2006 CCRTS

The State of the Art and the State of the Practice

Structure Mapping in Visual Displays for Decision Support

Cognitive Domain Issues, C2 Concepts and Organizations, C2 Analysis

Kevin Burns (POC) & Craig Bonaceto
The MITRE Corporation
202 Burlington Road
Bedford, MA
01730

781-271-8762
Fax: 781-271-2101
kburns@mitre.org

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Structure Mapping in Visual Displays for Decision Support	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **The MITRE Corporation,202 Burlington Road,Bedford,MA,01730**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**31**	

Structure Mapping in Visual Displays for Decision Support

Kevin Burns & Craig Bonaceto
The MITRE Corporation
202 Burlington Road
Bedford, MA
01730

Abstract

This paper adopts a cognitive theory of analogy and applies it to the design of decision support systems. The approach, called structure mapping, captures the conceptual structure of a C2 problem in the graphical features of a visual display. The result is a visualization that supports decision making better than standard mappings of geospatial information. As an example we present a system designed to support weapon-target pairing for time-sensitive targeting. This C2 system in particular, and structure mapping in general, are useful because solving a problem like weapon-target pairing requires reasoning about probabilities, priorities and other parameters that are not shown in standard mappings. Thus the contribution of this paper is threefold in: (1) reviewing the theory behind our structure mapping approach; (2) presenting a system designed in accordance with this approach; (3) discussing how the same approach can be used to design intuitively informative displays for other C2 systems.

1. Introduction

The practice of systems engineering often focuses on the technology by which visualizations and other machine interfaces are made, rather than the psychology by which decisions and other human inferences are made. The result is that users may not use the systems – or worse, that performance may actually be degraded by the systems that were intended to support decisions (Cummings & Mitchell 2005).

The problem, we think, is that computerized systems are designed without adequate attention to cognitive theories. In short, a systems engineer must have a thorough understanding of what needs to be supported and where it needs to be supported and why it needs to be supported in order to design a decision support system (Burns 2004). In this paper we propose a cognitive theory-based approach that can be used to improve support systems in command and control applications. We also present a visual system designed by this approach to support targeteers in weapon-target pairing for time-sensitive targeting.

2. Structure Mapping

Our approach to designing decision support systems is founded on a cognitive theory of analogy called *structure mapping*. In structure mapping (Gentner 1983; Gentner & Markman 1997), a body of knowledge in one domain (D_1) is mapped to another body of knowledge in another domain (D_2), $D_1 \rightarrow D_2$, based on common structure(s) shared by the two domains. A simple example is the statement, "an electric battery (D_1) is like a reservoir (D_2)", which allows a person who understands reservoirs (D_2) to apply this knowledge to batteries (D_1). Notice that D_2 is not necessarily visual, as in metaphorical analogies like the above simile where both D_1 and D_2 are verbal statements. However, D_2 can be a visual display and D_1 can be a complex problem (in C2) – and in that case, which is our focus here, the *structure mapping* can be seen as drawing a *visual analogy*.

The notion of structure mapping is implicit in standard mapping where geospatial features of a land, like roads and forests, are mapped to corresponding features of a map, like lines and shading. Such maps are useful because they allow decision makers to reason about complicated situations using simplified representations. But many C2 decisions require reasoning about attributes and relations that cannot be represented by the features of standard mappings. The problem of weapon-target pairing in time-sensitive targeting is a case in point. Although standard mappings can show locations, distances and other geospatial information, they cannot easily show priorities, probabilities and other parameters of the problem or its solution.

The theory of structure mapping (Gentner 1983) was developed to explain how people form abstract analogies between things when there appear to be no surface similarities between those things. For example, electric batteries are typically small and cylindrical while hydraulic reservoirs are typically large and complexly shaped, so there is little similarity on the surface. But at a deeper level the two share a common function, which entails a storage capacity and release mechanism to accomplish controlled flow of current. From a practical perspective, the advantage of analogies is that they allow people to solve problems in domains that they do not understand well (like the electric domain of batteries) by relating them to domains that they do understand well (like the hydraulic domain of reservoirs).

