Tracking Resource Allocation Cognitive Strategies for Strike Planning

Sylvain Bruni, Mary L. Cummings

Massachusetts Institute of Technology - Humans and Automation Laboratory
MIT Room 33-305, 77 Massachusetts Avenue, Cambridge, MA 02139 - USA

Abstract: Current procedures in the planning of Tomahawk Land Attack Missile (TLAM) strikes in the United States Navy require an operator (the Strike Coordinator) to match available missiles to preplanned missions. Such strike planning is essentially manual, with little help from any computerized tool or decision-support system. This causes high workload for the Strike Coordinator, and drastically increases the probability of error and degraded assignments in the matching process. Research in the Humans and Automation Lab at MIT is investigating a potential decision-support system aimed at helping Strike Coordinators increase both their efficiency in the matching of missiles to missions, and the quality of the resulting assignments. This paper presents a recent experiment where actual U.S. naval operators tested different prototypes of automated decision-support interfaces for TLAM strike planning. Of primary interest was examining the different cognitive strategies used by the subjects in order to complete the assignment process. To this effect, we developed a visualization tool, TRACS (Tracking Resource Allocation Cognitive Strategies). TRACS captures the different cognitive steps the subjects visited while attempting to solve the resource allocation problems. Interesting findings are presented and discussed.

Keywords: human-computer collaboration, cognitive strategies, visualization tool, levels of automation.

1. Introduction

Tomahawk Land Attack Missile (TLAM) strikes require advance planning including the complex resource allocation and optimization problem of assigning missiles to missions. Typically, a TLAM Strike Coordinator will use a Personal Computer – Mission Distribution System (PC-MDS) to visualize all missions that need to be carried out during the strike. Then, based on his knowledge of the resources (the missiles) and their respective characteristics available on the many surface warships or subsurface vessels, the Strike Coordinator will assign them to the missions, matching the individual characteristics of the missiles to the requirements of each missions [1]. Whereas PC-MDS allows Strike Coordinators to electronically combine missions and missiles for later transmission to the different Launchers (the ships or submarines carrying the missiles), it does not provide any help to the operator to improve the assignments from an optimization perspective. For example, one key feature for decision support would be to advise the Strike Coordinator on what missiles are the best, the most likely or the most capable to achieve a particular mission. As such options are not available, Strike Coordinators mostly use pen and paper to develop a plan and decide what combinations of missiles and missions would be acceptable. Because this planning is mostly manual, this procedure causes high memory load and increased workload. Generally, reducing the overall load on the operator will lead to an increase in performance and quality of the matching; the introduction of computerized decision-support is investigated.

After the development of the interfaces and associated heuristic search algorithms and the subsequent cognitive walkthrough, field users and subject matter experts analyzed three interfaces with different level of automated support for TLAM strike planning [2]. Following their recommendations, the interfaces were modified and tested with Navy personnel at the Naval Station Newport in Newport, Rhode Island and at the Submarine Base New London in Groton, Connecticut. Of primary focus was defining the cognitive strategies used by the subjects to navigate through the different interfaces in order to understand their reasoning behavior. This paper introduces TRACS (Tracking Resource Allocation Cognitive Strategies), a visualization tool created to represent the cognitive strategies used by the subjects to solve the particular resource allocation problem of assigning missiles to missions in a TLAM strike. For each interface configuration tested, we used TRACS to visualize our subjects' performance on the task. TRACS allowed us to compare the different cognitive strategies employed with the different interfaces.

2. Background

2.1 Levels of automation

A recurring challenge in the introduction of automation in a resource allocation problem is to evaluate how much automated decision support is needed, and how much will be most beneficial to the human operator. A widely used and accepted model, the SV scale of levels of automation [3] orders ten possible combinations of human and computer contributions in decision-making and action-taking, for partially or fully automated systems under human control (Table 1). These levels range from 1, where the human is fully in charge both in terms of decisions and actions, without any intervention of the computer; and increase up to 10, where the automation is fully in control, keeping the human out of the loop. In between, the involvement of automation progressively increases.

Although reducing workload is a generally desired design principle, to do so by increasing the involvement of automation in the process (that is, increasing the level of automation) has its own limits. Past research has shown how detrimental automation can be, when an inappropriately high LOA is embedded in the system [4, 5]. Of particular concern, automation bias and complacency will lead to a degradation in skills and performance, and, oftentimes, potential loss of situation awareness.

Table 1 : SV scale of Levels of Automation (LOA)

LOA	Automation description
1	The computer offers no assistance: human must take all decision and actions.
2	The computer offers a complete set of decision/action alternatives, or
3	narrows the selection down to a few, or
4	suggests one alternative, and
5	executes that suggestion if the human approves, or
6	allows the human a restricted time to veto before automatic execution, or
7	executes automatically, then necessarily informs humans, and
8	informs the human only if asked, or
9	informs the human only if it, the computer, decides to.
10	The computer decides everything and acts autonomously, ignoring the human.

2.2 Information Type

Mission/missile planning occurs prior to a strike, and Strike Coordinators both need to create mission/missile combinations and distribute these to the different Launchers (ships and/or submarines). Strike planners must consider target and mission priorities, different warheads on the missiles, and guidance capabilities of the missiles. Assigning missiles to missions when planning a TLAM strike requires evaluating the respective impact of all constraints on both the feasibility and the quality of the solution (a solution is defined as a set of assignments of missiles to missions).

A first set of characteristics is termed hard constraints. These include, in the present case, the launch basket (where the missile should be launched from, in order to be on the correct route to a target), the warhead (or payload needed on the missile to destroy the intended target), and the navigation equipment. For example, due to terrain constraints, a mission may only be achievable using the Global Positioning System (GPS) as navigation equipment, to monitor accurately the position of the missiles However, not all missiles carry the adequate navigation equipment to work with GPS. Hence, it is the Strike Coordinator's responsibility to assign to those missions requiring GPS equipment, only missiles that possess this equipment. This generalizes to the two other hard constraints previously mentioned. Computationally, hard constraints are trivial to manage: either the characteristics of the mission and the missile match, or they do not.

