Sylvain Bruni, Yves Boussemart, M.L. Cummings and Sergio Haro

Visualizing Cognitive Strategies in Time-Critical Mission Replanning

ABSTRACT

A post-hoc cognitive strategies visualization tool for operators in command and control environments, TRACS (Tracking Resource Allocation Cognitive Strategies) captures the critical steps performed in solving a resource allocation problem like those typically encountered in mission planning and replanning. TRACS presents a human operator's cognitive strategies in a two-dimensional space, using axes of automation levels and information types. TRACS has been successfully used on Tomahawk strike planning and path planning interfaces: the cognitive strategies visualizations created from mission planning scenarios displayed clear patterns of behavior that could be correlated to performance. Although an innovative way to understand and monitor human operators' strategies to solve multivariate, complex resource allocation problems, TRACS was previously limited to static environments. Thus, we developed an extension, TRACS 3D, which adds a third axis for time to the existing axes. A three dimensional representation of the operator's cognitive strategies, TRACS 3D attempts to capture the temporal aspects of human interaction in mission planning. However, an early pilot application of the tool to Tomahawk mission replanning problems showed strong limitations and cumbersome drawbacks. Thus, "TRACS 2.5D" was developed to address these drawbacks, and is a hybrid tool between the original two-dimensional TRACS and TRACS 3D,

INTRODUCTION

Missile strike planning is a complex example of multi-attribute resource allocation, where a set of resources must be paired with a set of missions or goals in a manner that meets constraints and achieves a certain level of quality. In terms of strike planning for Tomahawk missiles, our representative domain, the missiles and associated missions are characterized by a series of variables. A strike planner's task consists of making sure that no constraint on these variables is violated when a specific missile is assigned to a specific mission. In addition, the final solution, i.e. the set of mission/missile assignments, should be as "good" as possible, where "good" is often subjective or a dynamic reference that may change in the planning process.

Previous work (Bruni and Cummings 2005; Cummings and Bruni 2005; Marquez, Cummings et al. 2005; Bruni and Cummings 2006; Bruni, Marquez et al. 2006) investigated the creation of decision-support tools aimed at leveraging human-automation collaboration to enhance the quality of the Tomahawk mission planning process. Part of this work included the creation of a visualization tool, TRACS. This tool captures the cognitive strategies implemented by human operators while interacting with the decision-support systems. In these studies, it has been shown to be useful in identifying strategies, in correlation with performance data.

However, these implementations and uses of TRACS were limited to static environments, where the sensor data and problem constraints were not time-dependent. In most command and control systems, this is highly unlikely to be the case: replanning tasks might be needed at any time. In the missile strike example, the need for replanning might be triggered by the emergence of an unexpected target. Such an emergent target may be considered to be of the highest priority, hence requiring a strike plan revision in order to include this target in the strike. Such a situation is an undeniably time-sensitive matter, adding to the complexity of the task, as the operator must quickly come up with a feasible plan that includes the emergent target.

Previously versions of TRACS do not account for time, and fail to characterize time-pressure related course of action or changes in strategy, as all data was aggregated over time. This paper introduces new versions of TRACS aimed at capturing such information, applied to the missile strike example. Their benefits and drawbacks are discussed.

BACKGROUND

The matching task

A typical Tomahawk Land Attack Missile (TLAM) strike planning process involves a Strike Coordinator whose main task consists in pairing a set of pre-planned missions with missiles available aboard different launchers such as submarines or cruising ships. This constitutes a complex, multivariate resource allocation problem, where a human operator must not only satisfy a set of matching constraints, usually part of the rules of engagement (ROE), but also optimize the mission-missile assignments to minimize operational costs or enhance the quality of the overall plan. Generally it is left to the Strike Coordinator to manually assign missiles to missions, taking into account the different mission and missiles characteristics, as well as the constraints *du jour* included in the ROE.

Typical matching constraints include the navigation system (NAVSYS), the warhead, or the launch basket. In order to be a valid match, both the mission and the missile must feature the same characteristics in these domains. For example, in the case of NAVSYS, if a mission requires a Global Positioning System (GPS), then only missiles equipped with a GPS device can be assigned to that mission. If a missile's characteristics do not match those of the mission considered, then that missile cannot be paired with that mission. Such matching constraints are called "hard constraints".