In this paper we are concerned with structure mapping of C2 problems and solutions to visual displays in support systems. The basic issue is that, while standard mappings can provide analogical representations of C2 domains, the attributes and relations depicted in such maps are often not well suited to solving the C2 problem at hand. In these cases a better mapping would be a structured mapping that does depict the attributes and relations of the problem and its solution. In short our approach is to map a C2 problem to a visual display that makes the problem easier for a human to solve and easier for a human to see how a system solves it – thereby allowing the human to better understand a system's solution and develop his own solutions for cases where the system's solution may fall short.

Below we discuss a prototypical example in the domain of time-sensitive targeting.

3. Weapon-Target Pairing

3.1 Time-Sensitive Operations

The challenge of *time-sensitive targeting* arises in combat situations where the locations and priorities of targets are changing with time. In fact this is the case in all targeting, although sometimes the changes in targets are slow with respect to the speed of orders and assets, and in such cases strike operations can be planned as if the targets were fixed. Here we are concerned with situations where pre-planned targets are essentially fixed but where emergent targets (with high priority and time-constrained threat or vulnerability) may arise during the mission. One example is the case of scud missile Transporter Erector Launchers (TELs) in the Gulf War. These were time-sensitive targets because they were important to destroy and because they had a window of vulnerability of around an hour, which is much shorter than the day(s)-long planning cycle of Air Tasking Orders.

The basic problem in time-sensitive targeting, once targets have been identified, is one of asset allocation via *weapon-target pairing*. That is, given a constrained set of assets, the question is: Which weapons (assets) should be diverted from their originally scheduled targets to attack time-sensitive targets in order to optimize the overall effectiveness of the mission? In fact this problem entails two problems. The first problem, which is called the *priority problem*, is to establish numerical priorities for all targets and assets. This problem is difficult because priorities may change with time as targets are killed and/or assets are lost. The second problem, which is called the *pairing problem*, is to optimize the assignment of assets to targets, given an assumed set of assets and targets along with priorities and other parameters like target kill probabilities and asset loss probabilities.

The *priority problem* is actually the harder of the two, especially in joint operations where allies may not be in complete agreement, e.g., I may think that my assets and targets are more important than yours. Plus, even if allies agree, the priorities and other parameters such as probabilities of target kill and asset loss are subject to many ambiguities and assumptions. Nevertheless, previous designs for support systems (Pedersen et al. 1999) have assumed that the *priority problem* has been solved, and here we will assume the same in our system – leaving this problem for future research. However, unlike previous systems, our proposed system for solving the *pairing problem* recognizes current limits in solving the *priority problem* – and our design includes visualization capabilities that can help targeteers deal with the fact that they must often solve *pairing problems* without solid answers to *priority problems*.

Focusing on the *pairing problem*, support systems can be divided into two types. One type is an *optimization* system that finds overall-optimal solutions for weapon-target pairings, given a set of input data. The other type is an *organizational* system that tracks orders and approvals in the chain-of-command for target identification and prosecution. Here we are concerned with *optimization* systems, which solve computational problems, as opposed to *organizationa*l systems, which solve communication problems.

However, our proposed system for solving *optimization* (computational) problems can also help in solving organizational (communication) problems since the system provides a visual display of *what* assets should be assigned to what targets as well as the underlying rationale for *why* these pairings should be made. This dual display of *what* and *why* facilitates communications by illustrating solutions to the *pairing problem*, including parameters relevant to the *priority problem*, in a display that can be understood at a glance by all levels of command.

