Human-Automation Collaboration in Complex Multivariate Resource Allocation Decision Support Systems

M.L. Cummings[*]
Sylvain Bruni
Humans and Automation Laboratory
77 Massachusetts Ave, 33-311
Massachusetts Institute of Technology
Cambridge, MA 02139
617-252-1512
617-253-4196 (fax)
missyc@mit.edu

Abstract: In resource allocation problems for systems with moving planning horizons and significant uncertainty, typical of supervisory control environments, it is critical that some balance of human-automation collaboration be achieved. These systems typically require leveraging the computational power of automation, as well as the experience and judgment of human decision makers. Human-automation collaboration can occur through degrees of collaboration from automation-centric to human-centric, and such collaboration is inherently distinct from previously-discussed levels of automation. In the context of a command and control mission planning task, we show that across a number of metrics, there is no clear dominant human-automation collaboration scheme for resource allocation problems using three distinct instantiations of human-automation collaboration. Rather, the ultimate selection for the best resource allocation decision support system will depend on a cost-benefit approach that could include mitigation of workload, conformance to intended design characteristics, as well as the need to maximize overall mission performance.

Keywords: role allocation, decision support, resource allocation, satisficing, mission planning

[*] Corresponding Author

INTRODUCTION

Resource allocation problems typically involve large data sets including multiple and interrelated variables, conflicting trade spaces, and often significant uncertainty for problems in dynamic environments. Human cognitive limitations such as limited speed in computation [13] and the tendency towards biased decision making in the presence of uncertainty [33] suggest that some form of automation-based decision support should be used to solve such complex problems, particularly in time-pressured environments. However, automation is inherently brittle in that underlying models cannot account for all potential external conditions or relevant variables which could result in erroneous or misleading solutions [32]. In systems with moving planning horizons and significant uncertainty, especially with dynamic exogenous events such as those in command and control environments, it is critical that some balance of human-automation collaboration be achieved in order to leverage the computational power of automation, as well as the experience and judgment of human knowledge-based reasoning [15; 23; 30].

While attempting to generate an automated optimal solution in a resource allocation problem is relatively straightforward, even if theoretically an algorithm can guarantee an optimal solution quickly (which may not always be the case as problem complexity grows), such optimal solutions assume that all variables and constraints are identified *a priori*. A significant problem in the development of optimal resource allocation algorithms in command and control settings is that not all variables and constraints are known in advance. Moreover, since the situation is highly dynamic and uncertain, those algorithmic rules codified *a priori* could be invalidated on a day-to-day or even hour-to-hour basis. Thus, such "optimal" algorithms have inherent limitations. Unfortunately, it is also extremely difficult for humans to solve such problems, primarily due to the information processing limitations discussed above. As a result, they often resort to heuristic-based reasoning, which can be powerful [14] but can also equally lead to disastrous consequences [27]. However, humans are very effective in identifying patterns and applying judgment and knowledge-based reasoning [25] to ill-defined problems, thus filling in some of the information gaps that automation cannot recognize. Moreover, the use of “satisficing” or accepting a “good enough” solution [31] is key in dynamic, uncertain, and high risk settings like commands and control with no true optimal solution. Thus, for resource allocation command and control problems with significant uncertainty, we propose that collaborative decision support is needed that allows operators the ability to leverage the strengths of fast and comprehensive computations, but still allows them the ability to modify and interact with system parameters and respond to dynamic situations in order to find a satisficing solution.

For complex resource allocation problems, operators can be aided by intelligent automation in four distinct information-processing phases, which include data acquisition, data analysis, decision-making, and action-taking [21]. While computer-based data acquisition is a strength of automation (including sorting and filtering of data), especially in time-pressured environments, it is not clear to what degree automation-generated solutions actually “help” human decision makers in the cognitively demanding the data-analysis and decision-making phases. Automation-generated recommendations and solutions often cause mode confusion, a loss of situation awareness, and automation-bias [8; 20]. These issues are especially problematic when the automation presents sub-optimal or even erroneous solutions.

Recognizing then that operators need to leverage automation for complex resource allocation problems, but that too much automation can negate positive benefit from its introduction, we propose that including some form of human-automation collaboration in the form of sensitivity analysis tools can improve both operator and system performance. For our purposes, “collaboration” is defined as the “mutual engagement of agents, such as a human or a computer, in a coordinated and synchronous effort to solve a problem based on a shared conception of it” (adapted from [11], [28] and [12]).

In order to test our hypothesis that providing a decision support environment that provides human operators the ability to conduct sensitivity analyses in collaboration with the automation will improve performance, we developed a test bed for a complex, resource allocation problem that included varying degrees of human-automation collaboration. In the following sections, we detail the resource allocation scenario, the development and testing of the collaboration interfaces, and the results and implications of the findings.

BACKGROUND

Our representative scenario is that of mission planning for Tomahawk Land Attack Missiles (TLAM), which in reality requires a human operator to solve a complex, multivariate resource allocation problem, much like that found in air traffic control settings, the commercial trucking industry, and first-responder resource allocation. In the TLAM environment, human planners must pair missiles, which can come from different launchers (ships or submarines) with preplanned missions, including targets and routes to targets. These mission planners must consider many pieces of information such as different warhead and sensor requirements, location and priority of targets, terrain contour matching data, global positioning system (GPS) data, digital scene matching area correlation (DSMAC) data, etc. The ultimate goal of the mission planner is to develop a list of missile-mission pairings that best meets the overall objectives of higher-level commanders (e.g., such as striking as many terrorist training camps as possible within a specified period of time, while minimizing possible civilian casualties).

While current TLAM mission planners have some very basic computer-based decision support, the planners typically must use either pencil and paper or their memory to keep track of the different factors to include in the assignment task [10]. Such a process is time and resource consuming, and also gives operators no feedback as to the degree of optimality for their plans. Moreover, this mission planning typically occurs under moderate time pressure (1-2 days prior to a launch), often with dynamic environmental constraints (i.e., changing rules of engagement), so the entire mission planning process contains a high degree of uncertainty. Moreover, such strikes can include more than a hundred missions and missiles so the problem space is quite large.

This resource allocation problem features three types of information. The first type, “hard constraints”, refers to those pieces of information that require exact matching between the resource and the objective. For example, if a mission can only be performed with GPS data, only missiles with GPS equipment can be assigned to that mission. The second type, “probabilistic information”, refers to those pieces of information that embed uncertainty. In our mission-missile assignment problem, this is exemplified by the hypothetical notion of “firing rate” which we describe as the likelihood of successful launch of a missile from its launcher. The last category, “optimization information”, represents variables in a pre-existing algorithm that a planner can change in order to conduct a sensitivity analysis. In our case, the example “days to port” represents an actual critical operational variable, in that the U.S. Navy wishes to take as few missiles as possible into the next port for security reasons. Thus, those ships and submarines that are due into port the soonest should use their missiles first, but this is a variable that can be negotiated/optimized. The last two information types, “optimization” and “probabilistic” information, are considered soft constraints, as they involve trade-off analyses.

Given these different information types, the question is then how to design the algorithms and automation, as well as the accompanying decision support interface, including visualizations, in order to promote collaboration in the solution generation process. We propose that collaboration for solution generation of resource allocation problems can be defined across 5 levels (Table 1) [9]. Given the previous discussion of the limitations of humans and automation (i.e., neither are exclusively suited for this task), we propose that collaboration for this solution generation task should fall somewhere across the three middle levels of collaboration (1, 0, -1). The next section details how the different information types were integrated into three different collaborative interfaces, and were then tested in human-in-the-loop experiments.