Additional constraints may complicate the Strike Coordinator's task such as probabilistic information. For example, in our simulation, we assigned each warship carrying missiles a firing rate, which is a virtual percentage describing how successful a Launcher is at launching missiles. One consideration a TLAM Strike Coordinator may make is to prioritize the use of missiles aboard Launchers with a high firing rate, and to assign them to high priority missions. Such a strategy would increase overall mission success, as high priority missions would get a better chance to be successfully carried out.

Another typical constraint of complex resource allocation problems is the optimization of one or more variables. In the case of a mission-missile assignment procedure, this can be described by a variable we termed "days to port". "Days to port" refers to the number of days before a Launcher is supposed to get back to port. For safety reasons, it is of interest to minimize the number of missiles aboard ships when they enter ports. Hence, this could lead Strike Coordinators to prioritize the missiles aboard those ships with the fewest days to port.

Whereas these three types of information (hard constraints, probabilistic and optimization information) may be manageable when being taken into account separately, it is expected that evaluating them simultaneously in a stressful, mission-critical, time-pressured environment will generate potentially excessive workload for a Strike Coordinator [2]. Hence, some level of automated decision-support including easy-to-understand visualizations should assist TLAM Strike Coordinators in their task. In order to appreciate where automation could help, a deeper understanding of the cognitive processing associated with the resource allocation problem is needed. Moreover, by comprehending what cognitive steps operators go through while interacting with our software will allow us to potentially find the cognitive pitfalls linked with the specific interfaces and features implemented in StrikeView, which is our decision-support software for strike planning / resource allocation.

2.3 Interfaces

Interface 1 - LOA 2: The first matching interface designed (Figure 1) for the StrikeView decision-support software corresponds to a Level of Automation 2, where "the computer offers a complete set of decision/action alternatives", that is, only provides basic tools to the human to make information processing easier, but entirely leaves up the decisions and actions to the human operator. (See [2] for a complete description of the interfaces).

In the missile-mission matching problem, this translates into providing the Strike Coordinator with filtering and sorting tools that will help accomplish time-consuming tasks of the matching process. With Interface 1, the operator can ask the computer to sort the lists of missions or missiles by characteristic, in order to achieve faster identification of the pairs that obey the same hard constraints. In addition, a multi-sorting option is available in Interface 1, in order to nest sorting constraints. Besides sorting capabilities, Interface 1 provides a powerful filtering tool. When a mission (or a missile) is selected by the Strike Coordinator, the computer will automatically highlight the missiles (or the missions) that satisfy the hard constraints for an adequate match. The resources that violate these constraints are grayed out, providing a forcing function that prevents operators from making erroneous assignments. Moreover, additional color coding helps the human operator scan the lists of missions and missiles. If a mission (or a missile) presents characteristics that no missile (or mission) can fulfill at all, it will appear in red, so as to signal to the operator that no time should be wasted on this match. Also, missions and missiles that have already been assigned appear on a blue background, which tells the Strike Coordinator that those missions and missiles have already been assigned and are now part of the current solution.

Finally, Interface 1 provides the Strike Coordinator with displays of status assessments. At the bottom left side of the screen are two areas that display the missions and missiles that are unusable for the problem (either because no corresponding matching part exists, or because the current assignments have used all those potential corresponding matching parts). At the bottom right side of the screen are horizontal bars that fill to indicate how many targets have been assigned so far, with a breakdown by priority.

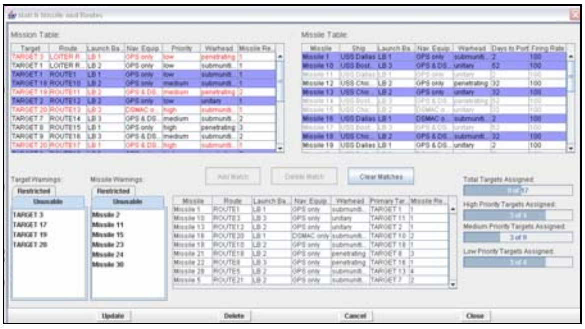

Figure 1 : StrikeView Matching Interface 1

Interface 2 - LOA 2 and LOA 3: Matching Interface 2 (Figure 2) includes both LOA 2 and LOA 3. Interface 2 features part of Interface 1, including the lists of missions and missiles. These lists can be interacted with using the filtering and sorting tools described earlier (LOA 2). In addition to these basic information visualization tools, Interface 2 helps "narrow the selection down to a few [decision/action alternatives]" (LOA 3, Table 1). An embedded automatic assignment tool, dubbed "Automatch", allows the Strike Coordinator to partially automate the assignment process. The Strike Coordinator specifies what criteria should be taken into account (e.g. "Assign GPS Missions"), and in what order of priority (e.g. first "Assign Penetrating Missiles", then "Assign Submunition Missiles"). By a simple click on an "Automatch" button, a heuristic-based algorithm searches the domain space for solutions that follow the constraints and criteria submitted by the Strike Coordinator.

In addition to this customizable, automated search algorithm, Interface 2 adds a visualization of the probabilistic and optimization information. A graphical representation of the days to port and firing rate information is displayed, along with the number of missiles left on each ship.

Finally, an option to save the current solution is provided in Interface 2. The saved set of assignments appears at the bottom right of the screen, represented by the status assessments graphics (horizontal bars, and days to port / firing rate / missiles left display). When a solution is saved, the operator can either build another solution or modify the one saved, and then use both status assessment displays to compare the solution and decide which one is better.

