Function Allocation Roundtable Report
September 23, 1998

Prepared by:
Laura Militello
Klein Associates
laura@klein-inc.com

Introduction

The day-long function allocation roundtable was held June 1, 1998 at the Airlie Center in Warrenton, VA. The purpose of the meeting was to begin to articulate boundary conditions for the function allocation process. One objective was to determine what is reasonable to expect from function allocation. Another objective was to discuss what tools are available for function allocation.

Three consultants with considerable experience with function allocation, automation, system design, and other design issues attended the roundtable: John Lee, Kim Vicente, and Chris Wickens. In addition, representatives from the Affordability project were present. Attendees included: Jan Cannon-Bowers, Kevin Bracken, Gwendolyn Campbell, Marvin Cohen, Harry Crisp, Floyd Glen, Tim Huddleston, Rob Hutton, Roger Keating, Gary Klein, John LaChance, Ron Laughery, Jean MacMillan, Tom Malone, Jennifer McKneely, Laura Militello, Amie Perry, and Daniel Serfaty. Attendence was purposely held to a small group to encourage interaction and discussion.

The meeting opened with an introduction from Jan Cannon-Bowers that laid out the primary challenges that the Affordability project faces. Amie Perry and John LaChance provided more specific, detailed information regarding function allocation issues in the context of the air dominance scenario. Next, the meeting was turned over to the consultants, each of whom provided a brief presentation of his perspective regarding function allocation. Following the consultant presentations, the discussion was opened to the larger group. The meeting closed with final comments from each of the consultants.

The body of this report contains the potential recommendations derived from the meeting, a synopsis of each consultant's presentation, issues that were discussed but unresolved, and closing remarks from the consultants.

Potential Recommendations

Two overarching conclusions were reached. The first is that proactive, model-based analyses are not sufficient to design a system. An iterative approach using simulations and high fidelity test-beds is needed in order to effectively visualize and anticipate actual system use and function. A second conclusion is that more meetings and workshops are needed to further explore many of the issues raised in this meeting and to settle on a process for the design of a revolutionary new ship. More specific recommendations are listed below.

1. Need a human factors facilitator in the design process
2. Need iterative design process utilizing simulations (models, mockups, and prototypes)
3. Need to include human factors issues, allocation, and human computer-interface principles in the operational test and evaluation
4. Need to delineate a design process including models and man-in-the-loop simulations
5. Need to distinguish between proactive function allocation using models, and reactive function allocation conducted iteratively using simulations and test beds

Consultant Presentations

John Lee: How does technology change the way people do their jobs?

Reactive versus proactive approaches. In the corporate world, a reactive approach to manning reduction is used. A new organization scheme will be implemented, and if people moan enough, a redesign or reallocation of tasks and people occurs. This may not be a bad approach. Perhaps we should not expect to be able to field a perfect system the first time around with a model-based approach. Maybe the most we can hope to do is to field something reasonable and then fix the problems.

Long term consequences of reduced manning. In the Coast Guard, technological and economic pressures have been driving reductions in crew size. For example, increased automation has permitted the implementation of a one-person bridge. While this new configuration has been an economical approach to running ships thus far, the reduced redundancy has resulted in a gap in training. It used to be that the redundancy in the crew afforded on-the-job training and mentoring relationships. This is no longer true because there is only one person on the bridge.

Short term consequences. The Coast Guard has also found increased fatigue in the engineering staff. Although much of the system-monitoring task has been automated, reducing the number of engineers needed, the engineers are "on call" an increased amount of time. The Coast Guard has found that simply being "on call" (even if not called), results in disrupted sleep cycles.

In addition, the decrease in redundancy results in fewer error checks. How do you provide error checks if everyone is occupied all the time?

A third potential consequences concerns workload peaks. How do you deal with peaks of work that are not supported by automation (e.g., power failures)? Automation does not have control mechanisms for every situation.

Defining changing functions. Simply by changing the configuration of people and machines in the work space, the functions themselves are changed. We, as a human factors community, should focus on predicting those changes rather than trying to name and allocate functions. It is important to distinguish allocation from change.