Table 1: Levels of Collaboration (LOC)

Level	Degree of Collaboration
2	All Human
1	Mixed, but more human
0	Equally Shared
-1	Mixed, but more automation
-2	All Automation

THE DECISION SUPPORT INTERFACES

In order to determine if and how human-automation collaboration could improve operator performance in resource allocation problems, a decision-support system for missile strike planning was developed, StrikeView, which enables planners to develop TLAM mission plans. An important overarching design consideration is that the mission planners understand the domain very well, but are not computer scientists or engineers so they should not require any significant technical experience beyond average computer skills.

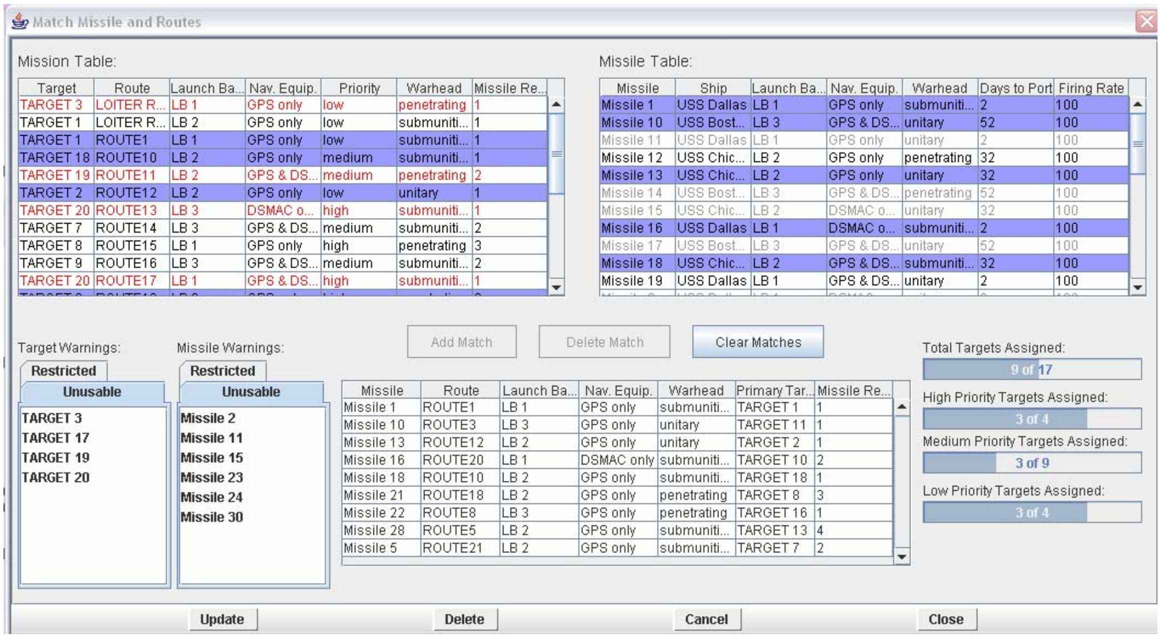
Given pre-defined threat areas, no-fly zones and launch baskets (the geographic location of the ship and submarine launch platforms), planners match candidate missiles to possible missions, which include location, number, and priority of targets, and their corresponding routes. Within StrikeView, three decision support interfaces were created to allow for the matching of the missiles to the missions, at three different levels of collaboration previously described for solution generation.

A. *Interface 1*

Interface 1 (Figure 1) only allows for the manual matching of the resources (the missiles) to the objectives (the missions). The operator selects a mission in the mission table (top left) and matches them to available missions by clicking the button “Add Match”. The mission-missile pair is then added to the match table (bottom center), which displays all matches, constituting the “solution” to the problem. The right table in Fig. 1 displays the primary characteristics of the missiles, such as assigned ship, launch basket, navigation equipment available, and warhead available; as well as those of the missions (left table, Fig. 1), including target, route, navigation equipment required, warhead required, number of missiles required and priority. The priority characteristic corresponds to the priority of the target, which can take the values of high, medium, or low.

Some missions contain loiter patterns, which correspond to the missile hovering over an area of interest (AOI), waiting for an emergent target, which would be considered a high priority target. In case no target emerges, the missile is redirected to a default target, typically a low priority target. Hence, loiter missions are considered of the highest priority despite their assigned, default low priority target. This scenario exemplifies the complexity and ambiguity inherent in such command and control domains.

The bottom left of Interface 1 features two warning tables that display those targets that cannot be reached (no missile is left to fulfill the hard constraints of these targets), and any unused missiles. Finally, Interface 1 includes a summarizing feature at the bottom right, where horizontal bars fill according to the number of targets that have been assigned in the solution.



Mission Table:

Target	Route	Launch Ba...	Nav. Equip.	Priority	Warhead	Missile Re...
TARGET 3	LOITER R...	LB 1	GPS only	low	penetrating	1
TARGET 1	LOITER R...	LB 2	GPS only	low	submuniti...	1
TARGET 1	ROUTE1	LB 1	GPS only	low	submuniti...	1
TARGET 18	ROUTE10	LB 2	GPS only	medium	submuniti...	1
TARGET 19	ROUTE11	LB 2	GPS & DS...	medium	penetrating	2
TARGET 2	ROUTE12	LB 2	GPS only	low	unitary	1
TARGET 20	ROUTE13	LB 3	DSMAC o...	high	submuniti...	1
TARGET 7	ROUTE14	LB 3	GPS & DS...	medium	submuniti...	2
TARGET 8	ROUTE15	LB 1	GPS only	high	penetrating	3
TARGET 9	ROUTE16	LB 3	GPS & DS...	medium	submuniti...	2
TARGET 20	ROUTE17	LB 1	GPS & DS...	high	submuniti...	1

Missile Table:

Missile	Ship	Launch Ba...	Nav. Equip.	Warhead	Days to Port	Firing Rate
Missile 1	USS Dallas	LB 1	GPS only	submuniti...	2	100
Missile 10	USS Bost...	LB 3	GPS & DS...	unitary	52	100
Missile 11	USS Dallas	LB 1	GPS only	unitary	2	100
Missile 12	USS Chic...	LB 2	GPS only	penetrating	32	100
Missile 13	USS Chic...	LB 2	GPS only	unitary	32	100
Missile 14	USS Bost...	LB 3	GPS & DS...	penetrating	52	100
Missile 15	USS Chic...	LB 2	DSMAC o...	unitary	32	100
Missile 16	USS Dallas	LB 1	DSMAC o...	submuniti...	2	100
Missile 17	USS Bost...	LB 3	GPS & DS...	unitary	52	100
Missile 18	USS Chic...	LB 2	GPS & DS...	submuniti...	32	100
Missile 19	USS Dallas	LB 1	GPS & DS...	unitary	2	100

Target Warnings:

TARGET 3
TARGET 17
TARGET 19
TARGET 20

Missile Warnings:

Missile 2
Missile 11
Missile 15
Missile 23
Missile 24
Missile 30

Missile	Route	Launch Ba...	Nav. Equip.	Warhead	Primary Tar...	Missile Re...
Missile 1	ROUTE1	LB 1	GPS only	submuniti...	TARGET 1	1
Missile 10	ROUTE3	LB 3	GPS only	unitary	TARGET 11	1
Missile 13	ROUTE12	LB 2	GPS only	unitary	TARGET 2	1
Missile 16	ROUTE20	LB 1	DSMAC only	submuniti...	TARGET 10	2
Missile 18	ROUTE10	LB 2	GPS only	submuniti...	TARGET 18	1
Missile 21	ROUTE18	LB 2	GPS only	penetrating	TARGET 8	3
Missile 22	ROUTE8	LB 3	GPS only	penetrating	TARGET 16	1
Missile 28	ROUTE5	LB 2	GPS only	submuniti...	TARGET 13	4
Missile 5	ROUTE21	LB 2	GPS only	submuniti...	TARGET 7	2

Total Targets Assigned: 9 of 17

High Priority Targets Assigned: 3 of 4

Medium Priority Targets Assigned: 3 of 9

Low Priority Targets Assigned: 3 of 4

Fig. 1 Interface 1

This summary is presented with an additional breakdown by target priority.