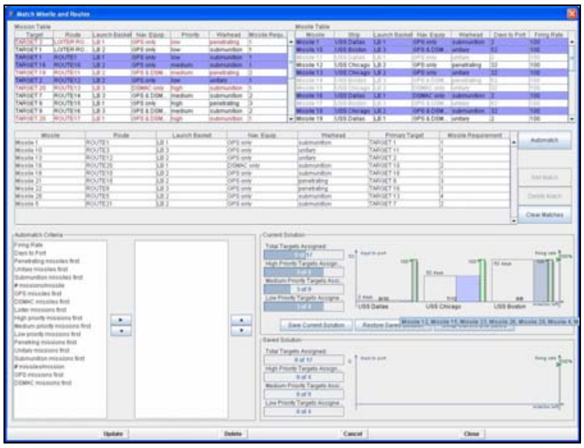

Figure 2 : StrikeView Matching Interface 2

Interface 3 - LOA 4: Interface 3 (Figure 3) is an example of LOA 4: the computer "suggests one alternative". This alternative, or solution (set of assignments), is obtained by clicking on the top left "Automatch" button of the interface. This "Automatch" is a heuristic-based algorithm that first prioritizes the missiles (least capable missiles first), then prioritizes their matching missions (least "matchable" first). Strike Coordinators can only slightly modify this algorithm by adjusting the weight of the days to port and firing rate criteria in the missile prioritization part of the search (using the sliding cursor at the top of the screen).

The graphical display in Interface 3 is a breakdown of status assessment by mission priority (loiter, high, medium, low), with a second breakdown by warhead (penetrating, unitary, submunition). This allows the operator to see how many of each type of mission has been assigned matching missiles by "Automatch". Once a solution is computed and displayed, the operator can use Interface 3 to adjust the solution, by forcing assignments that were not allowed by "Automatch". Each vertical slider indicating the number of missions of a specific type can be queried up (or down) to increase (or decrease) the number of assignments in that category. When the Strike Coordinator wants to increase the number of assignments of medium priority, penetrating missions, for example, a click on the corresponding up arrow in Interface 3 will have the computer search first for an unassigned medium priority, penetrating mission, and then for a missile (unassigned or not) that will fulfill all the said mission's constraints (e.g. navigation equipment). As a result, it can be that a missile that was assigned to a higher priority mission is reassigned from that mission to a mission that falls under the chosen category.

The central area of Interface 3 is a graphical representation of overall status assessment, and how much has been assigned so far. Ideally, the green area would fill the entire central area, showing that all missions have been assigned a missile (100% for loiter, high, medium and low priority missions). The overlaying grey shading is a representation of the solution as it stood before the very last operator's command, as a mean of comparison between solutions.

Finally, note that this particular interface does not allow the operator to see the specifics of the assignments. No knowledge of what missile is assigned to what mission can be gained here, as only high level information is displayed.

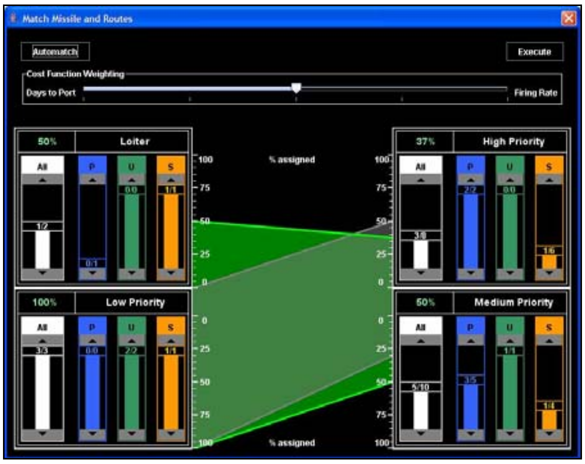

Figure 3 : StrikeView Matching Interface 3

3. Methods

3.1 TRACS

In order to visualize operators' cognitive strategies in the resource allocation problem of assigning missiles to missions, using the interfaces described in the previous paragraphs, we developed TRACS (Tracking Resource Allocation Cognitive Strategies), as a two dimensional representation, which axes include the level of information detail (LOID), and mode (Figure 4).

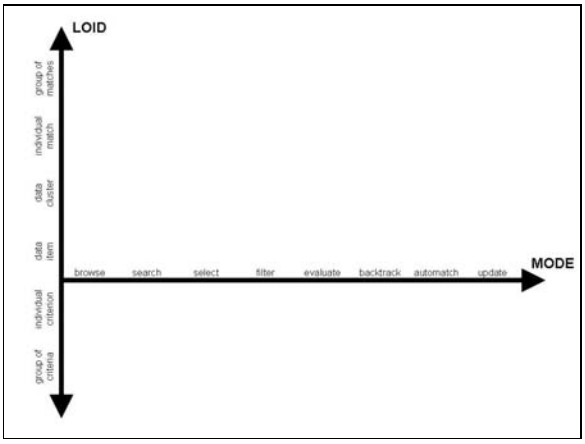

Figure 4 : TRACS: 2D configuration

The LOID axis refers to the information used by the operator at every step of his strategy. With respect to our specific problem, we define two exclusive types of information: the data and the criteria. Data refers to the resources involved in the problem, which are either missions or missiles. We subsequently divide the data portion of the LOID axis into four categories, with increasing level of sophistication: "data item" (e.g. Mission 21), "data cluster" (e.g. Missiles with Penetrating Warheads), "individual match" (e.g. Mission 13 paired with Missile 2), and finally "group of matches". The Criteria portion of the LOID axis is divided into two categories: "individual criterion" or "group of criteria".

The MODE axis regroups six cognitive actions operators have been observed to implement when solving resource allocation problems (browse, search, select, filter, evaluate, and backtrack), and two actions specific to the interfaces provided (automatch and update).

The steps of an operator's solving strategy are represented by circles within the cell corresponding to the information (LOID) and the action (MODE) employed at each step. The width of the circle's border is proportional to the number of times a specific step is visited. Steps are linked to one another by edges whose widths are proportional to the number of times the two connected steps follow one another during the overall solving strategy.