Function is an arbitrary level of description. Determining the appropriate level of resolution is not straightforward. In addition, much technology cuts across functions. Multiple functions are affected simultaneously. For example, when automated GPS-based navigation systems are introduced, is this a reallocation or a change in the nature of the task/function? Furthermore, function may change over time in a way that is not captured with function allocation.

Trust in automation. Lee described an example involving a cruise ship. The Royal Majesty was a cruise ship with a high level of automation in navigation. For example, there was no wheel to steer the ship. In most situations, the crew knew the limitations of the automated system (maneuvering close to shore) and developed manual workarounds to maneuver when needed.

One day, however, the GPS cable on the Royal Majesty broke. A cable had worn through so the information on the display the humans monitored was incorrect. Essentially, a “blindfold” had come down on their automated navigation capabilities, but they got very little direct information that indicated the system was no longer receiving data (a small light in a hard-to-see spot). The ship drifted 15 miles off course and ran aground. No one realized they were off course, even though there were multiple cues that their course was incorrect. The lookouts reported changes in water color (a definitive indicator that they were moving into shallower water). A fisherman hailed them and told them they were lost. The lookouts expected to see buoys that would confirm they were on the right course. These buoys were never seen. Clear indicators that the ship was off course were ignored because the level of trust in the GPS was so high.

How do you “manipulate” trust in a system so that it is appropriate? How much trust should people have in the system? How much cross check/redundancy should be built into the system? Does this change in specific situations? Appropriate use of automation depends on its level of capability. Misuse occurs when the operator’s trust exceeds the capabilities of the automated system. (Inappropriate) disuse occurs when operators don’t rely on automation when they should.

The operator’s trust in the system is a major factor in the effective use of automation that is not considered in a simple function allocation (human vs. machine) approach. The operator needs to know how much faith to put in the system.

Zone of disappointment. There may be a need to recalibrate expectations about function allocation. Often, claims about being able to allocate functions early in the design phase lead to impossible expectations (i.e., that the first fielded version of a system, ship, etc. will be the finished version). In actuality, much of what human factors can offer is reactive. Much of the calibration of the operator to the system and the resolution of conflicts can only be done reactively, after the system has been fielded and tested.

We need to accept that there is a need for a test and evaluation phase in which we can conduct empirical investigations of the efficacy of the system, and make refinements. This needs to be recognized as a design step. There is a need for simulations, reconfigurable displays, and trial runs.

Kim Vicente: Three-way solution to problem requirements

Function allocation is an output of design. Function allocation is a design solution to problem requirements. However, one must first analyze the problem before developing solutions. It is important to distinguish inputs from outputs in the design process.

Focus on problem requirements. There is a need for a systematic and design-independent way of identifying problem requirements rather than assuming the answers up front. It is not a matter of function allocation (that's output) -- you need to specify the space.

You must first identify the problem requirements independent of the design of the interface of a decision support system. What are the hard constraints (vs. soft constraints)? Constraints might include budget, technology limitations, manning requirements, etc.

There are three agents, the human, the machine, and the *designer,* that must be taken into account in specifying the problem space. The designer must formalize and make explicit the design constraints or boundary conditions. These include rationale, assumptions, intentions, and performance criteria. The goal is to design the system based on constraints rather than events. With an event-based approach, one can never predict all possible events, so there is always the possibility that the system will not be prepared to deal with an unanticipated event. If x gets out of balance, the system needs to be able to take it into account. The system doesn't need to know what even caused x to get out of balance.

Discussion around this issue addresses the following two points: 1.) In the typical piece-by-piece design, the designer's intent(s) is not reconciled into a single concept for the operator. 2.) Making the designer's intentions explicit does not prevent errors or problems. It does not tell you to do it one way or the other.

Allocation equals design. Defining the problem space will aid in representing the problem. It is then up to the designer to allocate functions. Function allocation is clearly part of the design process, but can only be done adequately if one first understands the problem space.

Cognitive Work Analysis (CWA) as a tool for representing the problem space. One method that utilizes this approach is CWA. In CWA, the goal is to present the design intention either implicitly or explicitly in the system/interface through training, doctrine, procedures, or hypertext help. CWA is currently being used in the development of a Japanese power plant.