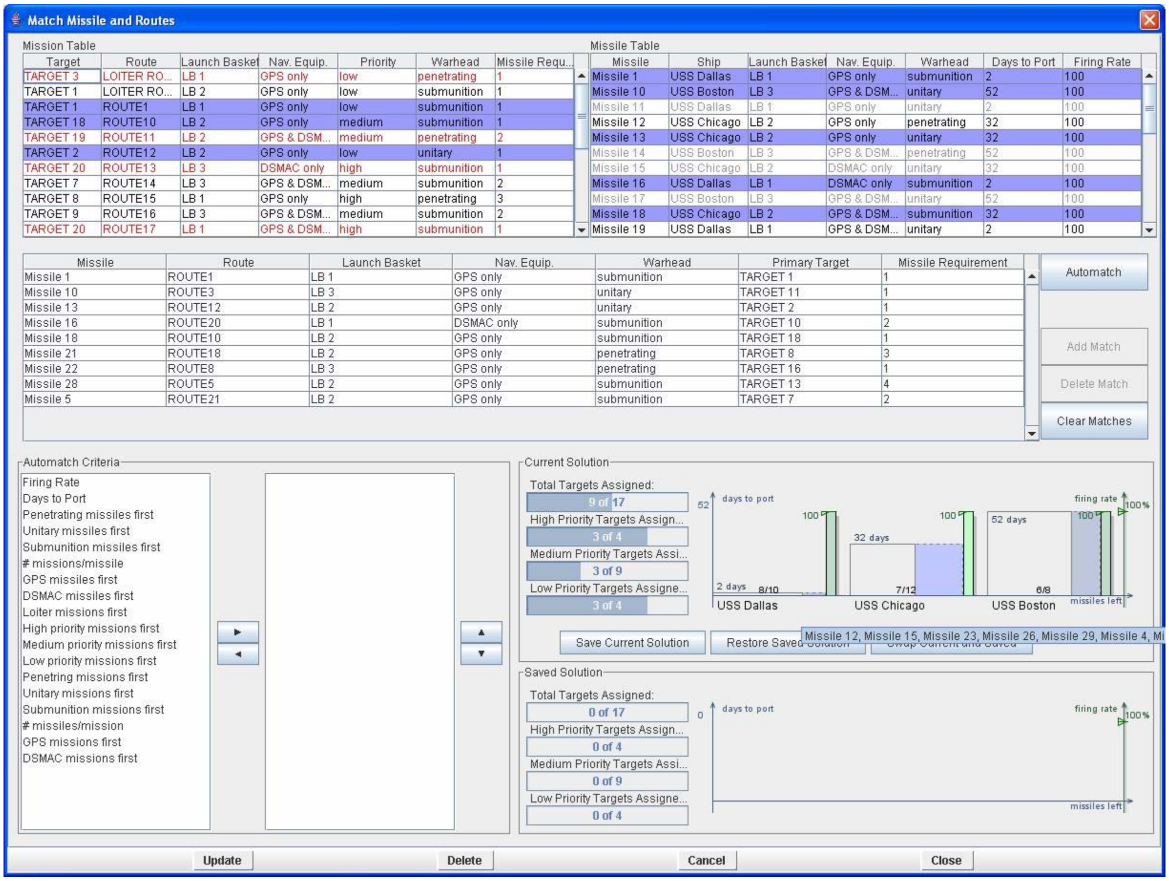
Interface 1 is given the LOC of 1 because the computer only provides basic tools for easier decision and action selection such as filtering, sorting, warning and summarizing features. The human planner is responsible for defining what constitutes locally and globally “good enough” solutions. For example, when a mission (and respectively a missile) is selected in its table, the computer filters the missiles (and the missions) and grays out those which cannot fulfill the hard constraints of the match. So if a mission requires a penetrating warhead, the computer will gray out those missiles which do not carry a penetrating warhead. But the human must decide of all the remaining possible candidates, which constitutes the best match.

B. Interface 2

Interface 2 (Figure 2) allows for more collaborative interaction in the matching of the missiles to the missions, and thus is labeled a LOC of 0, which represents a relatively balanced system. Interface 2 features all the elements of Interface 1, thus providing the human operator with the necessary tools to implement manual matching. However, Interface 2 also features advanced collaborative tools designed to leverage the automation’s computational power. In addition to the mission, missile, the match tables (and their sorting and filtering features), and the summary bars seen in Fig. 1, the main evolution of Interface 2 is the provision of an automated feature called “Automatch”. This automated solution tool includes a customizable heuristic search algorithm. The bottom left of Interface 2 includes a column listing eighteen criteria of potential use when solving this resource allocation problem. These criteria, for example “Match all high priority missions first” or “Use penetrating missiles first”, allow the human to change algorithm variable weights and thus the overall cost function, which allows them more solution flexibility. Operators can select and order (in the adjacent right column) the list of criteria they want to use in order to perform the search to match the missiles to the missions. Once this list is selected and ordered, a simple click on the button “Automatch” will perform an automated heuristic search and subsequent matching based on those criteria.

A heuristic search can be defined as “any search strategy which makes use of heuristics to suggest the best nodes to consider at each stage of the search.” [29]. Heuristics are typically “rules of thumb”, or subjective estimates of the best probable course of action, as defined by the operator. Heuristic search algorithms are generally fast and complete, i.e., if there is a solution, even partial, the search algorithm will find it. However, heuristic search algorithms are hindered by their non-optimality: they can easily get stuck in local optima, hence offering no guarantee that the solution proposed is the best (provided there is an adequate measure of a solution’s quality). In the StrikeView environment, a heuristic-based solution is typically generated by the computer in no more than five seconds.