An example of a cognitive strategy is shown in Figure 5. In this situation, the operator is using Interface 2 to solve the problem. Step 1: the operator browses the data, by individually looking at all resources available. Step 2: the operator selects one criterion in the list of criteria available. Step 3: the operator launches an automatch on this criterion. Step 4: the output is deemed unsatisfactory, so the operator removes one assignment from the matching table (this is backtracking on an individual match). Step 5: the operator saves the current solution (which will allow him to compare the current group of matches to another solution). Step 6: the operator selects another individual criterion in the list of criteria available. Step 7: the operator launches an automatch on this criterion. Step 8: the output is deemed unsatisfactory, so the operator clears the matching table (this is backtracking on a group of matches). Step 9: the operator selects several criteria in the list of criteria available. Step 10: the operator launches an automatch on this group of criteria. Step 11: the output is deemed satisfactory, and the operator updates the current solution to end the task.

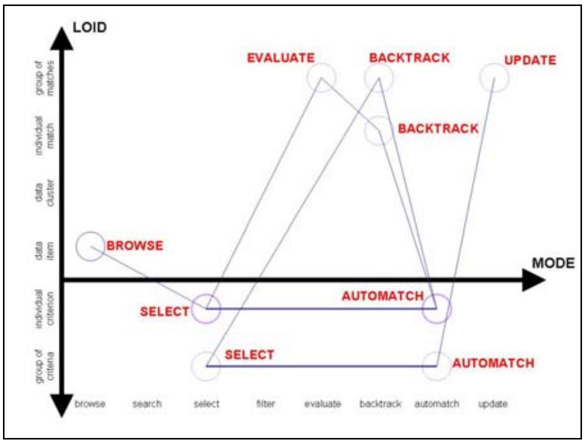

Figure 5 : Example of cognitive strategy visualization using TRACS

3.2 Subjects

Twenty subjects (18 males, 2 females) aged 25 to 37 (mean: 30 ± 2.6 sd) participated in this experiment. Subjects were from the Surface Warfare Officers School Command (SWOSCOM), at the Naval Station Newport, in Newport, RI, or from the Submarine Base New London in Groton, CT. All subjects had between 4 and 28 years of service in the U.S. Navy (mean: 8 ± 3.5 sd). All subjects had the same basic Navy strike training, two had extensive experience with TLAM Strike planning (more than 500 hours each), while seven had about 100 hours of experience each with TLAM Strike planning. Thirteen subjects had participated in live operations or exercises involving the use of Tomahawks, and three additional subjects had undergone TLAM classroom training.

3.3 Design of Experiment

Five interface configurations were tested: interface 1 (I1), interface 2 (I2), interface 3 (I3), interfaces 1 and 3 together (I13), and interfaces 2 and 3 together (I23). Each subject was randomly assigned one interface configuration, and performed three scenarios in a random order: a complete scenario (scenario C: at least one solution exists for the matching of all missions), an incomplete scenario (scenario I: not all missions can be matched at the same time), and a replanning scenario (scenario R: subjects start from a pre-computed solution and are given new rules of engagement - namely, all medium priority missions become high priority missions). Each scenario involved the matching of 30 missions with 45 missiles.
All subjects were tested on similar workstations: a Dell Dimension 8250 with a Pentium 4 processor running at 2.8 GHz, with 1.5 GB of RAM, with a dual-screen setup consisting in two identical Dell 19in flat panels, at a 1280x1024 resolution each, placed side by side. Interface 1 and 2, when used, were always displayed on the left screen. Interface 3, when used, was always displayed on the right screen.

3.4 Protocol

Subjects trained in a 30-minute session with two practice scenarios. Training included an explanation of the environment and the task to perform, a walkthrough of the interface configuration that the subject would be using, and a short practice session where subjects would use their assigned interface configurations with two different practice scenarios. All subjects went through the exact same training presentation and the same practice scenarios. Only the interface configuration differed between subjects.
After this training, subject started the experiment. During the experiment, screen capture software was running in the background, recording the screen activity during the entire experiment. Subjects were provided with Rules Of Engagement, describing the three main criteria ordered by priority, to be used to solve the problem (Table 2).
A 30-minute debriefing session followed the experiment. One randomly chosen scenario performed was replayed using the screen capture software, and subjects were asked to explain to the experimenter what strategy they used to solve the problem.

Table 2 : ROE Priority List of Criteria

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.

3.5 Measure of Performance

Performance scores on the scenarios were inferred by inspection of the updated solutions after completion of all scenarios. Namely, the matching percentages for each type of mission were compared to those of the best possible solution. Matching percentages were grouped under the quadruplet a/b/c/d, where *a*, *b*, *c* and *d* respectively represent the percentages of loiter, high, medium and low priority missions matched. The "best possible" solution refers to a solution that follows PL1 and PL2 (see Table 2) and maximizes the total number of successful matches.

4. Results

4.1 Using Interface 1

Figure 6 presents the TRACS visualization of three different subjects solving scenario I with interface 1. No subject using interface 1 managed to reach the best possible solution (80/85/75/70). However, it is interesting to notice that the two subjects (top and center, Figure 6) found equivalent solutions (80/57/50/80) in roughly the same amount of time (respectively 6'03" and 6'52"), and followed very similar cognitive strategies. However, the subject at the bottom did not perform as well. This subject did not manage to match as many missions and missiles as the others did (80/57/50/70), and took significantly more time to execute the scenario (26'42"). It clearly appeared that this subject was lost in the interface: he continuously backtracked on the same data and spent a lot of time browsing the data tables without apparent purpose.

4.2 Using Interface 2

Figure 7 shows two TRACS visualizations for scenario R (replanning), solved with interface 2 only (I2), and with interfaces 2 and 3 combined (I23). Both clearly show the same pattern of use of the automation, pictured as a triangle between the selection and filter of individual criteria, automatch, and the evaluation of the computed solution. In this case, both subjects found solutions equivalent to the best solution (80/57/50/80). However, although the subject using I23 followed more cognitive steps (as represented by the sum of the width of all circles), he reached his solution faster than the other subject (3'55" vs. 5'13"). The difference can actually be understood from the TRACS visualization (Figure 8): it appears that the subject using I2 spent a significant amount of time (3'09") trying to manually compute a solution (browse, search, and filter of several sets of data) before saving it and using the automatch function, which then quickly led to the best solution.