Designers need systematic and design-independent ways of identifying problem requirements rather than assuming answers up front (e.g., Does it make sense to say up front that there will 95 people on the ship?).

Chris Wickens

The operator needs to know more about the uncertainty and unreliability of information propagated by the automated system.

Human information processing and automation. There are some things we know about human information processing. For example, humans can focus attention flexibly (with a cost in the form of accessing information); there are limits to working memory (modified by expertise); humans can build expectancies based on experience and prepare for events; humans are not consistent.

There are also some things we know about automation. Automated systems are constant over time, are reliable, and can flexibly contrast multiple options and states.

These characteristics of human information processing and automated systems impact human-automation interactions. Issues include proximity compatibility (human must be able to access information in order to attend to it), situation awareness (including attention allocation, understanding, and prediction), feedback to both the human and the automation, navigation, trust (calibrating an appropriate level of trust in the automation in different contexts), and transfer (slips).

Information such as the state of the world, sensors, and automation impacts the human's selective attention and drives an integrated belief. The integrated belief impacts the human's level of trust in the system, the resource allocation, and choice selection.

Air traffic control automation. Wickens recently served on the National Research Council Panel on Human Factors of Air Traffic Control Automation. The committee made recommendations about levels of automation, failure recovery/emergency response to system failure, free flight, and introducing human-centered automation in air traffic control.

Levels of autonomy between human and machine. A single scale for determining the level of autonomy between the human and the machine doesn't work. The distribution varies depending on the kind of cognitive work required of the operator.

For example, if information is highly reliable, it is possible to automate much of the information integration function. In the case of decision tasks, if the decision is risky it makes more sense to have a low level of automation and leave the decision making up to the human. If the human plays a large role in making the decision, it is possible to highly automate the response execution. On the other hand, if the decision is not risky and the decision-making process is automated, it may make sense to have the human have a larger role in response execution so that the human retains the final word on how to act. However, it is not a good idea to implement automated execution of automated choices.

Cost-benefit tradeoffs. One must consider the cost-benefit tradeoffs of automation. If one considers information integration tasks and the different levels of automation possible, a low level of automation might provide information extrapolation to the human user to aid in diagnosis, inference, and prediction. A mid level of automation might include display integration, in that the display is configured so that the information is presented in an integrated representation. A high level of automation might include attention filtering devices such as highlighting or cuing.

While display integration seems to have a relatively low costs, research shows differing findings regarding how automation in this case affects attention and inference on the part of the human. For information integration, you need to look at the costs if the process is not reliable. Information integration includes attention filtering and cueing, display of integrated information, and extrapolation, prediction, and inference. In general, filtering and configuration can be done reliably. On the other hand, automated calling of the operator's attention to information can be incorrect, and errors can have high costs (i.e., overtrust in automation).

Why automate? The desired benefits are better decisions and lower workload. The costs can include complacency, lower detection of failures, and lower awareness of state -- as well as lower manual skills, leading to serious consequences if there are automation failures.

An issue is how to deal with low-probability events. How do we design to support the unexpected? Redundancy in the air traffic control center is believed to be one of the reasons why there are so few mid-air collisions (air traffic control personnel watch out for mistake-prone situations and offload each other). Safety versus efficiency must be considered. What is the tradeoff between safety and efficiency? It may be necessary to sacrifice some efficiency in the routine to support extreme situations.

Unresolved Issues

The following section contains important points that were raised during the general discussion. No consensus was reached on these issues. They were identified as important points for further discussion.

Tools for designers

The existing tools designers have to rely on are instinct, lessons learned, and heuristics from predecessors. Design tools should support and improve this toolbox.

We need to provide some generic models to rule out some bad and stupid designs. These generic models exist to some extent, but there is a need to use more simulation capabilities. You cannot build prototypes or mock-ups to deal with all permutations, so you need (engineering) models.

How can you represent the task so that the designer knows how to approach allocation? Can we present guidelines to representing the mission/functions/tasks?

We need to provide designers with the constraints and boundaries of the system. We need an integration of systems, not piecemeal design and development, so that systems work together and design interfaces are common and consistent across systems.