This implementation of an automated planner allows for efficient use of the computer’s computational power, but under human guidance. The computer’s solution appears in the match table, allowing the operator to manually “tweak” it if deemed necessary. The planner can reject


Match Missile and Routes

Mission Table

Target	Route	Launch Basket	Nav. Equip.	Priority	Warhead	Missile Requ...
TARGET 3	LOITER RO...	LB 1	GPS only	low	penetrating	1
TARGET 1	LOITER RO...	LB 2	GPS only	low	submunition	1
TARGET 1	ROUTE1	LB 1	GPS only	low	submunition	1
TARGET 18	ROUTE10	LB 2	GPS only	medium	submunition	1
TARGET 19	ROUTE11	LB 2	GPS & DSM...	medium	penetrating	2
TARGET 2	ROUTE12	LB 2	GPS only	low	unitary	1
TARGET 20	ROUTE13	LB 3	DSMAC only	high	submunition	1
TARGET 7	ROUTE14	LB 3	GPS & DSM...	medium	submunition	2
TARGET 8	ROUTE15	LB 1	GPS only	high	penetrating	3
TARGET 9	ROUTE16	LB 3	GPS & DSM...	medium	submunition	2
TARGET 20	ROUTE17	LB 1	GPS & DSM...	high	submunition	1

Missile Table

Missile	Ship	Launch Basket	Nav. Equip.	Warhead	Days to Port	Firing Rate
Missile 1	USS Dallas	LB 1	GPS only	submunition	2	100
Missile 10	USS Boston	LB 3	GPS & DSM...	unitary	52	100
Missile 11	USS Dallas	LB 1	GPS only	unitary	2	100
Missile 12	USS Chicago	LB 2	GPS only	penetrating	32	100
Missile 13	USS Chicago	LB 2	GPS only	unitary	32	100
Missile 14	USS Boston	LB 3	GPS & DSM...	penetrating	52	100
Missile 15	USS Chicago	LB 2	DSMAC only	unitary	32	100
Missile 16	USS Dallas	LB 1	DSMAC only	submunition	2	100
Missile 17	USS Boston	LB 3	GPS & DSM...	unitary	52	100
Missile 18	USS Chicago	LB 2	GPS & DSM...	submunition	32	100
Missile 19	USS Dallas	LB 1	GPS & DSM...	unitary	2	100

Missile	Route	Launch Basket	Nav. Equip.	Warhead	Primary Target	Missile Requirement
Missile 1	ROUTE1	LB 1	GPS only	submunition	TARGET 1	1
Missile 10	ROUTE3	LB 3	GPS only	unitary	TARGET 11	1
Missile 13	ROUTE12	LB 2	GPS only	unitary	TARGET 2	1
Missile 16	ROUTE20	LB 1	DSMAC only	submunition	TARGET 10	2
Missile 18	ROUTE10	LB 2	GPS only	submunition	TARGET 18	1
Missile 21	ROUTE18	LB 2	GPS only	penetrating	TARGET 8	3
Missile 22	ROUTE8	LB 3	GPS only	penetrating	TARGET 16	1
Missile 28	ROUTE5	LB 2	GPS only	submunition	TARGET 13	4
Missile 5	ROUTE21	LB 2	GPS only	submunition	TARGET 7	2

Automatch Criteria

Firing Rate
Days to Port
Penetrating missiles first
Unitary missiles first
Submunition missiles first
# missions/missile
GPS missiles first
DSMAC missiles first
Loiter missions first
High priority missions first
Medium priority missions first
Low priority missions first
Penetring missions first
Unitary missions first
Submunition missions first
# missiles/mission
GPS missions first
DSMAC missions first

Current Solution

Total Targets Assigned: 9 of 17
High Priority Targets Assign... 3 of 4
Medium Priority Targets Assi... 3 of 9
Low Priority Targets Assigne... 3 of 4

Missile 12, Missile 15, Missile 23, Missile 26, Missile 29, Missile 4, Mi

Saved Solution

Total Targets Assigned: 0 of 17
High Priority Targets Assign... 0 of 4
Medium Priority Targets Assi... 0 of 9
Low Priority Targets Assigne... 0 of 4

Fig. 2. Interface 2

all solutions, adjust just a few matches, and in general, explore the automation's proposed solution space. In addition, Interface 2 features a visualization of the impact of probabilistic and optimization information (namely, firing rate and days to port) on the used and available resources (Fig. 2, right corner). Finally, Interface 2 includes an option to save the current assignment for a "what-if" comparison. When used, the "Save" option stores the assignment currently displayed in the match table at the bottom right of the interface, only showing the summarizing graphical display. The operator can then compute another solution (manually or using Automatch) and then compare the two graphical solution summaries.

C. Interface 3

As compared to Interfaces 1 and 2, Interface 3 (Fig. 3) does not provide access to the raw data (mission, missiles, matches and their respective characteristics), but only presents aggregated information. The "Automatch" button at the top leverages the same heuristic search algorithm as in Interface 2, however, the user is limited in the number of variables they can manipulate. Instead of modifying the weights of all the variables included in the cost function, in Interface 3, planners can only slightly modify this algorithm. This is accomplished by using the "Cost Function Weighting" sliding bar, which sets the relative importance between the probabilistic information (firing rate) and the optimization information (days to port) in the heuristic search algorithm. Moreover, as will be discussed in further detail below, operators can change missile assignments by priority and warhead type, so operators have access to four variables in Interface 3, as opposed to eighteen in Interface 2.

The other significant difference in Interface 3 (Fig. 3) is the use of a configural display, which relies on a higher-level visual representation of the solution, as opposed to the text-based tables in Interface 1 and Interface 2. Configural displays have been shown to promote faster and more efficient perceptual cognitive processes, as compared to those relying on memory, inference and integration [1; 2; 7; 18; 24]. Configural displays typically include emergent features, or "high-level, global perceptual features that are produced by the interaction among individual parts or graphical elements of a display" [3].

The global perceptual feature of this display is the depiction of the quality of the overall automation-generated solution, displayed in the center with the solid green area. This trapezoidal figure metaphorically represents the assignment solution quality, i.e., the more missions that have been assigned, the more filled in the central area is. A complete solution (all missions assigned) is represented by a full rectangle (up to 100% on all corners).

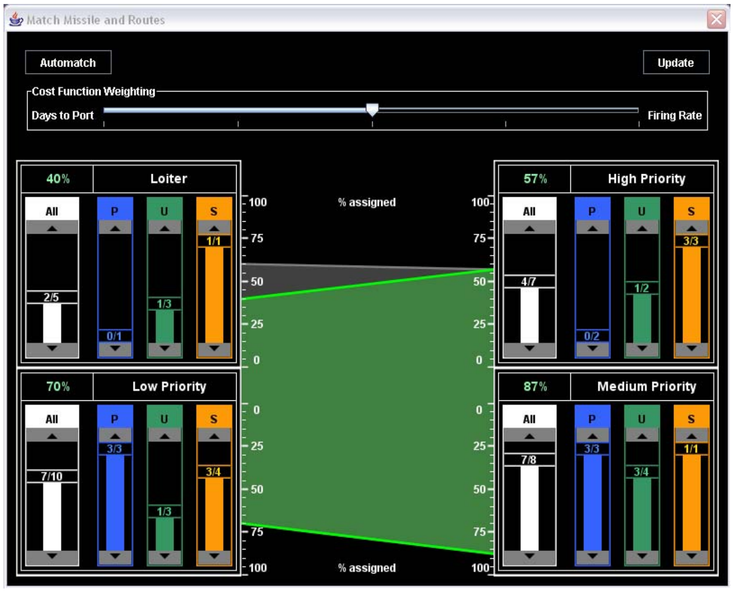


Fig. 3: Interface 3

In addition to manipulating the Cost Function Weighting, planners can also force the algorithm to make assignments by clicking on the up or down arrows in the vertical sliders for the different warhead (penetrating (P), unitary (U), and submunition (S)) and priority types (loiter, high, medium, and low). This forces the computer to find a way to increase or decrease the number of assignments that correspond to that specific slider. For comparison purposes, each time the solution is modified by the user, the new solution is displayed in solid green, while the previous solution is displayed in pale grey in the background, as seen in Fig. 3.

Interface 3 is assigned a LOC of -1 since more of the burden is on the automation to do the computation and solution generation, and only gives the human planner some ability to tailor the search space. It does not have the same level of automation transparency available in Interfaces 1 and 2. However, Interface 3 provides a higher-level visualization that allows the user to more easily compare possible solutions.