3.2 Trade-Off Calculations

Focusing on the need for an *optimization* system to support users in solving *pairing problems*, the C2 challenge is to assign assets (weapons) to targets in a way that optimizes the overall effectiveness of an operation – given input numbers for the various parameters that affect the output answer. Here we consider the case of N assets and N targets, where each asset can attack only one target. This one-on-one context often applies to time-sensitive targeting where the window of vulnerability for each target is small. The NxN analysis is easily extended to NxM cases (N≠M) since these cases can be cast in the form of NxN problems by creating fictional assets and targets (with zero-valued priorities) to make M=N. In that case, if a fictional asset or target is part of the optimal solution, then the associated target (paired with a fictional asset) is not prosecuted or the associated asset (paired with a fictional target) is not diverted.

The complexity of the *pairing problem* can be characterized by the number of possible solutions, which is N-factorial (N!). For example, with N=10 there over three million solutions, which makes a targeteer's dilemma of finding the optimal solution seem like searching for a needle in a haystack. Therefore engineers have implemented sophisticated procedures like an auction algorithm (Bertsekas 1992) in designs for decision support systems (Pedersen et al. 1999). As input, these systems take a listing of targets and assets along with numbers for priorities, probabilities and other parameters. As output, the systems give the optimal solution, i.e., the needle in the haystack.

More formally, the pairing problem is concerned with maximizing a *value function* over all possible (N!) sets of N one-on-one (asset-to-target) pairings. For each individual asset-target pair, the value function can be written as follows:

$$G = (U_T * P_T) – (U_O * P_O) – (U_A * P_A)$$

where U_T is the utility (priority) of a time-sensitive target (T) to which an asset (A) can be paired and P_T is the probability that asset A will be effective in destroying target T. Thus $U_T * P_T$ is the *expected utility* (score) of pairing asset A to target T. Similarly, $U_O * P_O$ is the expected utility (cost) of diverting asset A from its originally scheduled mission, hence this term is subtracted in the equation. Finally, $U_A * P_A$ is the expected utility (risk) of losing asset A due to threats at or along the way to/from target T, which is also subtracted from the score. Then, using this value function along with input numbers for all the U's and the P's, the auction algorithm finds the one-on-one set of N pairings (N A's to N T's) that maximizes the overall gain G.

3.3 Previous Visualizations

Here we do not attempt an exhaustive review of all visualizations that have been designed to support solving the pairing problem or other problems like it in command and control. Rather, we refer to others (Pederesen et al. 1999) that have summarized the state of the art and note that the two main types of visual displays are geospatial mappings and numerical tables. Our main point here (discussed below) is that neither of these visual devices is very effective for solving the problem of weapon-target pairing.

Geospatial mappings are ubiquitous in command and control, and for good reason since commanders must often reason about movements along and above terrain. In the case of time-sensitive targeting, such maps can be very useful for planning routes of assets to targets. But for the specific challenge of the *pairing problem*, as discussed above, such maps are almost useless because they do not represent the basic structure of the pairing problem – which is an NxN matrix where the goodness of any set of N pairings depends on probabilities and utilities.

Similarly, numerical tables are also ubiquitous in C2 because they are easy to construct and efficient in organizing large volumes of data. But the numbers in such tables are usually harder for people to read than graphics by which the numbers might be displayed. Moreover, in the case of weapon-target pairing, the numerical tables in previous systems did not capture the matrix structure of the *pairing problem* as discussed above. Instead, numerical tables have been used mostly to display data for *drill down* purposes, i.e., to allow commanders to get more details about data associated with the objects that they see on their geospatial mappings.

As such, the geospatial mappings and numerical tables used in previous system designs are not the best ways to display data in accordance with a structure mapping approach – at least not for the specific structure of the *pairing problem*. Below we present our design, which is different because it is based on structure mapping.

