. IEEE. IEEE Guide for Information Technology – System Definition-Concept of Operations Document. Std. 1362-1998. IEEE Electronic Library.
2. Capability Maturity Model Integrated (CMMI), Systems and Software: V1.1, Software Engineering Institute, 2001.
3. Air Combat Command. "Fact Sheet: Over-the-Horizon Backscatter Radar: East and West." U.S. Air Force Fact Sheet <www.acc.af.mil/factsheets/factsheet _print.asp?fsID3863&page=1>.
4. FAS Military Analysis Network <www.fas.org/man/dod-101/sys/ac/equip/an-tpq-43.htm>.
5. PAVE PAWS. Wikipedia <http://en.wikipedia.org/wiki/PAVE_PAWS>.

Note

1. While many people tried to create an acronym meaning for PAVE, it was never an acronym for anything, it simply meant an Air Force Program.

About the Author

Alan C. Jost (Lt. Col., U.S. Air Force, retired) is a senior software program manager in the Raytheon Northeast Software Engineering Center (SWEC), where he works multiple tasks including the Software Engineering Program Group Executive Committee for SWEC's CMM/CMM Integration Level 5 sustainment and as a process engineer on the AutoTrac III Air Traffic Control Product Line and DD(X) ExComms Software Cross Product Team. Jost joined Raytheon after 20 years in the Air Force. He has served in a variety of line management, process engineering, and software task management roles in SWEC.

Raytheon
MS 3-1-3914
1001 Boston Post RD
Marlborough, MA 01752
Phone: (508) 490-4282
Fax: (508) 490-1366
E-mail: alan_c_jost@raytheon.com