Another set of constraints, dubbed "soft constraints" are less strict, and are aimed at optimizing the quality of the solution, i.e. the set of mission/missile matches. For example, each missile can be given a value based on the probability that its launch will be successful. Such probabilistic information is of primary importance to the Strike Coordinator, who may decide to only assign those missiles with a high probability of successful launch.

The StrikeView interface

In an effort to decrease Strike Coordinators' workload and improve the quality of the strike planning process, a decision-support system called StrikeView was developed (Bruni and Cummings 2005; Bruni and Cummings 2006). One particular implementation of StrikeView is shown in Figure 1. This interface allows the operator to solve the problem, i.e. build a set of mission/missile assignments, either manually or with the help of the computer.

Two tables at the top of the interface respectively list all the pre-planned missions with their characteristics (such as the target to destroy in the mission, the launch basket the route originates from, the required navigation equipment, the target priority, the required warhead) and the missiles with their characteristics (the launcher they belong to, the launch basket they are in, the navigation system they are equipped with, the warhead their payload carries, the probability of successful launch).

If the operator decides to manually solve the problem, she would simply select a mission from the mission table, then a missile from the missile table, and finally click on "Add Match" in order to add the mission/missile pair to the solution, or strike plan, which is summarized in the central table. In order to hasten the process and help the operator's decision-making, sorting and filtering features are implemented. Indeed, when a mission (respectively a missile) is selected in the appropriate table, the computer automatically filters out those missiles (respectively missions) that do not satisfy the hard constraints, such as navigation system, warhead, or launch basket, by graying them out.

The interface in Figure 1 also allows the operator to leverage the computational power of automation to search the solution space for assignments combinations. A heuristic search algorithm, called "Automatch", is embedded in this interface. Such algorithms follow simple