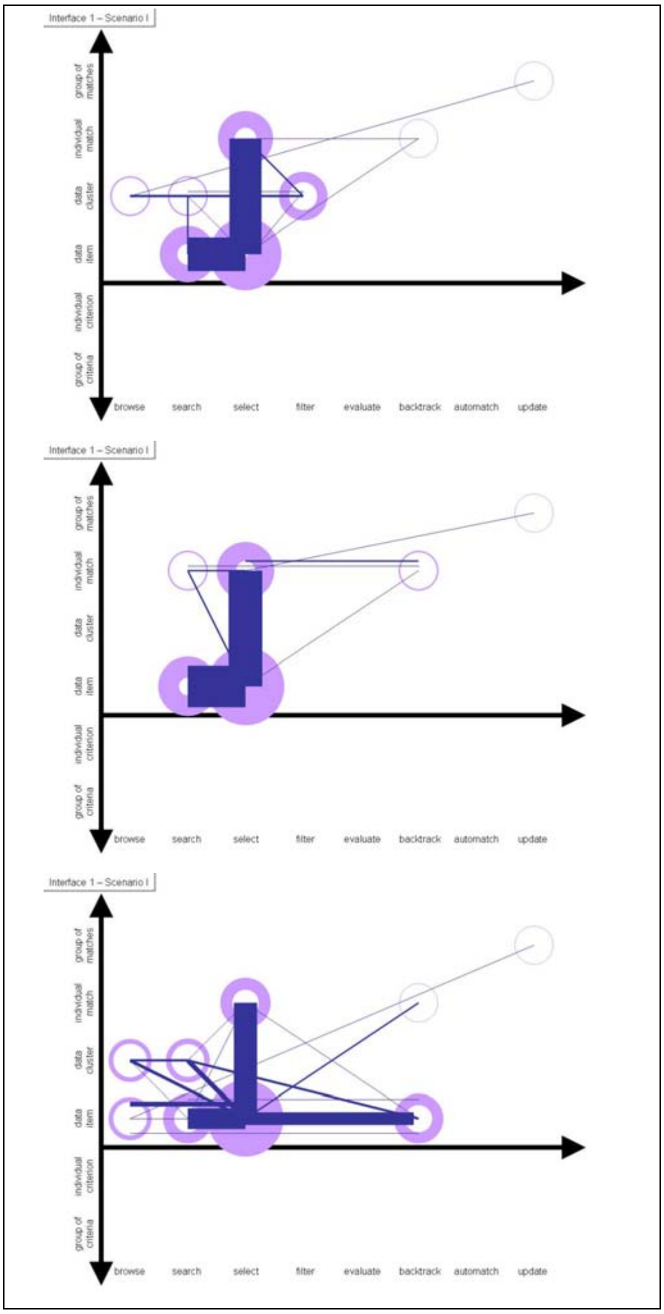

Figure 6 : TRACS visualization for solving scenario I with interface 1.

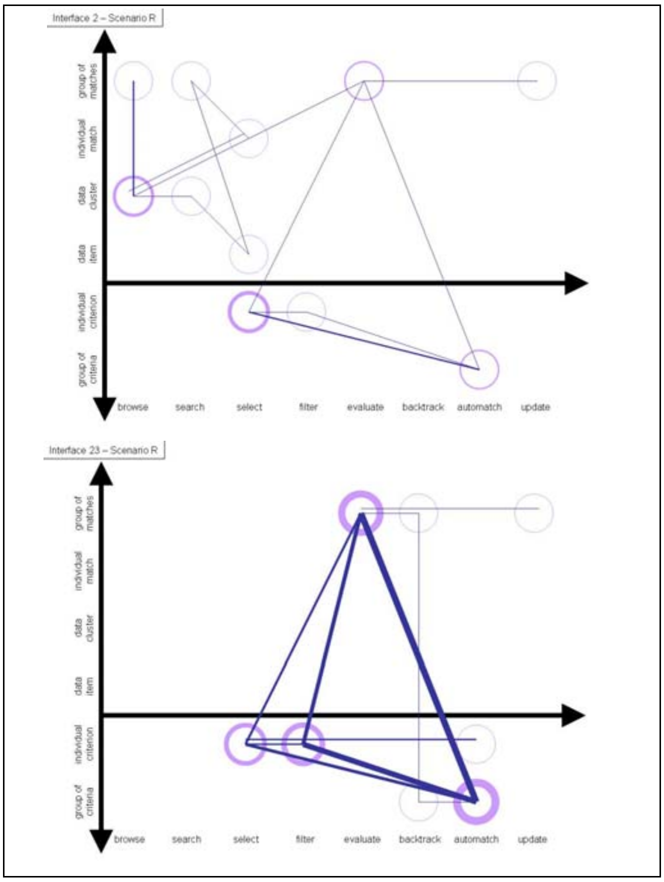

Figure 7 : TRACS visualizations: Interface 2 (top) vs. Interfaces 2 and 3 (bottom)

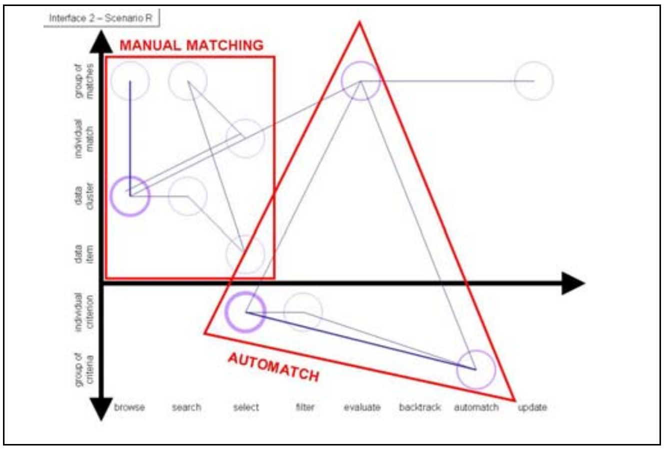

Figure 8: Manual matching and use of automatch on TRACS for Interface 2

4.3 Using Interface 3

Figure 9 displays the TRACS visualizations of three different subjects solving scenario R with interface 3. The subjects at the top and center both reached the best possible solution (80/57/50/80) quickly (respectively in 1'23" and 2'13"). However, the subject at the bottom did not reach the best possible solution (his solution was a 60/57/50/80), and took 4'26" to complete this scenario. This last subject did not use automatch at all, and chose to leverage specific criteria (bottom left of the representation). However this did not pay off as the subject constantly came back to previously generated solutions (backtracking) to explore the solution space in other ways, without ever finding the best solution.