4. Pairing Pictures

4.1 Structure-Mapping Visualization

Mathematically speaking, the problem of finding the needle in the haystack is quite hard (see Section 3.2) since it scales with N!. Psychologically speaking, the problem is that system users have been reluctant to accept the needle that a system finds (Pedersen et al. 1999), for two reasons. First, targeteers realize that there are many assumptions and uncertainties that go into the system's calculations, yet the system has not shown or told them these. Second, targeteers realize that in fact the problem is not one of a single needle and a bunch of hay, i.e., there may be many sub-optimal solutions that are almost as good as the so-called optimal solution that the system finds. Moreover, putting these two concerns together, targeteers realize that the system's so-called optimal solution may in fact be sub-optimal if all of the system's assumptions and input numbers are not just right – and in reality the system's assumptions and inputs are often far from right.

In light of these concerns, targeteers are reasonable in not trusting the system's needle when it is given to them blindly, or when it is given to them in the context of geospatial mappings and numerical tables that say nothing about how the needle was found by the system or why it is considered optimal by the system. And that is why we developed a decision support system using structure mapping as the guiding approach.

In structure mapping, the design challenge is to map the computational structure of the *pairing problem* to the graphical structure of a visual display, so a targeteer can understand *what* the system recommends as well as *why* the system recommends it. And from this perspective, the structure of interest is an NxN matrix of assets and targets – which forms the foundation of our design (Figure 1).

Besides this matrix, the other structure that is central to the pairing problem is given by the three terms on the right hand side of the value function (Section 3.2) – as well as the fourth term, which is the gain (G) of pairing a particular asset to a particular target. This structure exists in each asset-target cell of the pairing matrix (Figure 1), and it can be represented by a mapping of values to colors. Our scheme uses the convention of Red for enemy (target) and Blue for friendly (asset) along with Gold to denote the cost of diverting an asset from its originally scheduled target. Thus the value function can be written as follows:

Black (Pairing Gain) = Red (Target score) – Gold (Divert Cost) – Blue (Asset Risk)

Then in a colored matrix, the mapping we call *Pairing Pictures* (Figure 1) simply shows the value function using colored bars inside matrix cells.

In this form the problem is readily seen as one of assigning each target (row) to an asset (column), one-on-one such that each target and each asset is paired only once. The overall-optimal solution is the one that achieves the highest sum of N black bars (pairing gains), as illustrated in a *Solution Summary* (Figure 2). Here possible solutions are numbered in order of the total pairing gain, for example solution 1 is the optimal solution that includes the three asset-target cells labeled 1 in the matrix (Figure 1). This allows the targeteer to see how the optimal solution compares to other possible solutions, and to exercise judgments in order to overcome system limitations, as discussed below.

4.2 Example Situations

For example, referring the Solution Summary in Figure 2, a commander's guidance may include the desire to minimize Divert Cost (Gold). Since the system does not know this, its optimum solution 1 is actually sub-optimal compared to solution 3, which is the solution that the targeteer would choose in order to best reflect his commander's guidance. Of course the same result could be obtained if the system were given appropriate weighting factors to reflect the commander's guidance, but that is the basic problem – i.e., that that the commander's guidance and appropriate weighting factors often cannot be established a priori.

In short, the user's main job is one of exercising judgment over a simplified system, and to do so a targeteer must know both the results (*what*) and reasons (*why*) behind the system's solution. Our visual displays in Figure 1 and Figure 2 are intended to give the targeteer a better feel for this *what* and *why*. A demo version is available online at: http://mentalmodels.mitre.org.

Similarly, referring to the Pairing Picture in Figure 1, a targeteer may know that asset C's effectiveness against target α is sensitive to weather conditions, which are not reflected in the system's input numbers for the probability that the asset will destroy the target. Here again the targeteer needs to exercise his expert judgment in order to overcome system limitations. In this case the targeteer may choose to avoid a solution that includes the cell α-C, e.g., he may select solution 2 as being the best because he can see from the Solution Summary (Figure 2) that solution 2 is just as good as solution 1 in the system's eyes – and it is better in his own eyes since it avoids the weather concern.