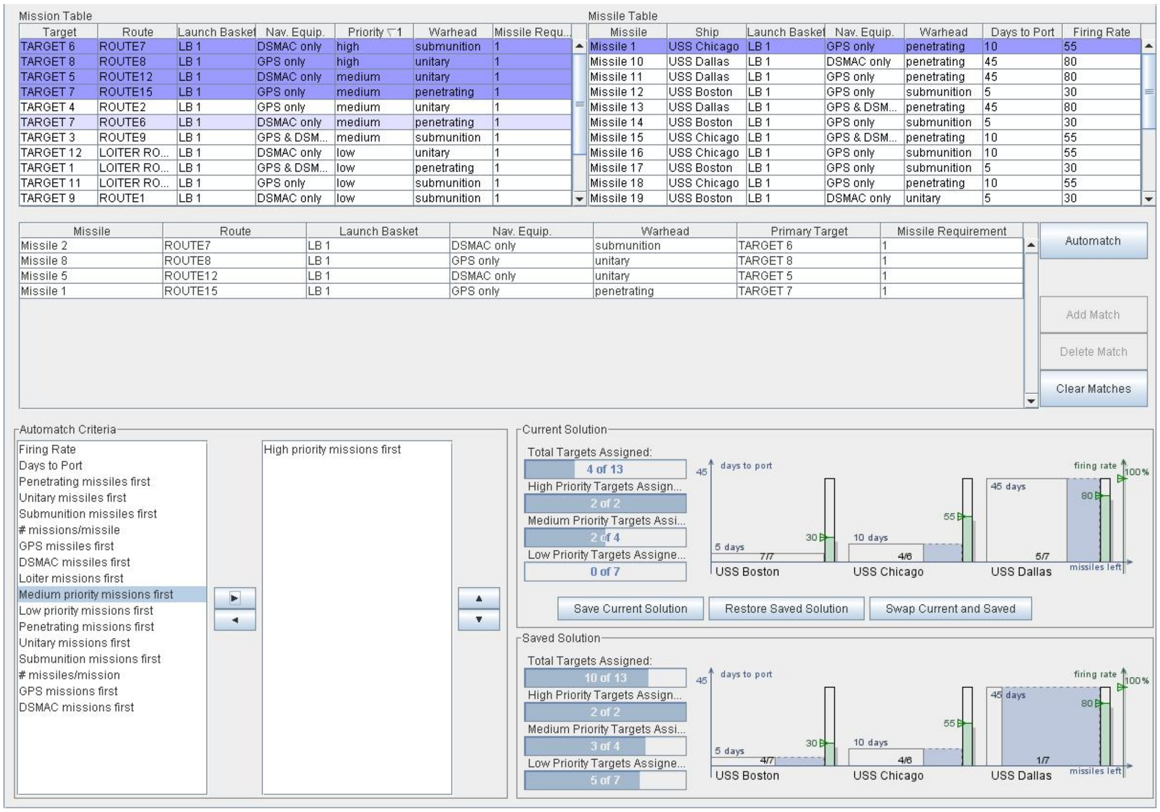

Figure 1 - StrikeView, matching interface

heuristics, or "rule of thumbs", in order to rapidly explore the solution domain. Although heuristic search algorithms may be sub-optimal or fail to find a solution, they are very rapid and easily understandable by a human operator. In the case of StrikeView, a collaborative search process can also be implemented: the operator selects an ordered list of search criteria (or heuristics) from the two columns at the bottom left of the interface, and then asks the computer to search the tree of assignments following these criteria. For example, the Strike Coordinator might have been informed that all targets requiring a penetrating warhead are critical and must be destroyed to the fullest extent. Hence, the Strike Coordinator might choose the criterion "Penetrating Missions First" at the top of the list of heuristics, so that the search algorithm first attempts to assign missiles to said missions, before considering other, non-penetrating, missions.

Additional decision-support tools are featured in this interface. Horizontal summary bars in the bottom center part display how many targets have been assigned in the current solution, with a breakdown by target priority. Also, a graphical display shows the impact of the soft constraints on the current assignments (Bruni and Cummings 2006).

The tracking tool: TRACS

Because comparing different interface implementations is difficult, a post-hoc visualization tool was created as a means to gather better insight about how the operators use these interfaces and how they impact overall performance (Bruni and Cummings 2006; Bruni, Marquez et al. 2006). The "Tracking Resource Allocation Cognitive Strategies" (TRACS) tool was designed to capture the cognitive strategies developed by a human operator interacting with a decision-support system in a multivariate resource allocation problem. In its original format, TRACS is a two dimensional diagram featuring a "Level Of Information Detail" (LOID) axis, and a "Mode" axis. The LOID axis categorizes the type of information in a hierarchical order (e.g. from low level sensor data, to high level aggregated information) used by the operator at each step of her strategy. The Mode axis regroups the different cognitive actions that operators can take while interacting with the interface, such as browsing or selecting data, filtering information, or backtracking.

A typical TRACS visualization is shown in Figure 2. At each step of the problem solving process, a circle is added to the diagram in the cell corresponding to the type of information used by the operator during that step, and the cognitive action implemented. Successive steps are linked together by a line. When the same steps (respectively links) are visited again, the width of the circles (respectively lines) is increased. The example of Figure 2 was built while an operator used the StrikeView interface. The operator's cognitive strategy included several repeated cognitive steps of combining "data items" and the "select" mode. It can thus be inferred that the operator proceeded with a manual matching strategy. The strong link between that step and the cognitive step that combines the same "data item" level with the "backtrack" mode suggests that the user was in a cognitive state of exploration of the solution space. Similarly, the upper part of the representation shows a classical pattern, but for the use of the automatch functionality. The operator started by choosing a specific heuristic for the search algorithm by selecting a criterion from a list, which is depicted by the circle in the cognitive steps combining the "individual criterion" LOID and the "select" mode. Then the operator launched Automatch, which was captured by the "individual criterion" LOID in the "automatch" mode. Finally the resulting solution was evaluated by the operator: it corresponds to the "group of matches" LOID at the "evaluate" mode. Typically, if the solution provided by the automation is not acceptable, the operator can either go through the Automatch loop again and select another heuristic, or try to tweak the solution manually. Extensive testing of this initial tool was performed with StrikeView, as well as with a traversal path planning interface (Marquez, Cummings et al. 2005). TRACS proved to be successful at identifying strategy patterns and strategies switches, and it was also possible to correlate the identified patterns with performance (Bruni, Marquez et al. 2006). This version of the TRACS tool and its application to the implementation of StrikeView is available for public use and download at http://web.mit.edu/aeroastro/www/labs/halab/media.html