4.4 Using Interfaces 1 and 3

Interesting results appeared when applying TRACS to scenario I solved using interfaces 1 and 3 (I13), as depicted in Figure 10. The subject at the top decided not to use the automated tools of interface 3 and solved the problem using interface 1 only. His TRACS representation looks very similar to those subjects with just interface 1 (Figure 6). This subject and that of Figure 6, bottom, both found equivalent solutions (80/57/50/70). However, the subjects in the center and at the bottom of Figure 10 both reached equivalent, but poor solutions (80/57/37/60). The subject at the center, who completed the scenario in

20’23”, was “all over the place”, as depicted in his TRACS representation. There was no clear strategy implemented, as he continuously switched back and forth between manual matching (using the data tables), and automatch. The subject at the bottom completed his scenario in 11’31”, and implemented a more understandable strategy: first he tried to use the automatch function of interface 3, and then decided to start over again, but matching missions and missiles manually. In the process, this subject spent a lot of time browsing the data tables without any clear sense of what to do with the data.

4.5 Using Interfaces 2 and 3

Finally, Figure 11 displays how three different subjects solved scenario R with interfaces 2 and 3. All three of them found a solution equivalent to the best solution possible (80/57/50/80), but using different strategies, and in different time periods (respectively from top to bottom, 2’11”, 3’28”, and 7’28”). The subject at the top only used interface 2 and its customizable automatch to rapidly converge to the best solution. The subject at the center used alternatively the automatch of interface 2 and the automatch of interface 3, to approach the best solution. The subject at the bottom completely ignored the automated tools available in interfaces 2 and 3, and derived the solution manually using the data tables.

5. Discussion

5.1 Interface 1

As shown in Figure 6, TRACS indicates simple cognitive strategies of two subjects (top and center) who reached a “good enough” solution in a relatively short amount of time. During debriefing, these two subjects elaborated on the strategy they used to solve the problem. It appeared that both had a clear *a priori* plan on how to solve the task using interface 1. They primarily tried to solve the problem by blocks, and optimize their solution “by block”. Indeed, both first tried to make the best possible assignments on all loiter missions, and then iterate with other types of missions, in decreasing order of priority. However, the third subject (bottom of Figure 6) did not lay out a specific strategy during debriefing. Instead he explained that he “tried to match the missions as they came”, without any real wish to optimize or think ahead of the current match. He also mentioned that he tried not to backtrack, which contradicts what TRACS shows (heavy backtracking on the data). This subject later on described how he would sacrifice low priority missions to afford better firing rate on loiter and high priority missions. Such a statement reinforces what TRACS depicts (backtracking) and contradicts the earlier subjective report. It seems probable that this subject did not realize he was backtracking as much as he thought he did, which confirms the hypothesis that he was overwhelmed with the task.

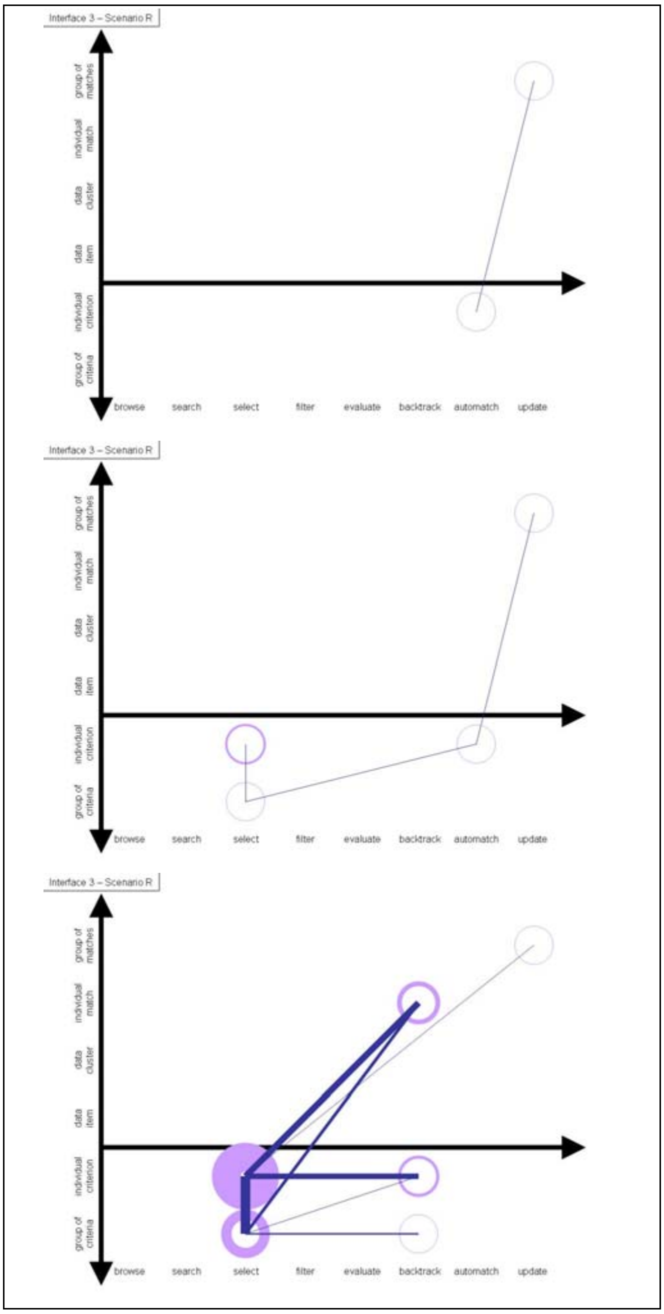

Figure 9 : Top to bottom: decreasing performance in solving scenario R with interface 3.