In the same example, the targeteer may also "think outside the box" as a result of his being able to "see inside the box" of the system (matrix). For example, even if the targeteer accepts solution 1 as being the best, he could see at a glance that almost all of the gain comes from two of the three pairings in solution 1, i.e., asset C against target α and asset B against target γ. In the cell for asset A against target β, which is part of solution 1, the tradeoff between positive factors (Target Score in Red) and negative factors (Divert Cost in Gold and Asset Risk in Blue) is almost balanced such that the net Pairing Gain in Black (for pairing asset A to target β) is almost zero. Thus, even though the action to send asset A to target β is part of the so-called optimal solution developed by the system, the targeteer might reasonably choose not take this action and instead only send assets C and B to targets α and γ, respectively.

Besides this Pairing Picture that focuses on mission *gains* (Figure 1), our system design includes another Pairing Picture that focuses on mission *time* (Figure 3). This picture is an animated display in the same matrix format as Figure 1 but instead shows how each asset's Time on Target (TOT, in Blue) compares to each target's Window of Vulnerability (WOV, in Red), for every possible asset-target pair.

This display allows a targeteer to overcome another system limitation, namely an assumption associated with the probability that an asset will arrive in time to destroy a target. In current systems this probability is assumed to be either one (if TOT is inside WOV) or zero (if TOT is outside WOV), which is obviously an oversimplification. In light of expert knowledge about this limitation in system calculations, and given the Pairing Pictures in Figures 1-3, a targeteer would note that solutions 1 and 3 both involve pairing asset C to target α for which the TOT is very close to being outside the WOV. Since this is a "fragile" pairing, which is sensitive to system assumptions, the targeteer may prefer solution 2, which is more "robust" in that the three cells of solution 2 all have TOT well within WOV. [Like the case of commander's guidance for Divert Cost discussed above, more complicated calculations could be developed in more advanced systems to obtain a more realistic probability distribution. But these calculations could also be illustrated in a similar TOT/WOV matrix display that would also aid users.]

Also, besides the *output* that is graphically illustrated in these Pairing Pictures, our design provides for graphical illustration of key *input* – namely the numerical priorities for assets and targets. Recall that these numbers, which are input to the *pairing problem*, are output from the *priority problem*, which is often not well solved. While further research may improve efforts to solve the *priority problem*, the fact is that numerical priorities for targets and assets will always be uncertain so a key task for the targeteer is (and will remain) to judge the sensitivity of pairings to uncertainties in priorities.

To address this need our system provides two things. First, we provide a graphical illustration of Adjustable Assumptions (Figure 4) so the user can see what priorities the system is assuming in its calculations. Here, previous system designs (Pedersen et al. 1999) did not allow the user to see the system's assumed priorities at all, let alone in a graphic format. Second, our system allows the user to adjust these assumptions via a graphical interface (sliding the bars) and then recalculate solutions in Pairing Pictures – which can then be compared in Solution Summaries.

This allows the targeteer to perform *sensitivity studies* in order to find and choose solutions that are robust with respect to uncertainties in the inputs. For example, if the priority for target α is known to be particularly uncertain then the targeteer can try two cases with bounding numbers (high and low) for the priority of target α and compare the solutions for these two cases. In this way the user can see what system solutions are most robust with respect to the underlying uncertainty – and select a solution accordingly.

Finally, an additional feature of our system is that the level of automation is completely adjustable by the user, from fully manual to fully automatic. That is, besides computing and displaying optimal and near optimal solutions, the system also allows the user to develop partial or fully manual solutions by selecting X individual cells (where $X \leq N$) while the system grays out the cells that are excluded by the user's selections.