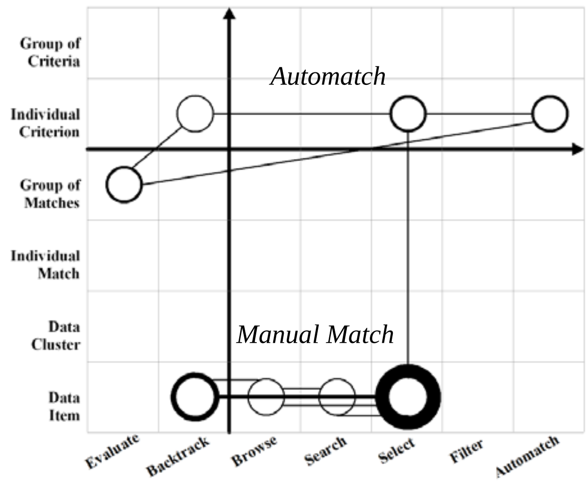

Figure 2 - An example of TRACS visualization

TRACS 3D

A main problem with the original version of TRACS is that its non-directed graph representation does not allow capture of temporal data. First, the link between any two states does not contain the information about which state came first. For example, in Figure 2, two equally correct interpretations of the graph exist. The first one would predict that the participant started by the manual matching strategy, followed by a strategy switch to an automated solution generation. The second interpretation would be the exact opposite, where the operator started by using automatch and then a switch to a manual strategy. In addition, the concept of aggregation where all visits of one cognitive step are combined and represented by the width of the lines or the circles, does not allow for temporal discrimination. The current version of TRACS does not capture time-induced behavioral changes, such as, for example, an increase in the use of certain cognitive steps as a temporal deadline approaches.

In order to capture the time aspect of the replanning situation, and therefore resolve the ambiguity described above, TRACS was augmented with a third axis: time. Instead of aggregating steps visits over time on the same circles or lines, by proportionally increasing their width, this implementation, called “TRACS 3D”, separates each step occurrence from the next, by displaying them as separate spheres and tubes along the time axis. Figure 3 displays an example of visualization of cognitive strategies tracking using TRACS 3D. This figure actually corresponds to the same case as that of Figure 4, which was obtained with the original two-dimensional TRACS. The sphere and tubes becomes larger every time the corresponding step is visited, similarly to the increase of width in two dimensions. Additionally, TRACS 3D offers the possibility to manipulate the point of view from which this three-dimensional construct can be viewed. Zoom-in, zoom-out and panning functionalities allows for total visualization control.

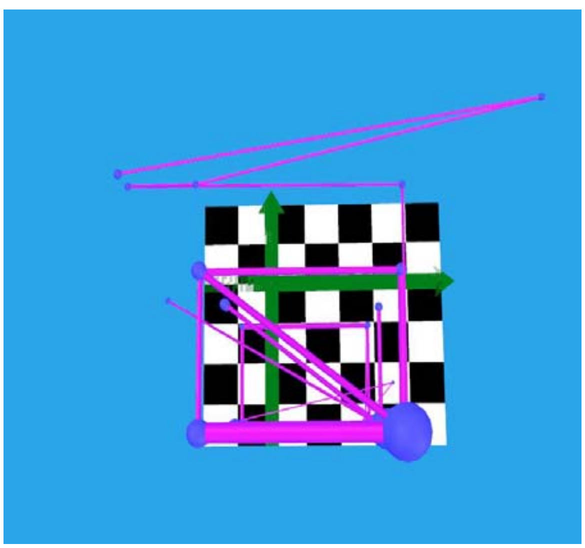

Figure 3 - TRACS 3D, view from the top

This visualization tool was evaluated by several subject matter experts, MIT graduate students and faculty, all with a background in interface design and human factors engineering. Although they all appreciated the novel visualization, they also all agreed that such implementation of TRACS in three dimensions was overly confusing and cumbersome to manipulate. Overall, it appeared that the 3D representation did not provide an effective way to visualize and extract human cognitive strategies in the time-pressured, replanning scenario.

Six distinct drawbacks of TRACS 3D were identified from our subject matter experts’ feedback.

1. **Loss of granularity and clutter**. It was noted that as the number of steps increases, their representations tend to come so close to each other that it makes it difficult to discriminate what sphere corresponds to what LOID, mode or time. The concentration of spheres is the result of the implementation of the time dimension in TRACS 3D. Indeed, in order to contain the overall graph in a reasonable size that is appropriately viewable on a regular 1280x1024 screen, the size of the unit time-increment is decreased every time full scale is achieved, which results in a periodical “shrinkage” over the time axis.