5.2 Interface 2

The observation established in Figure 8, that one subject was using two separate strategies in a single interface (manual match and automatch), was confirmed through the post-experiment debriefing. This subject explained: “I first tried to do the replanning with the new [Rules of Engagement] manually, then saved my solution, and then realized an automatch to see what the computer would find. After that, I compared my solution and that of the computer, and I decided to validate [the computer’s] because it looked better”.

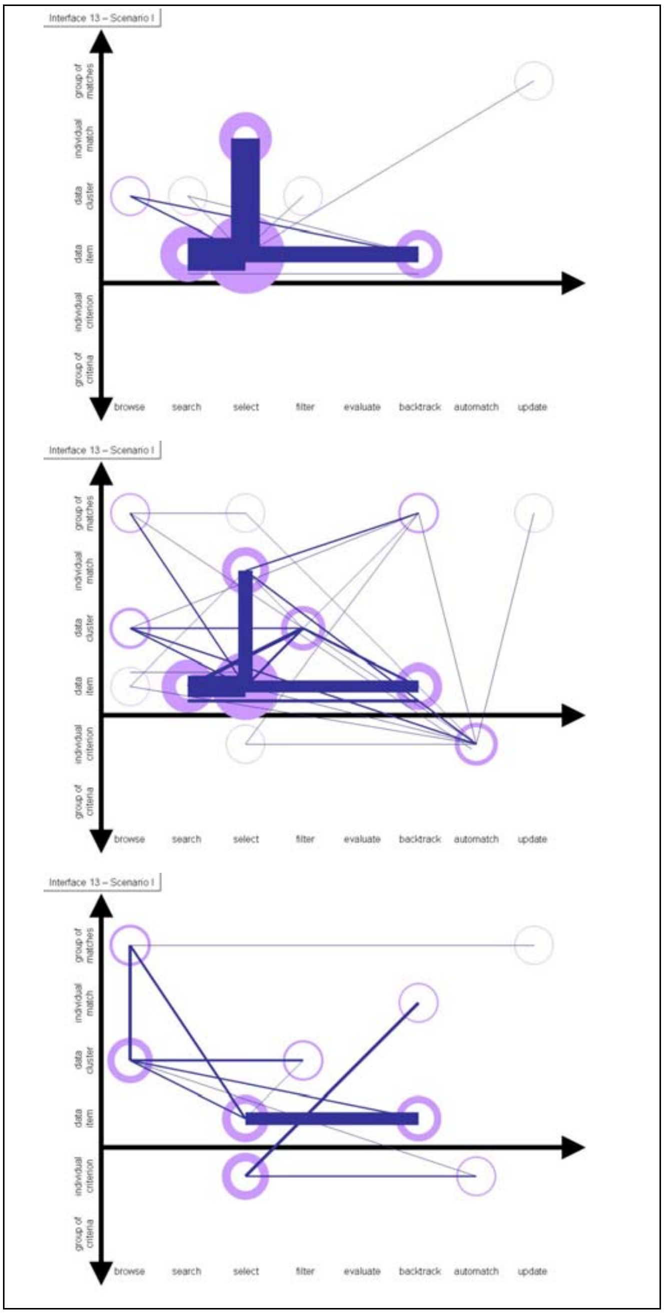

Figure 10 : TRACS visualization for solving scenario I with interfaces 1 and 3

5.3 Interface 3

For interface 3, the most graphical of the interfaces with automatch capability, one subject (Figure 9, bottom) did not use automatch in this scenario, despite the fact that automatch was programmed to output a perfect solution (all missions matched). When asked during debriefing, he explained that in the two other scenarios that he had performed before this replanning scenario, he followed this process: “I first use automatch to get to some solution, and then I modify it with the vertical sliders”. Since the replanning scenario starts with a pre-computed solution to modify, it is plausible that this subject did not even think about using automatch as a solution was already there, waiting to be modified. Instead, he may just have followed the pattern he previously established, that is modifying the solution using the vertical sliders, which is what TRACS shows. Indeed, in such a case, TRACS was helpful in noticing what steps were missed (here: the automatch).

Both other subjects whose cognitive strategies are depicted in Figure 9 explained during the debriefing that when they clicked automatch, it presented them with a complete solution; hence they did not see the need to do anything else.