II. Human-in-the-Loop Experiment

In order to investigate planners' behavior with these different collaborative decision-support systems, an experiment was conducted. Twenty four subjects took part in the experiment: 13 U.S. Naval officers and 11 reserve officers in training from the Air Force (8) and Army (3). These participants (21 males, 3 females, mean age: 26 yrs ± 4 standard deviation (SD)) had on average 2.6 ± 3.2 yrs military experience. All Naval officers received basic mission planning as part of their standardized Navy schooling, four had extensive experience with TLAM planning (more than 500 hours each), and four had ~100 hours of experience with TLAM mission planning. Eight participants had been involved with live operations or exercises involving the use of Tomahawks.

Participants were randomly assigned to a counterbalanced interface configuration. The three configurations of the StrikeView interfaces that were tested were Interface 1 (I1), Interface 2 (I2), and Interface 3 (I3), as described in the previous section. The test scenario required matching 30 missions with 45 missiles, but due to a lack of required resources (reflective of real world constraints), not all missions could be matched with the required number of missiles. Participants were tested on similar computer configurations: a Dell Dimension 8250 with two Dell 19in flat panels (1280x1024 resolution); or a Stratosphere Titan QX6, with a Samsung 21in flat panel (1280x1024 resolution). The difference in computers used did not impact operator experience as StrikeView requires little computer resources, and thus operated with no latency between the experimental settings.

Each experiment started with a 30-minute training session, which included a short briefing about the TLAM environment and the task, a walkthrough of the interface configuration, and two practice sessions. During the experiment, screen-capture software recorded the screen activity during the entire experiment. Participants were provided with Rules Of Engagement (ROE), describing the three main criteria ordered by priority, to be used to solve the problem (see Table 2, PL = priority level). A 30-minute debriefing session followed the experiment, where one randomly chosen scenario performed was replayed for each participant using the screen capture software. Participants were asked to explain to the experimenter what strategy they used to solve the problem.

Table 2: ROE Priority List

PL1	Prioritize the missions in this order: Loiter (most important), High, Medium, Low (least important).
PL2	Assign at least one missile to each target. (Note that a target may be reached by several missions).
PL3	Maximize firing rate and minimize days–to-port constraints.

RESULTS AND DISCUSSION

A. Performance Score

Overall performance (P) across the three interfaces was evaluated using a weighted objective function of the number of matches accomplished by the operator, with a breakdown by priority (1). The PMs correspond to the Percentages of Matches for the four levels of mission priority: loiter, high, medium and low, in decreasing order of importance. Each PM is computed as the number of missions of each priority that were matched, over the total number of missions of that priority. The weights of the performance function (1) were chosen to reflect the hierarchy of levels of priorities, i.e. the weight is doubled at each level of priority in increasing order. The sum of weights is one, in order to ensure a 0-to-100 range of performance (the same scale as the PMs).

$$P = \frac{8}{15} \cdot PM_{Loiter} + \frac{4}{15} \cdot PM_{High} + \frac{2}{15} \cdot PM_{Medium} + \frac{1}{15} \cdot PM_{Low} \quad (1)$$

The statistical model used was a 3x2 ANOVA with the interface type as the first factor (1, 2, or 3), and experience (novice vs. expert) was the second factor. The experience factor was included since 13 of the subjects were experienced Naval officers, while the rest were officers-in-training, so this difference in general military experience could be a confounding factor. One subject (a novice) was removed from the sample of 24 as an outlier (4.2 standard deviations (SD)), because of lack of adherence to pre-specified rules of engagement. The overall omnibus test was significant for interface type (F(2, 22)= 6.482, p=.008), but was not significant for experience. There was no significant interaction, and Figure 4 illustrates the interface factor level results. Tukey pairwise comparisons between Interfaces 1 and 2 reveal no significant difference, however, the differences between Interfaces 2 and 3 (p=.015) and Interfaces 1 and 3 (p=.014) were significant.

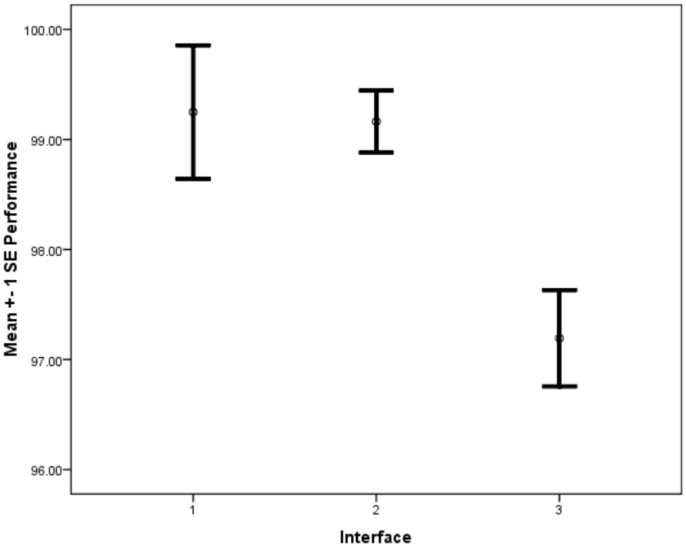


Figure 4: Performance Comparison for Interfaces 1, 2, and 3

It is not surprising that experience level did not have a significant effect since all three interfaces represent a revolutionary technology over current strike planning tools, so while the officers had significantly more actual strike planning experience, the playing field was leveled for all participants because of the introduction of a new tool.

The interface factor results proved interesting in that the two interfaces (1 and 2) that both supported the more traditional spreadsheet approach to mission planning essentially promoted statistically no different performance. However the more graphical configural display, not familiar to participants, promoted the lowest performance scores. While statistically significant, it should be noted that while participants using the graphical Interface 3 performed "worse" than those with the more traditional spreadsheet displays, they were also within 5% of the best solution.

B. Strategies

Given the fact that overall performance scores were very similar across the interfaces, even those for Interface 3, investigating those strategies used across the interfaces is critical for assessing the primary research question, which was how the balance of automation and human decision-making in the solution generation process influenced performance. The primary tool used to investigate these human-automation interaction strategies was the Tracking Resource Allocation of Cognitive Strategies (TRACS) tool that plots plot users' decision making and action selection steps as a function of levels of interface information aggregation [5; 6].

TRACS is a two dimensional representation of strategies exhibited by operators in supervisory control systems. It is a grid representation that maps interface information aggregation (y axis) across information processing stages (x axis). As depicted in Figure 5, the information aggregation axis ranges from data element to criteria set which represents the level of "control" an operator has in guiding embedded automation in supervisory control resource allocation. For example, in Interfaces 1 and 2, an operator can select individual data elements like specific missiles or missions for solution filtering. In Interfaces 2 and 3, they can specify higher-level criteria such as groups of missiles for automation-generated plans, thus allowing the automation to aggregate the necessary lower level information for faster processing (but with less "control" in that operators do assign each individual mission/mission.) The information processing axis represents the basic stages required in supervisory control decision making and action [22], which include requesting information, analysis of this information, the resulting decision, evaluation of this decision, and lastly, integration of a local solution into the overall resource allocation plan.

