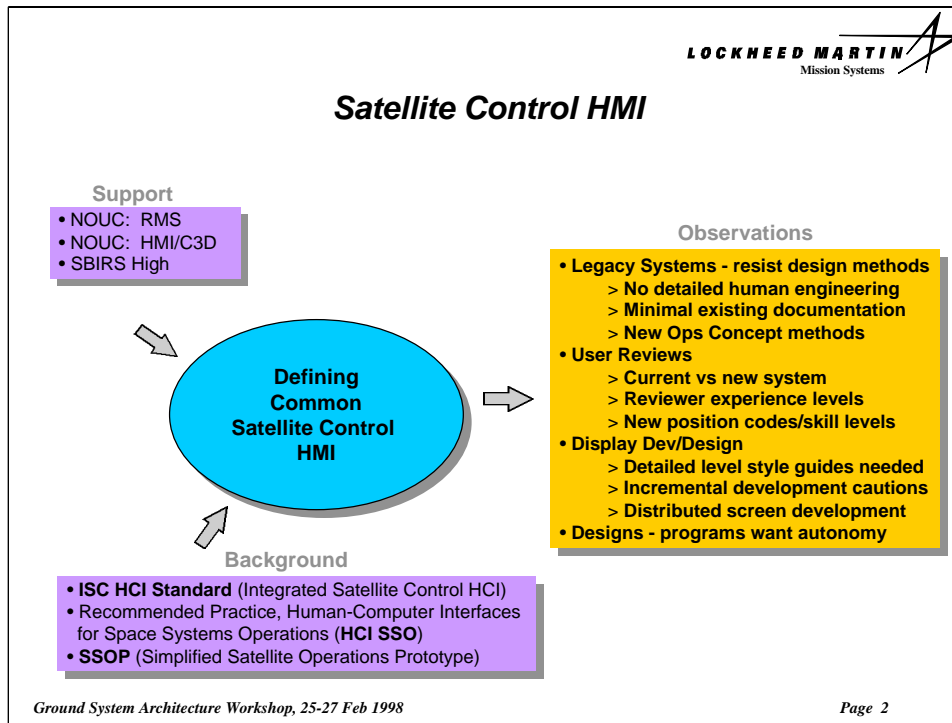


***Experiences in Defining
Common Satellite Control HMI***

**Debbie Hankins
(debbie.hankins@lmco.com)
Lockheed Martin Mission Systems**

Satellite Control HMI



We are currently supporting the NOUC (Network Operations Upgrade Contract) and SBIRS (Space Based Infra Red System) programs. These include RMS (Resource Management Segment), the HMI CCP (standardization efforts sponsored by SMC/AXE & CWG), and the C3D (Commercial Command and Control Demo) CCP.

Background on our previous experience includes work on the predecessor to the HCI SSO, the ISC HCI, as well as work on defining what commonalties could exist across programs in terms of satellite control operations, the SSOP effort.

Observations on legacy systems, user reviews, displays, and design are discussed in the following charts.

Legacy Systems

- **Resistance to methods**
 - No detailed human engineering (e.g functional/workload/task analysis, etc.)
 - Argument..... “not building system from scratch”
 - Typically HMI concepts for differences between old and new are not documented
- **New methodologies for ops concepts need to be considered for new ways to incorporate HMI analysis**

Our experience has been that legacy systems resist typical methods of “detailed” human engineering when updating their programs. This includes functional analysis, workload analysis and task analysis. At best, these methods are not in the budget to the same degree as new programs with brand new concepts.

We believe this is largely due to the notion that a legacy system is known and is not being “built from scratch”. This may be true; however, we find that minimal documentation is available to adequately describe the program as it has evolved. This is particularly true of older programs which have undergone years of change. Even if the original analysis and design documentation were available they would likely be outdated.

Another observation related to methodologies in this area is the fact that there are new methods for describing and documenting Ops Concepts (e.g. Use Cases). These new techniques have not been fully mapped into methods for functional/task analysis and HMI design. This is an area which lends itself to more investigation and would result in better overall systems development, giving us the best bang for the buck.

User Reviews

- **Typically focused on current versus proposed new system**
- **Very different feedback from experienced/current operator vs new users**
- **Position codes and skill levels change and must be considered**

One common aspect of user reviews which we have to account for is that more often than not, the users/operators reviewing our designs are current operators and respond to designs based on the current system rather than the proposed new system. This requires that we take extra care in designing our questions around the proposed system's functionality as well as in providing clarification to operators on the new system concepts.

Another variable is that we typically have operators of varying levels of experience. This requires us to be cautious that we understand the source of the feedback and make sure it maps/is relevant to the proposed system functionality. Also note that we typically don't get the same set of operators to comment on screen designs and therefore have many opportunities to review concepts with them.

Lastly, no one really has a good mapping for position codes into these newly automated and sometimes brand new operator tasks. As roles are combined and redefined, we are asked to verify and assess skill levels. To do this we need good task analysis of existing positions and skill levels. These don't exist, so we attempt to assess the old position(s) and bridge to the the new.

Displays (Dev/Design)

- **Geographically distributed development groups require “detailed” level style guides**
- **Style guides typically at too high a level**
- **Incremental development if not carefully applied can result in screen development which is isolated from main thread scenarios**
- **Distributed screen development is not effective**

We have several observations regarding the actual development and design of HMI. First, we find that with larger teams, geographically separated and/or working in parallel, we need more detailed level style guides. For example what we have on SBIRS describes the common constructs (buttons, lists with scroll capability etc.), but it doesn't include placement information or behaviors. Also we find that as development continues, issues come up which require revision of the style guide. This argues for detailed screen definition and documentation early on in the development process, before functional code is impacted.

The next observation is that incremental development can result in designs which don't flow when taken in context with main thread scenarios and can cause schedule impacts and/or inadequate screen design for the overall ops concept.

Lastly, distributing responsibility for screen design across the development organization to each owning application can cause problems in overall consistency of conventions. Different levels of user involvement/feedback can result in conflicting guidance which is not realized by each owner. One organization within the program should have oversight and decision authority for conventions, standardization, and participate in user reviews.

Designs (Commonality/Variations)

- **Each program wants its own HCI**
- **Resistance to commonality due to programmatics**
- **Mission needs vary**

Our final observation is that each program wants its own HMI. This is typically due to the COTS tools which have been selected and are limited in how they can be modified for standardization. It is also due to use of legacy code which is usually not re-written to accommodate screen design. And then the obvious reason is that missions vary and screen conventions which work for one program can conflict with the conventions of another, particularly when the goal is to have an integrated look-and-feel across a particular system.

Summary of Observations

- **Legacy Systems - Resist design methods**
 - > **No detailed human engineering**
 - > **Minimal existing documentation**
 - > **New Ops Concept methods**
- **User Reviews**
 - > **Current vs new system**
 - > **Reviewer experience levels**
 - > **New position codes/skill levels**
- **Display Dev/Design**
 - > **Detailed level style guides needed**
 - > **Incremental development cautions**
 - > **Distributed screen development**
- **Designs - Programs want autonomy**

We offer these observations for panel discussion and hope to get some insight into the similarities and/or differences as experienced by the rest of the panel in these areas. In general we believe that if the community is aware of these observations we, as a team/community, can better plan for them in the future.