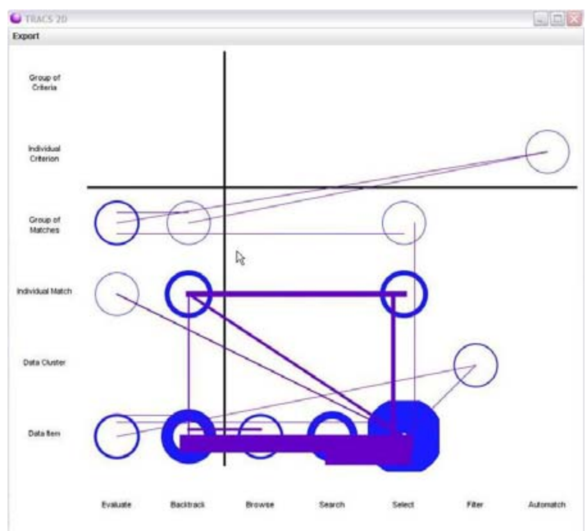

Figure 4 - The original TRACS

2. **Loss of 2D information (occlusion effect)**. At times, visualizing the cognitive strategies in three dimensions can be detrimental to full perception of the strategies: occlusion problems were identified in certain views, where some steps’ spheres would be hidden behind others (see Figure 3). Similarly, short

links would disappear behind or within spheres, thus providing an incomplete picture of the strategy. This occlusion effect triggered the need to manipulate the orientation of the construct to gain awareness of hidden paths.

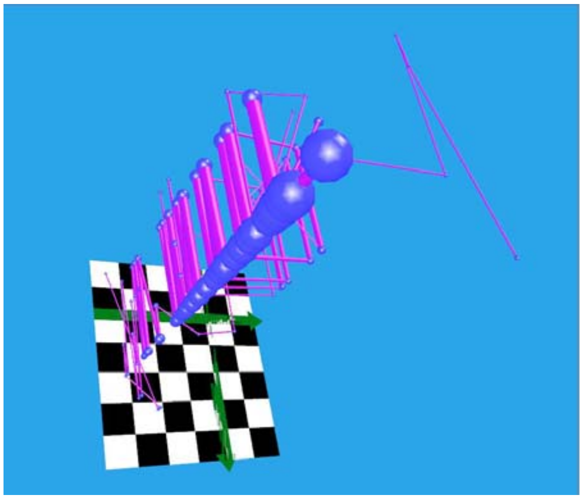

Figure 5 - TRACS 3D, rotated view

3. **Detrimental perspective (parallax effect)**. A typical pitfall of perspective representation is the parallax error, where closer but smaller objects appear identical to larger but farther other objects. In this case, the size of the spheres has a very precise meaning: the time aggregate of the number of visits in one step. The three-dimensional perspective may bias viewers in their evaluation of the size of the spheres, hence of the relative importance of some steps compared to others. This effect in three dimensions has been shown to decrease human performance in size and magnitude estimates compared to a two-dimensional case (Carswell, Frankenberger et al. 1991; St John, Smallman et al. 2000).This parallax effect also triggers the need to manipulate the orientation of the visualization to better compare spheres to one another.

4. **Difficulty to manipulate**. Another drawback of the TRACS 3D implementation is the difficulty of visualization reorientation. The multi-axis, mouse-controlled rotation system, coupled with approximate zooming features, make it a challenge to manipulate the observer viewpoint. This could cause spatial disorientation (Haskell and Wickens 1993).

5. **Difficulty to orient oneself**. Concurrent to the difficulty to manipulate the view, TRACS 3D fails to provide correct reference anchor points for an observer to efficiently assess their orientation on the three-dimensional space. In order to bypass the occlusion and parallax effects, extensive manipulation of the interface viewpoint is sometimes necessary. But since this task can be cumbersome to implement, we hypothesize that it draws significant cognitive effort, to the detriment of the user's spatial and situation awareness (Haskell and Wickens 1993; St John, Smallman et al. 2000).

6. **Lack of emergent temporal analysis feature**. Subsuming the previous points is the fact that the three-dimensional representation of time does not permit the desired temporal analysis of the operator's cognitive process. TRACS 3D does not efficiently capture and represent behavioral changes that would be caused by time pressure. One of the reasons is that the features may not be visible at all times due to occlusion, but another more significant reason is that the lack of referential on the time axis does not allow the estimation of the overall timescale of sequences of actions.