In order to develop guidelines for designers, we need to understand design process and what kinds of guidelines help them. We should also look at the acquisition process, which is distributed across multiple contractors for whom integration is not an objective.

We may need to accept the plurality, creativity, and insights that drive the design process. Perhaps the best we can do is articulate the heuristics we use to convey ideas to designers.

Iterations and simulations

We cannot expect design based on a priori assumptions. We need iterative design with simulations of best a priori designs.

How do we create simulations at the organizational or team level? How can we recreate the team dynamics (cross-functional issues)? We have traditionally simulated only individual functional systems working within their own function.

Realistic test and evaluation

One challenge of the Affordability project is to find a way to conduct realistic test and evaluation. It will not be as simple as it was with the Smartship project where an existing ship was retrofitted and realistic exercises and missions could be conducted. Because of the revolutionary design intended for DD21, it will be challenging to develop a realistic testbed for test and evaluation. After the DD21 has been built it will be important to consider the initial deployment to be part of the test and evaluation phase.

What are the correct questions regarding allocation?

What functions can be automated? Is it possible to automate the task of inferring intent based on doctrine and ROE? Does it make sense to try to automate communication?

Does it make sense to design for the worst case scenario? Damage control is an important issue. Ships are not disposable.

Can we automate and to what extent? Should we automate? Who should do the function allocation? Human factors professionals, systems engineers, the operational person?

Isn't function allocation synonymous with design? Let's just drop the term and couch it as design.

Carlow has a functional analysis based on the mission function hierarchy — can we use this as a basis for characterizing the functions and begin to allocate these?

Allocation strategies assume a correct generation of functions. Are functions abstracted appropriately, at the right level of granularity, taking into account all relevant situations?

What is a function? When does a function become a task? How does one separate functions? For example, navigation is meaningless without propulsion. Are they one task? Are they separated in some situations?

How do you know when you've defined the functions to an appropriate level for allocation? Currently that is part of the creative process.

When should we allow automation of a task/function? When not? Context sensitivity may be an important issue.

Project goals

Isn’t the key issue reducing cost? What if we could reduce cost without changing manning? Same goal different perspective. Manning costs are tightly coupled to training costs. We could address it either way, but we need to address the system designers’ needs. What impact does technological change have on the manning and training costs?

Closing Remarks from Consultants

Chris Wickens

The focus should not be on function allocation per se. The system is not done until its been fielded, tested, and re-iterated. Failures are good things in terms of lessons learned. Contractors are not done when they make the first delivery. A system is not a failure if it doesn’t work right the first time.

A framework is needed for determining where to put limited resources for more or less automation. Guidelines are needed that consider signal reliability and the risk associated with choices (as discussed earlier). It is important to signal or somehow convey the reliability of information and automation. Automation is more feasible at the information/data level than at the decision making or situation assessment level. Response choice automation depends on certainty in the decision making and situation assessment phase.

It is important to use lessons learned from past cases of automation implementation, including failures and successes. Why have there been successes and failures? How does this apply to DD-21? Look at how function allocation was accomplished in previous efforts, not just the result.

John Lee

It is important to combine both reactive and proactive approaches. Models can be used to predict and develop a reasonable design. Prototype field testing can be used to iterate and refine the design.

At the modeling stage, it is important to develop different models. For example, it will be important to model the automation of short term individual workstations to examine specific tasks or functions. It will also be important to examine things at the macro level. What are the effects of operating with 95 people over time (e.g., training, maintenance)? It will be important to link up these models.

We will have to address things we cannot model using prototypes, mock-ups, and field testing.

Kim Vicente

What do we mean by the word "function?" Is it an attribute of the controller or the thing being controlled?

Vicente proposed a research project in which several groups (made up of representatives from several areas including human factors, system engineers, and the operational community) are given the same subset of the problem and asked to solve the design problem. It would be interesting to see what they come up with in terms of function allocation. We would learn what the different approaches generate and the language they use might provide insight into the process of function allocation.

Design is a creative process. You would expect the solutions from each group asked to solve the design problem to be different. However, they may all be functionally equivalent and equally good.