The TRACS two axis representation forms an operator "state" map which shows the chaining of operator states (e.g., requesting low-level data or making decisions based on high-level aggregated data.) The lines in Figure 5 show transitions between states and the circles show percentages of times in a state. The x and y axes are anchored as seen in Figure 5 so that the lower half of the map represents use of low-level automation, which the upper half represents reliance on automated planners. The left side generally represents information acquisition and analysis (either by the human or the automation), which the right half depicts the use of automation for decision-making and action implementation. Figure 5 represents an operator who

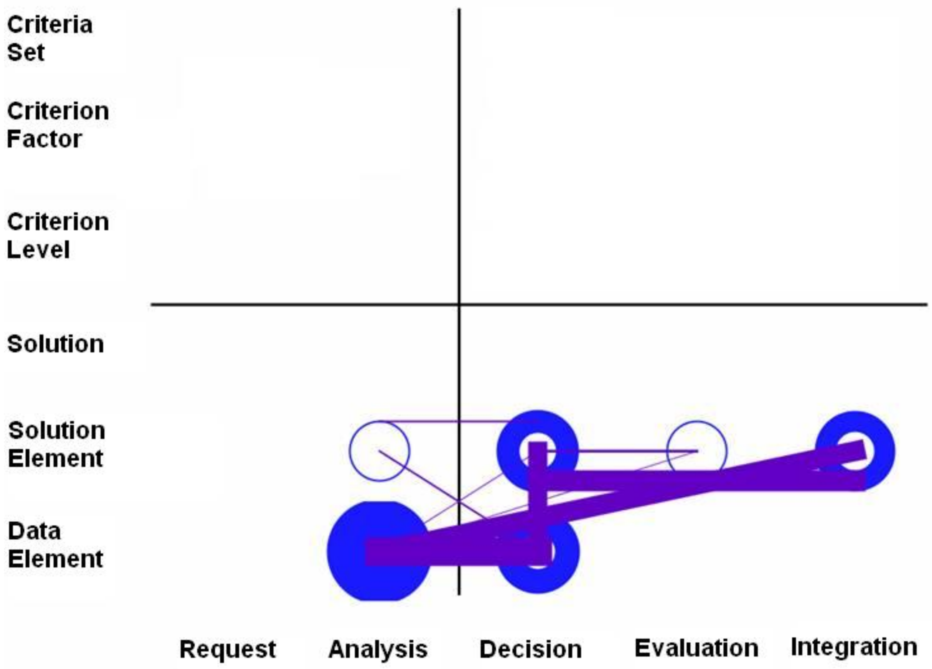


Figure 5: Tracking Resource Allocation Cognitive Strategies (TRACS) Representation

spent most of his time in the analysis of low level data, and did not leverage any automation planning tools. TRACS is not an analytic tool in and of itself, but it is a representation of operator state transitions that is meant to enable analyses of different strategies which may vary, which will be demonstrated in the following sections.

The first variable which examined different strategies used across the three interfaces was the simple count of how many steps a participant exhibited in generating a solution across all the information processing steps. This simple count of steps taken along the TRACS x axis can be seen as a surrogate task load measure. The same 2x2 statistical model described previously was used. The interface factor had a significant effect on the number of steps to a solution ((F(2,22) = 9.122), p = 0.002) while status (novice vs. expert) did not, and there was no significant interaction. As suggested in Figure 6, Tukey comparison tests reveal a significant difference between Interfaces 1 and 2 (p = .004) and Interfaces 1 and 3 (p = .006), but no difference between Interfaces 2 and 3.

These results demonstrate the manual interface (Interface 1), which required more human

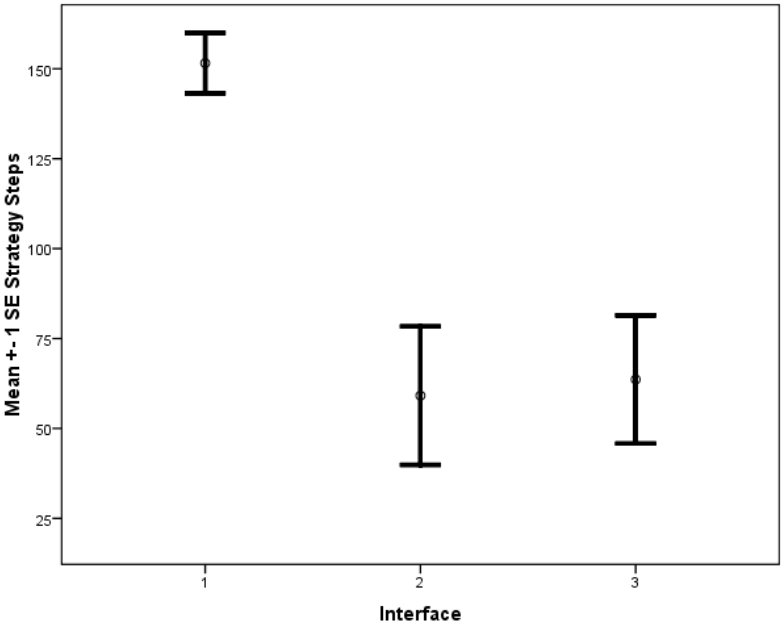


Figure 6: Average number of steps taken per interface for a single solution

interaction than automation, expectedly generated significantly more steps in the solution generation process. However, unexpectedly Interfaces 2 and 3 were not statistically different, even though there were 64% fewer inputs available for Interface 3. Interface 3 was expected to produce fewer steps than Interface 2; however, this was not achieved. This discrepancy is likely due to the fact that operators were not sure if their Interface 3 solutions were correct, and could be an indicator of distrust.

While the simple count of steps is useful as a broad indicator of possibly different strategies, understanding how this number of steps compares to the expected number of steps for generating a solution, assuming the perfectly-trained operator, can reveal how robust an interface is to deviations from expected strategies. This measure, the percentage of actual steps compared to the expected number of steps (termed conformance), was analyzed using the same 2x2 ANOVA model described in the previous section. The expected number of steps was represented in the TRACS grid as a function of the "optimal" strategy as intended by the decision support designer, and then compared to the actual operator results. We recognize what is intended to be optimal by a designer may not actually be, and all decision support systems carry inherent biases

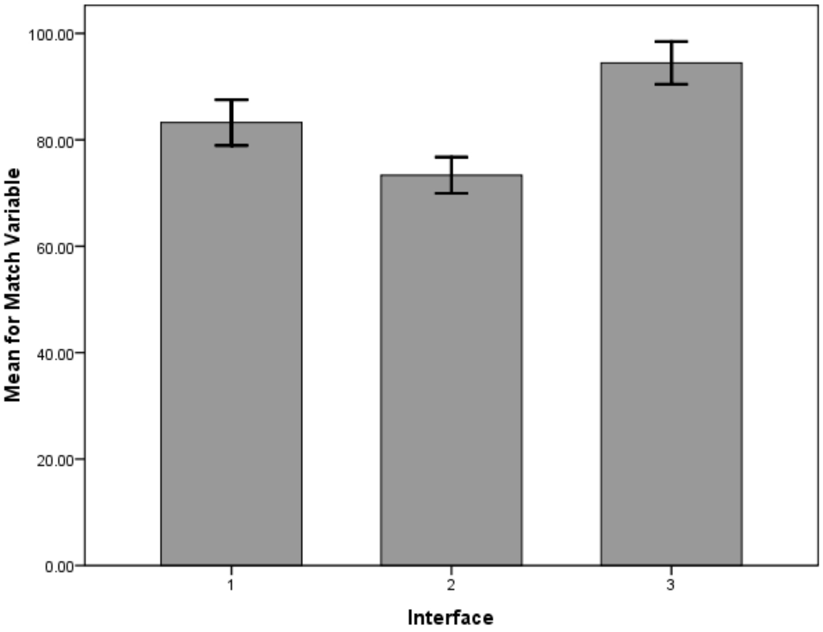


Figure 7: Actual vs. Expected Number of Strategy Steps by Interface

of their designers. However, this is a real limitation of all decision support systems, which we assert can and should be assessed, both to highlight possible design flaws as well as possible training issues. Thus resultant operator strategies that deviate from the expected are not wrong, just different from intended.