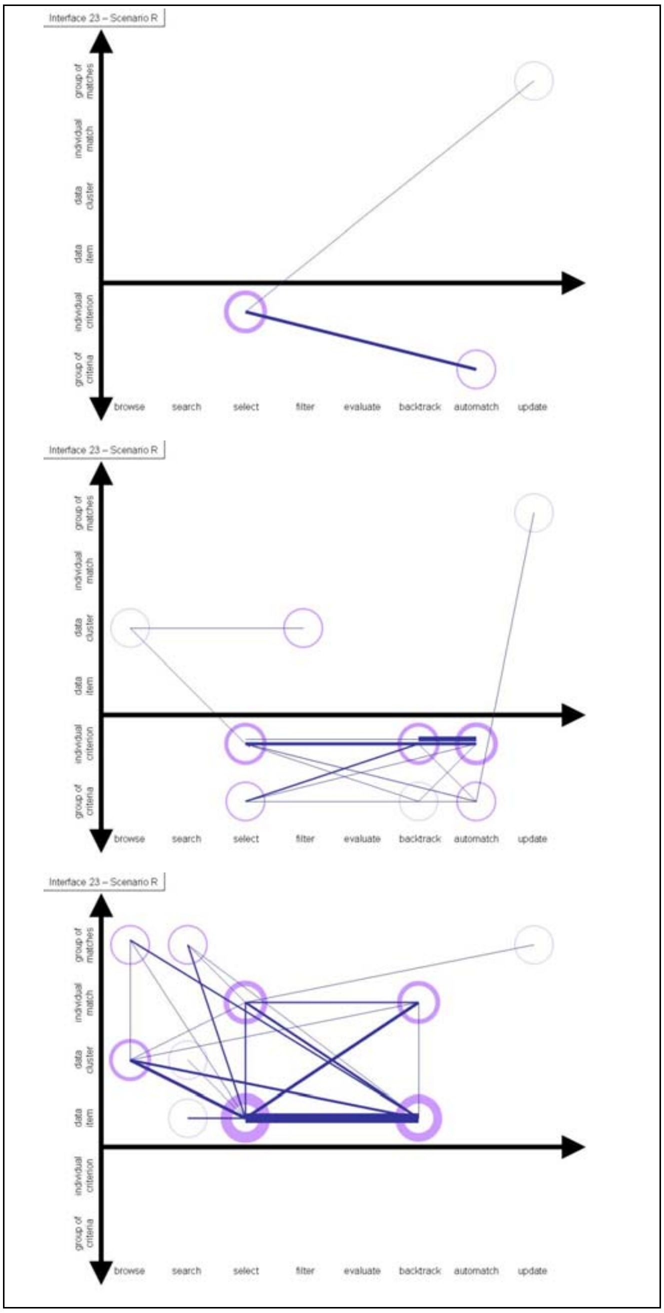

Figure 11 : TRACS visualization for solving scenario R with interfaces 2 and 3

5.4 Interfaces 1 and 3

In general, the performances of those subjects with the combination of interfaces 1 and 3 were poorer than expected. The subject in this condition who performed the best essentially ignored the automatch of interface 3 (Figure 10, top). During the debriefing session, he explained that he "never used [interface 3] because it looks complicated, and [he likes] spreadsheets". TRACS shows that this subject never used automation as the cognitive strategy never goes in the criteria part of the LOID/MODE plane (below the MODE axis). This subject also explained that his strategy was to assign first the most constrained resources (least capable missiles) in order to maximize the overall number of matches. This explains the strong links between search, selection and backtracking, the main cognitive steps to perform when navigating the data, looking for these most constrained missiles.

The other two subjects in this condition appeared to get lost easier than subjects using single interfaces. One subject explained he tried to use a dual strategy in that his purpose was to come up with his own solution using interface 1, and compare it to another solution created by the computer with interface 3. In the end, it appears that this subject started a manual matching process with interface 1, then cleared his solution, and computed another one with automatch in interface 3. It is important to note that he had to memorize in some way the first solution to compare it to the second, as no option to save is available in interface 1. Moreover, this subject backtracked from automatch to manual match seven times and never achieved a good solution. The remaining subject in this condition (Figure 10, bottom) tried to see what the computer would do (using interface 3's automatch), but then, because she "was not sure how the solution was computed", she started to browse the data tables to try to understand the automatch's solution, but quickly stopped to compute a manual solution. She also achieved a sub-par solution.

5.5 Interfaces 2 and 3

For the combination of interfaces 2 and 3, the reluctance of some subjects to use the automation appeared again. One subject "did not use [the automation] because of a lack of comfort… in the Navy, we are trained to not trust the automation… that's why I never used it". He added: "I wanted to be able to control everything… sort of in the Navy way…". This resulted in a complex strategy representation in TRACS more or less similar to that of interface 1 where the subject would select and backtrack between data items, as well as search intensively through the data tables (Figure 11, bottom). This subject added that he would have preferred to be able to compute his own solution manually, have the computer compute its, and then compare both solutions and decide which one to validate. Indeed, it is possible to implement such a strategy, as interface 2 has an option to save solutions for easy comparisons. But this subject did not use it.

The other two subjects (Figure 11, top and center) more readily accepted the role of automatch. On subject stated that "doing it manually was too difficult" (based on previous scenario), hence he "decided to use the customizable automatch all the time from now on" (Figure 11, top). The remaining subject on Figure 11 (center) partially trusted automatch in that he believed the customizable automatch of interface 2 would be powerful, but he "wanted to check the solutions". This explains the multiple automatches, and selection and backtracking of criteria.

6. Conclusion

As shown with our TLAM Strike Planning experiment, TRACS allows for capture of cognitive strategies used by human operators in the complex problems of multi-variable resource allocation. Not only does this visualization tool enable quick comparison of strategies between different operators (e.g. as depicted in Figure 11), but it also allows for a rapid understanding of what tools are being used by an operator (e.g. see Figure 8). TRACS can also be used to monitor operators' strategies, understand where they spend their cognitive resources, and how much of an understanding they have of what is happening. Because of the ability of TRACS to characterize an operator's mindset during the task of solving a resource allocation problem, we can determine which components of the interfaces, or the interfaces themselves, cause problems or are particularly useful.

7. Acknowledgments

This research effort is sponsored by the Office of Naval Research Long Range Navy and Marine Corps Science & Technology Research Opportunity (BAA 04-001). The views and conclusions presented in this paper are those of the authors and do not represent an official opinion, expressed or implied, of the Office of Naval Research or the acknowledged individuals. Special thanks to Albert Leung, who programmed the interfaces, to Anunaya Pandey, who assisted in data extraction and analysis, and to Mark Duppen, who assisted in interface pre-testing.

8. References

[1] 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," in *Systems Engineering*. Charlottesville: U. of Virginia, 2003.

[2] S. Bruni and M. Cummings, "Human Interaction with Mission Planning Search Algorithms," presented at Human Systems Integration Symposium, Arlington, VA, 2005.

[3] T. B. Sheridan and W. Verplank, "Human and Computer Control of Undersea Teleoperators," Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT, Cambridge, MA 1978.

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

[5] 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*, vol. 30, pp. 286-297, 2000.