TRACS 2.5D

The next implementation of TRACS was developed with the purpose of representing the time component of the problem more effectively, while specifically addressing the drawbacks which caused the problems with TRACS 3D. Dubbed "TRACS 2.5D", this next version features a time axis, not as a third spatial dimension, but as an interactive feature of the tool. The temporal data is therefore represented as a dynamic property of the visualization as opposed to the spatial analogy used in the three dimensional case. As shown in Figure 6, TRACS 2.5D displays the regular, two-dimensional TRACS visualization, with the addition of a timeline at the bottom. Similarly to the widely used timelines features of any commercial, video or audio media player, the small cursor on the timeline can be dragged along the horizontal axis, in order to access specific moments of the scenario. The recording time is also displayed to the right, where the operator can also manually input a desired time to go to. On the timeline, red triangles are used as time stamps, to represent each step visit at the time they occurred in the scenario (see insert in Figure 6). Figure 6 to Figure 8 represent in chronological order several instances of TRACS 2.5D. Note that the cursor on the timeline goes from the left to the right, and that the TRACS construct is progressively building up. Furthermore, an additional feature of TRACS 2.5D is the ability to select a specific timeframe of interest, and only the states visited during that period will be displayed on the graphical representation. For example, Figure 9 shows this feature applied to the end of the scenario, and only the last few states are displayed.

This simple timeline feature in TRACS 2.5D addresses the six drawbacks of TRACS 3D in the following manner:

1. **Loss of granularity and clutter**. While states are linked to their respective time stamps, the temporal data does not influence the visual representation of the state transitions. Thus, there is no confusion in terms of which LOID a state represents at any time. Yet clutter is still an issue with TRACS 2.5D, as it is possible to have to enough states to make the timeline difficultly usable. However, as opposed to the case of TRACS 3D, the size of the time stamps do not increase with the size of the representation of a state. The maximum achievable timeline granularity is therefore significantly higher. Lastly, the ability to select a specific timeframe essentially removes the states that take place outside of the period of interest, therefore alleviating the display clutter issue.

2. **Loss of 2D information (occlusion effect)**. Since the TRACS construct is displayed in two dimensions, there is no occlusion as depicted in TRACS 3D. However, since the time component is not fully displayed at all times, but must be manipulated to reach specific instants, earlier constructs are hidden within latter ones. It is nevertheless hypothesized that the ease of use of the time slider allows for a better manipulation to reach these earlier constructs than the three dimension manipulation of TRACS 3D. Because the timeline is unidimensional, time stamp occlusions can only occur if two steps are taken very close to one another and the scale of the timeline is extremely small. Still, because the size of the step representation on TRACS increases accordingly, the information is not lost and can be accessed. In addition, the ability to select a specific time window for investigation allows disambiguating the process by only showing the states pertaining to that specific time interval.

3. **Detrimental perspective (parallax effect)**. A parallax error is a purely three-dimensional problem; it is therefore a non-issue in this two-dimensional representation.

4. **Difficulty to manipulate**. Slider manipulation is very common in computer environments, from web pages to multimedia applications. Computer users do not generally need any training or specific “skill”, hence manipulation is not expected to be an issue.

5. **Difficulty to orient oneself**. Because the user’s point of view does not change, re-orientation of the TRACS figure is not necessary. The time manipulation occurs on a one-dimensional axis, which is considered simple enough to provide

good time position awareness to the observer.

6. **Lack of emergent temporal analysis features.** As opposed to TRACS 3D, all timing data is visible at all times, and this allows for temporal emergent features to appear. For example, in Figure 6, one can see a zoomed-in insert of the time line, and time stamp clusters emerge. Similarly, it appears that the density of time stamps at the end of the run is significantly lower than at the beginning. Time stamps density thus constitutes a basic emergent feature of this display. This feature, coupled with the ability to visualize the time trends (either backward or forward) help make the spatio-temporal patterns apparent. On Figure 8, for example, one can identify three phases on the timeline. Firstly, a sequence of high frequency states shift followed by a short pause. Secondly, a single stamp preceding a long period of inactivity can be seen, with a few stamps at the end. By replaying the action, it appears that the state visited in the second phase is use of the Automatch function. It can be hypothesized that the longer pause is the time used by the operator to evaluate the complete solution obtained from automatch.