The ANOVA results revealed that the interface omnibus test was significant (F(2,22) = 9.442, p = .002), and the experience level was not significant (F(1,22) = 2.814, p = .112). There was no significant interaction between the experience and interface factors. The Tukey pairwise comparisons yielded no statistical difference between Interfaces 1 and 2, but there was a difference between Interfaces 2 and 3 (p = .001), and a marginal difference between Interfaces 1 and 3 (p = .097) (Figure 7).

Those participants with Interface 3 performed the best in terms of strategy conformance, which occurred because of the decreased degrees of freedom for strategy selection, so operators using Interface 3 were more bounded than those using Interface 2: This form of bounded collaboration means that operators with Interface 3 did not have access to the same amount of raw data as those with Interface 1, or fewer filtering criteria than in Interface 2. However, as seen by the performance scores, while the configural display in Interface 3 more closely matched the expected strategies, those additional degrees of freedom for emergent strategies allowed participants to achieve superior performance.

The last strategy analysis was a categorical, qualitative classification based on the TRACS patterns. Analysis of the patterns of all participants' strategies revealed four distinct categories which are 1) Manual, in which participants used only manual features (as seen in Figure 5), 2) Comparative, in which participants created an automated solution and a manual solution, followed by selection of best of the two, with slight modifications of this final solution, 3) Augmentative, which included the creation of an automated solution followed by manual tweaking of that solution, and lastly 4) Automated, which was the use of automation only. Examples of the last 2 strategies can be seen in Figure 8.

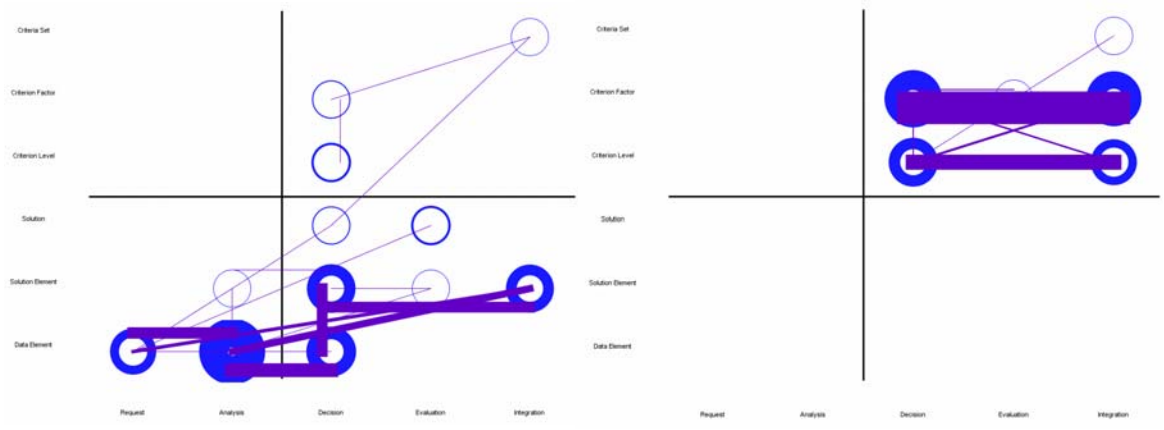


Figure 8: Augmentative (left) vs. Automated (right) Strategy

Table 3 illustrates how these four strategies were used across the three interfaces. Because these strategies were interpreted from the TRACS images as seen in Figures 5 and 8, two raters were used and achieved 100% agreement in the assessment of each participant's strategy. Those cells in gray in Table 3 indicate those strategies were not possible for the given interface.

Table 3: Dominant Strategies per Interface

	Interface 1	Interface 2	Interface 3
1 - Manual	8 (100%)	0	
2 - Comparative		3 (37.5%)	
3 - Augmentative		4 (50.0%)	7 (87.5%)
4 - Automated		1 (12.5%)	1 (12.5%)

Table 3 illustrates that in all cases where fully automated strategies could be used (Interfaces 2 and 3), participants overwhelmingly chose to use some other mixed initiative approach that allowed them greater degrees of freedom. This is not to say operators were adverse to automation. On the contrary, the most frequent strategy used across Interfaces 2 and 3 was the Augmentative approach, which was an approach that heavily leveraged automation. In this approach, participants used automation to quickly create a base solution, and then modified this solution to some small degree before submitting the final solutions. This meta-strategy and avoidance of fully automated solutions seen across Interfaces 2 and 3 appeared to be an effective one, as a Kendall Tau b association performed for the strategy vs. performance variables yielded a significant negative relationship (correlation coefficient = -.499, p = .03). This means that those participants who used more automation typically performed worse that those using comparative or augmentative strategies.

CONCLUSION

We originally proposed that providing a decision support environment that provides human operators the ability to conduct sensitivity analyses in collaboration with automation will improve performance for large multivariate resource allocation problems with uncertainty. However, since collaboration can take on three general forms in that it can be human-centric (-1), balanced (0), and automation-centric (1), it was not clear how such collaboration should be structured to promote the best system performance. To this end, an experiment was conducted in a realistic military mission planning setting with these three mixed-initiative architectures embedded in three separate interfaces. Given a complex resource allocation task for this mission planning setting, the experimental results demonstrated no clear dominant superior collaboration decision support scheme.

As seen in Table 4, the interfaces that resemble a typical spreadsheet decision-space formulation (Interfaces 1 and 2) which allow the human to guide the solution process, proved to have the highest performance scores. However, the interfaces that allowed the operator the ability to offload much of the manual sorting, filtering, and assignment functions (Interfaces 2 and 3) promoted the lowest workload. And while conformance to intended strategies may lead to somewhat less than optimal strategies, it is important to note that the most collaborative, balanced interface promoted the least conformance (Interface 2). This lack of conformance could be a positive result in unanticipated situations with unforeseen variables, but it can also be negative in that operators could more easily violate established procedures or miss critical details. We leave the investigation of the positive and negative aspects of strategy conformance to future work.

Table 4: Experimental Results Summary

	Human-Centric (Interface 1)	Equivalent Balance (Interface 2)	Automation-Centric (Interface 3)
Performance	Best	Best	Worst
Workload	Worst	Best	Best
Conformance	Moderate	Worst	Best

There are other considerations not explicitly considered in this research effort which could have influenced the results. As previously mentioned, trust was likely an issue for those operators using Interface 3, which was nothing like current decision support tools in use. As previous supervisory control research has indicated [16; 17; 19], trust can dramatically influence both operator and system performance, so it is not a trivial issue to consider. All participants in the study reported here all had some exposure to spreadsheet interfaces prior to the study, but none with a configural decision support display so this bias alone could have significant influenced the results.

Another factor that is likely to influence these outcomes is the nature of time pressure. In the mission planning task simulated in this study, time pressure was on the order of minutes to hours, so operators had time to explore the decision space. Had the time pressure been on the order of seconds-minutes, behaviors could have been much different, and indeed previous research indicates that under second-minutes time pressure, operators will often change their strategies [26], including those concerning the use of automation [4]. Thus the nature of how much collaboration a decision support system should promote will have to explicitly consider this question of time pressure.

