**SPADOC**

Capt Russ Smith
CC4040
22 March 1999

**The System**

The Space Defense Operations Center (SPADOC) version 4C computer system is used in the Space Control Center (SCC) within Cheyenne Mountain Air Station, Colorado. It performs the functions required by Commander in Chief, North American Aerospace Defense Command (NORAD) and United States Space Command for Space Surveillance and Space Defense. SPADOC was developed under the Cheyenne Mountain Upgrade (CMU) acquisition program that began in the early 1980’s. The system maintains the Satellite Catalog (SATCAT) which is a database containing the location of all space objects being monitored by NORAD’s Space Surveillance Network.

The location of each object, whether it is a payload or a piece of space junk, is represented by an element set (ELSET). Currently, the database contains a little less than 9000 ELSETs. The SATCAT is maintained by receiving observations from the surveillance network, then updating the ELSET with the new observations. The updated ELSETs are then supplied to the surveillance network which propagates the objects over the sensor site for acquisition and to provide updated objects back to Cheyenne Mountain.

Most activities related to SATCAT maintenance are done automatically within the cycle described above. There are, however, activities that require manual intervention by an Orbital Analyst (OA) on crew in the SCC. When a new object is launched into space the payload and each associated piece of orbiting debris is cataloged by an OA. Also, if an object maneuvers on orbit, there is a manual process to update the ELSET. Other duties performed by crew-members in the SCC include Laser Clearinghouse (LCH) for laser operators and Collision Avoidance (COLA) for NASA; both functions are intended to protect on-orbit assets. COLA and LCH use the SATCAT to propagate objects over the laser or launch site to determine if there is a possibility of collision.

**Problems with SPADOC**

Since the SPADOC program began before Graphical User Interfaces became commonplace, the user interface is made up of 3 letter menu commands and text boxes that have to be typed in. Some menu commands are 3 or 4 levels deep and rely on the operator to memorize 40 or 50 different 3 letter commands. Executing a command often requires the user to scroll through multiple pages of data input fields. This text based system is a source of frustration for users. Especially since most users have become quite comfortable with the typical Windows interface. Attempts to change the Human Machine Interface of the system is often put at the bottom of the list of changes because of two reasons, the heart of the system is the astrodynamic routines and the requirement for multi-level security.

The scientific variables that are used clutter the input screens but are rarely manipulated by the end user. These variables are important from an astrodynamic standpoint, but not operationally. Multi-level security was required because of the Canadian Air Force members of NORAD. The technology for multi-level security had not been mature enough (and is still not) to meet the requirements of NORAD. The system’s ability to provide multi-level security ahead of its time has been very costly, and created a complex architecture that makes changing the user interface very costly.

The SPADOC system performs the functionality it was designed to perform, but is designed very poorly in terms of its human machine interface. This poor designed has created a lot of work arounds that make learning the system difficult. Attempts have been made to make changes to the system. Unfortunately, making changes to the interface may pay off in the long run, but this creates a near term requirement for training the "old dogs new tricks".

There was a real disregard for human factors in the design of SPADOC. The system development suffers from many years of modifications, where poor design was compounded by the fact that resources were applied to getting the "right answer" instead of making it easy for the operator to get the "right answer".

Under the postulates of human factors, people are adaptable and they can learn to operate a poorly designed machine, like SPADOC, resulting in increased training time, increased stress for the operators, increased human errors under stress and unused machine capabilities. Users have been able to "work around" many of the shortcomings of SPADOC to arrive at the right answer.

**Prevention**

Preventing these shortcomings would come from apportioning development costs and effort toward human factor requirements. Requirements for SPADOC never included human factors. The core operational requirements were related to availability of the system, accuracy of the resultant processing and timeliness of the processing of data. Once the system was fielded, changes have been categorized based on modifying and fixing functionality supporting these core requirements.

Preventing the development of a system that lacks a quality human machine interface requires integrating human factors requirements into system performance requirements. Within the development of a system, human factors could include, according to the Introduction to Human Factors class handout, a review of engineering planning documents to ensure that behavioral factors have been considered. This can be done early in the design but should include at a minimum, a prediction of where significant behavioral impacts and difficulties are.

During the demonstration and validation phase, information flow and workload should be analyzed in order to determine if all data items are available to the system user. Also, during this phase, a Human Factors (HF) portion of the Test and Evaluation Master Plan (TEMP) should be written. By having a Human Factors section in the TEMP, the human factors concerns will be identified up front, before the system goes into full-scale engineering, where it would be too late.

During the full-scale engineering development phase, the human- factors issues should be reviewed and evaluations should be done as the system is developed. Finally, during production and deployment, operational testing should uncover shortcomings in the design of the human machine interface and improvements should be evaluated and recommended.

**Recommended Fixes**

Improving the human-machine interface of SPADOC is difficult because of how tightly coupled the interface is to the rest of the system. The first step is to recognize that training dollars spent now to re-train current users will pay off later in shorter training time for new users. Another recognition needs to be made by the managers of the SPADOC program in going to a graphical user interface with a partitioning of functionality will be easy for new users since so many people are comfortable with windows like interfaces.

The steps necessary to segregate this portion of the system includes re-writing portions of the SPADOC system in a systematic way. It would be necessary to have the old and new systems coexisting side by side. This is the architecture that existed as SPADOC was being developed. SPADOC was the new system and the 427M system was the system being phased out. This will allow human factors to be considered with the new system and evaluated against the old.

The first function to be moved is the Manual Differential Correction (MDC) process (see attached). This function is the heart of the system. By sitting down with users and considering what processing needs to be done, human factors experts could design an interface that is not only easier to use, but allows for quicker execution of tasks. Once the "feel" of the new SPADOC is developed, other functions could be moved over with the same "feel". The feel of the new SPADOC speaks to the uniformity that makes executing functions similar across tasks. This uniformity will also make adding additional functionality easier and quicker for the user.

Since there are a lot of data that is not necessary for common execution, segmenting the common data from the advanced data items would make it much easier to quickly move through the processes. Confusion will also be eliminated by relying on graphics to represent tasks, rather than three letter codes. Eliminating as much typing as possible will speed up the processing of different tasks. If a user is required to select from a list of two or three different items, a pull down menu should be used to select, rather than by typing in the information each time.

As noted earlier, change to the current system is very difficult because the design of the system has grown into a tightly coupled mess that prevents isolation of interface modules from other processing modules. By making the interface segment independent, user interface issues could be easily improved upon relatively cheap. And by allowing a portion of the change budget to be allocated to interface changes, changes would not have to compete with operational fixes or modifications, which often exhaust the entire budget before a change could be incorporated. The new SPADOC could grow out of the old, and could include an easy to use interface that is easy to learn and easy to modify.