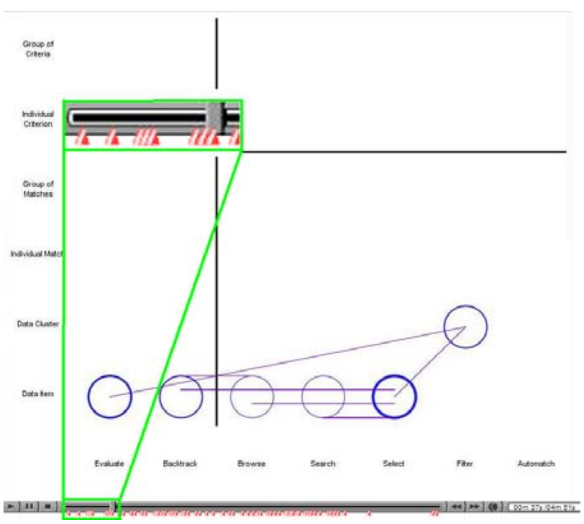

Figure 6 - TRACS 2.5D, beginning of scenario

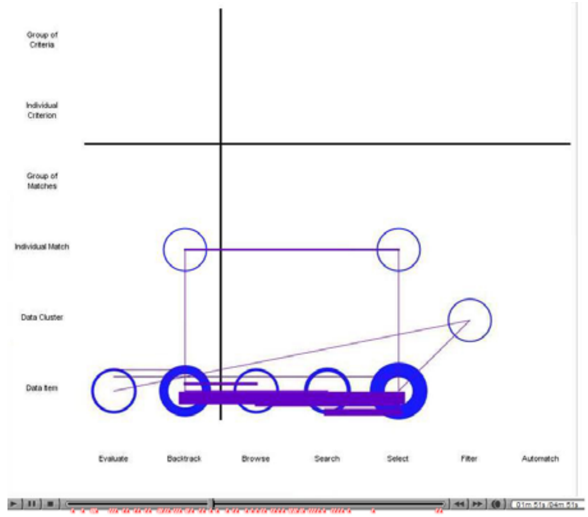

Figure 7 - TRACS 2.5D middle of scenario

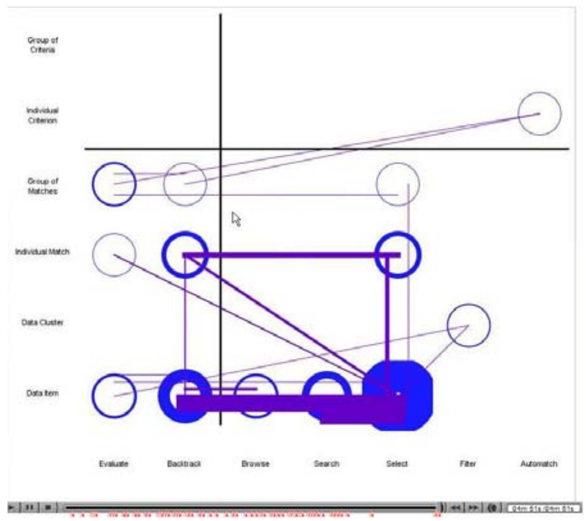

Figure 8 - TRACS 2.5D end of scenario

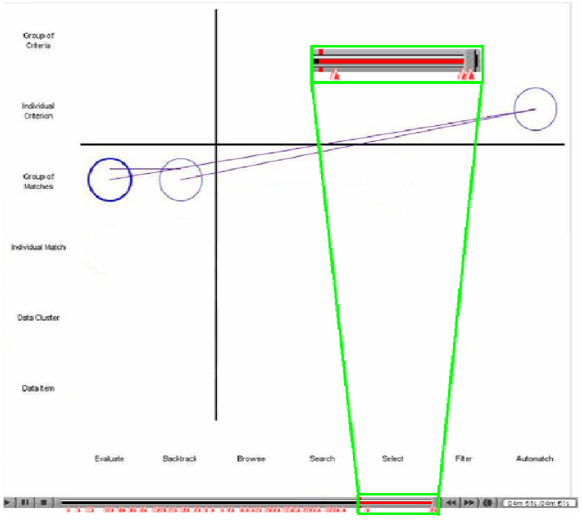

Figure 9 - TRACS 2.5D with timeframe selection

FUTURE WORK

While the current installment of TRACS permits tracking of an individual's cognitive strategies, future research is aimed at providing supervisors with a tool to monitor collective performance at the team-level. This involves two extensions of the TRACS tool.