Given the results of this study, as well as the implication of trust and time-pressure on the outcomes, further work is needed to quantify the cost-benefit nature of the design of collaborative systems. It is generally recognized that operators need to leverage automation for complex resource allocation problems, but that too much automation can negate positive benefit from its introduction. As we have demonstrated in this investigation, if the system designer's goal is to ensure operator conformance, then one collaborative architecture design that promotes sensitivity analysis is superior to that design which would reduce workload. Thus the critical next step in research efforts examining "optimal" design of resource allocation collaborative decision support tools is determining how to design such systems for competing objective functions across various stakeholders, and still ultimately achieve satisficing results.

Acknowledgments

This work was sponsored by the Office of Naval Research (Code 34). We would like to thank Cláudia Ferraz, Andrew Hsiao, Javier Garcia, Andrew Clare, Albert Leung, Anunaya Pandey, and Mark Duppen for their critical support in the experimental data collection and analyses.

References

[1] B.J. Barnett, and C.D. Wickens, Display Proximity in Multicue Information Integration: The Benefits of Boxes. Human Factors 30 (1988) 15-24.

[2] K.B. Bennett, Graphical Displays: Implications for Divided Attention, Focues Attention, and Problem Solving. Human Factors 34 (1992) 513-533.

[3] K.B. Bennett, and D.D. Woods, Emergent Features and Graphical Elements: Designing More Effective Configural Displays. Human Factors 35 (1993) 71-97.

[4] Y. Boussemart, B. Donmez, and M.L. Cummings, Impact of time pressure on behavioral patterns in resource allocation tasks, 15th International Symposium on Aviation Psychology, Dayton, OH, 2009.

[5] S. Bruni, Y. Boussemart, M.L. Cummings, and S. Haro, Visualizing Cognitive Strategies in Time-Critical Mission Replanning, HSIS 2007: ASNE Human Systems Integration Symposium, Annapolis, MD, USA, 2007.

[6] S. Bruni, J. Marquez, A. Brzezinski, and M.L. Cummings, Visualizing Operators' Multivariate Optimization Cognitive Strategies, HFES 50th Annual Meeting, San Francisco, CA, 2006.

[7] M.A. Buttigieg, and P.M. Sanderson, Emergent Features in Visual Display Design for Two Types of Failure Detection Tasks. Human Factors 33 (1991) 631-651.

[8] M.L. Cummings, Automation Bias in Intelligent Time Critical Decision Support Systems, AIAA 3rd Intelligent Systems Conference, Chicago, 2004.

[9] M.L. Cummings, and S. Bruni, Collaborative Human Computer Decision Making. in: S. Nof, (Ed.), Handbook of Automation, in press.

[10] J. Cushing, A Measurement of the Effectiveness of an Overview Display for the Mission to Launch Platform Assignment Process Essential to the Tomahawk Strike Coordinator, Systems Engineering, University of Virginia, Charlottesville, 2003.

[11] P. Dillenbourg, M. Baker, A. Blaye, and C. O'Malley, The Evolution of Research on Collaborative Learning. in: P. Reimann, and H. Spada, (Eds.), Learning in humans and machines. Towards an interdisciplinary learning science., Pergamon, London, 1995, pp. 189-211.

[12] P. Dillenbourg, and J. Self, Designing Human-Computer Collaborative Learning. in: C. O'Malley, (Ed.), Computer Supported Collaborative Learning, Springer-Verlag, Berlin, 1995, pp. 245-264.

[13] P.M. Fitts, Human engineering for an effective air navigation and traffic control system, National Research Council, Washington, D.C., 1951.

[14] G. Gigerenzer, and P.M. Todd, Simple Heuristics That Make Us Smart, Oxford University Press, New York, 1999.

[15] G. Klein, Recognition-Primed Decisions. Advances in Man-Machine Research 5 (1989) 47-92.

[16] J.D. Lee, and N. Moray, Trust, Control Strategies and Allocation of Functions in Human-Machine Systems. Ergonomics 35 (1992) 1234-1270.

[17] J.D. Lee, and K.A. See, Trust in automation: designing for appropriate reliance. Human Factors 46 (2004) 50-80.

[18] G. Lintern, Waite, T., Talleur, D. A. , Functional Interface Design for the Modern Aircraft Cockpit. International Journal of Aviation Psychology 9 (1999) 225-240.

[19] B.M. Muir, Trust Between Humans and Machines, and the Design of Decision Aids. International Journal of Man-Machine Studies 27 (1987) 527-539.

[20] R. Parasuraman, T.B. Sheridan, and C. Wickens, A Model for Types and Levels of Human Interaction with Automation. IEEE Transactions on Systems, Man, and Cybernetics 30 (2000).

[21] R. Parasuraman, T.B. Sheridan, and C.D. Wickens, A Model for Types and Levels of Human Interaction with Automation. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 30 (2000) 286-297.

[22] R. Parasuraman, Sheridan, T. B., & Wickens, C. D., A Model for Types and Levels of Human Interaction with Automation. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 30 (2000) 286-297.

[23] J. Rasmussen, Skills, rules, and knowledge: Signals, signs, and symbols, and other distinctions in human performance models. IEEE Transactions on Systems, Man, and Cybernetics 13 (1983) 257-266.

[24] J. Rasmussen, Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distractions in Human Performance Models. IEEE Transactions on Systems, Man, and Cybernetics SMC-13 (1983) 257-266.

[25] J. Rasmussen, Information Processing and Human-Machine Interaction. An Approach to Cognitive Engineering, North Holland, Amsterdam, 1986.

[26] J. Rieskamp, and U. Hoffrage, Inferences under time pressure: How opportunity costs affect strategy selection Acta Psychologica 127 (2008) 258-276.

[27] G.I. Rochlin, Iran air flight 655 and the USS Vincennes: Complex, large-scale military systems and the failure of control. in: T.R. La Porte, (Ed.), Social responses to large technical systems, Kluwer, London, 1991, pp. 99-125.

[28] J. Roschelle, and S. Teasley, The Construction of Shared Knowledge in Collaborative Problem Solving. in: C. O'Malley, (Ed.), Computer Supported Collaborative Learning, Springer-Verlag, Berlin, 1995, pp. 69-97.

[29] M. Sharples, D. Hogg, C. Hutchinson, S. Torrance, and D. Young, Computers and Thought: A Practical Introduction to Artificial Intelligence, The MIT Press, 1989.

[30] T.B. Sheridan, Humans and Automation: System Design and Research Issues, John Wiley & Sons, Inc., Santa Monica, CA, 2002.

[31] H.A. Simon, R. Hogarth, C.R. Piott, H. Raiffa, K.A. Schelling, R. Thaier, A. Tversky, and S. Winter, Decision Making and Problem Solving. in: N.A.o. Sciences, (Ed.), Research Briefings 1986: Report of the Research Briefing Panel on Decision Making and Problem Solving, National Academy Press, Washington D.C., 1986, pp. 16.

[32] P. Smith, E. McCoy, and C. Layton, Brittleness in the design of cooperative problem-solving systems: The effects on user performance. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 27 (1997) 360-371.

[33] A. Tversky, and D. Kahneman, Judgment under Uncertainty: Heuristics and Biases. Science 185 (1974) 1124-1131.