The first extension is the ability to predict the performances of an individual based on the strategies exhibited by TRACS. Learning algorithms can provide the basis for pattern detection mechanisms. Distinctive cognitive patterns tend to indicate different individual strategies, thereby allowing the inference of a probable user performance level. Moreover, future versions of the tool could provide the same performance prediction but at a team level. This will take into account collaboration factors such as intra-team communication and distributed cognition.

CONCLUSION

The time-enabled implementations of TRACS, TRACS 3D, exemplifies the typical drawbacks of inappropriately incorporating three dimensions into interface design. TRACS 3D seemed to be the natural way to include time along a third spatial axis. However, the costs of such visualization, in terms of clarity and interpretability of the representation, considerably outweighed its usefulness. Rather, giving the user the opportunity to interact with the representation and control what they can see along one additional dimension, as implemented in TRACS 2.5D, is more efficient in displaying the impact of time on problem solving. Future research will explore further development of the timeline and its time stamps, as well as applications to team environments and real-time replanning.

REFERENCES

Bruni, S. and M. Cummings (2005). Human Interaction with Mission Planning Search Algorithms. Human Systems Integration Symposium, Arlington, VA.

Bruni, S. and M. L. Cummings (2006). Tracking Resource Allocation Cognitive Strategies for Strike Planning. COGIS 2006 - Cognitive Systems with Interactive Sensors, Paris.

Bruni, S., J. Marquez, et al. (2006). Visualizing Operators' Multivariate Optimization Cognitive Strategies. HFES 50th Annual Meeting, San Francisco, CA.

Carswell, C. M., S. Frankenberger, et al. (1991). "Graphing in depth: Perspectives on the use of three-dimensional graphs to represent lower-dimensional data." Behaviour and Information Technology **10**(6): 459-474.

Cummings, M. L. and S. Bruni (2005). Collaborative Human-Computer Decision Making in Network Centric Warfare. TTCP HUM TP7 Workshop on Aerospace Human Factors Issues in Network-Centric Warfare. Salisbury, UK.

Haskell, I. D. and C. D. Wickens (1993). "Two- and three-dimensional displays for aviation: A theoretical and empirical comparison." International Journal of Aviation Psychology **3**: 87-109.

Marquez, J. J., M. L. Cummings, et al. (2005). Collaborative Human-Computer Decision Support for Planetary Surface Traversal. AIAA Infotech@Aerospace, Arlington, VA.

St John, M., H. S. Smallman, et al. (2000). Designing for the task: Sometimes 2D is just plane better. IEA/HFES 2000 Congress, San Diego, CA.

ACKNOWLEDGMENT

This research was sponsored by the Office of Naval Research. 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.

Sylvain Bruni obtained a Master's Degree in Aeronautics and Astronautics at MIT, and a Master's Degree in Electrical Engineering and Computer Science from Supélec (France), both in 2004. He is currently a PhD candidate in the Humans and Automation Laboratory, in the Aeronautics and Astronautics Department at MIT. His research focuses on collaborative human computer decision making, human interaction with automation and complex systems, network centric warfare and the joint operation of heterogeneous agents.

Yves Boussemart graduated from McGill with a Bachelor of Computer Engineering in 2002 and obtained a Master of Engineering in the Center for Intelligent Machines in 2005. His research work was focused on gestural interfaces for collaborative work in virtual environments. Yves subsequently joined the Embedded Systems Lab at MIT, where he worked on safety critical real-time systems. He is now a PhD student in the Humans and Automation Lab and is working on a Collaborative Time-Sensitive Targeting research project.

Mary L. Cummings received her B.S. in Mathematics from the United States Naval Academy in 1988, her M.S. in Space Systems Engineering from the Naval Postgraduate School in 1994, and her Ph.D. in Systems Engineering from the University of Virginia in 2003. A naval officer and military pilot from 1988-1999, she was one of the Navy's first female fighter pilots. She is currently an Assistant Professor in the Aeronautics & Astronautics Department at the Massachusetts Institute of Technology. Her previous teaching experience includes instructing for the U.S. Navy at Pennsylvania State University and as an assistant professor for the Virginia Tech Engineering Fundamentals Division. Her research interests include human interaction with autonomous vehicle systems, humans and automation, decision support, human-computer interaction, and the ethical and social impact of technology.

Sergio Haro is pursuing his Bachelor of Science in Electrical Engineering and Computer Science at MIT.