4.3 Human-System Validation

Ideally, a system design would be tested to ensure that it provides benefits. But this is much easier said than done because testing must balance rigor and relevance. On the one hand, rigorous testing must be accomplished under controlled conditions – which usually requires scenario simplification in the lab. On the other hand, relevant testing must assess system and user performance with respect to the complex conditions in the field. The upshot is that system testing is often one or the other: either rigorously performed in the lab but lacking in relevance; or relevantly performed in the field but lacking in rigor.

In the case of our system, field tests could be done to measure if the system improves decision making. But it is not clear how performance could or should be measured, since the system was designed to help targeteers deal with ambiguities and uncertainties that cannot be captured and controlled in testing. In short, the system was intended to help users deal with situations that cannot be anticipated in experiments and hence cannot be assessed in experiments. This Catch-22 limits the conclusions that can be drawn from field testing to informal observations like "users seem to like it" and "combat seems to improve". Thus, while such testing is possible and may be performed for our system, we have not done it yet and we are pessimistic about what it could prove one way or the other.

A more rigorous testing of our system is possible in the lab, but here too we cannot test the main point – which is how well the system allows targeteers to solve realistic problems. The best we can do is to create artificial problems in a fixed problem space and then generalize the findings to more realistic cases of interest. In this light our validation has thus far been limited to answering two questions: (i) *How hard* is the pairing problem, i.e., to find reasonable solutions in a fixed problem space? (ii) *How well* do our Pairing Pictures help users, i.e., to find reasonable solutions in a fixed problem space?

To answer these questions we created a lab game called *TicTac Tank*, which can be found and played online here: http://mentalmodels.mitre.org. This game ignores the colored components of the value function (see above) and focuses on the net gain (black bar) in Pairing Pictures. To start, the player is dealt a distribution of black bars in an NxN matrix. The object of the game is to find the best solution (highest sum of N bars) as fast as possible.

To answer the question of (i) *How hard*, we generated various distributions for the bars among the cells and computed solutions in two ways. One way was via the auction algorithm that gives only one solution, i.e., the optimal solution, and the other way was via exhaustive enumeration of all N! possible solutions. The latter allowed us to assess *How hard* it was to find a solution that was optimal or near optimal. What we found was that there were often many solutions that were equally optimal or nearly optimal and hence that solving the pairing problem was not really one of finding a "needle in the haystack". It was more like finding a "good patch of hay".

To answer the question of (ii) *How well* our Pairing Pictures help people solve the pairing problem, we had players develop manual solutions using the matrix display (like Pairing Pictures) and we compared their manual solutions to the optimal solutions. The results showed that players developed a manual solution in about N seconds for a given NxN problem, and that their manual solution was within about 10% of the optimal solution in the hundreds of cases tested. Further details are not reported here because interested readers can try the game and see for themselves – that finding a "good patch of hay" is as easy as this: just pick a high bar (cell) in the matrix; then of the remaining cells (some of which are excluded by the first pick) just pick a high bar; and continue like this until a complete set of N bars has been picked.

The result of this "greedy grabbing" approach is almost always close to the optimal solution obtained by the "auction algorithm" simply because: if the matrix only contains a few big bars then the greedy grabber will find them; and if the matrix contains a lot of big bars then it does not matter much what bars are picked.

In short, our analysis and experiment showed that: (i) the pairing problem was not as hard as engineers had previously thought, i.e., it was not like finding a needle in a haystack of N! possible solutions, and (ii) a greedy grabbing algorithm, like the one that people employ when the problem is illustrated in the matrix structure of Pairing Pictures, is effective for finding near-optimal solutions in a way that scales with N rather than N!.

These insights are important because they suggest that the basic problem of weapon-target pairing is not that hard when it is cast in the form of our Pairing Pictures. Moreover we would argue that what makes the problem hard is the need for exercising expert judgments in order to overcome the limits of system assumptions and input numbers. As such, we think that previous designers missed the mark in focusing on support systems that compute optimal solutions without giving users graphical illustrations of the system solutions or graphical interfaces for developing manual solutions.

In short, while we have performed only economical and informal lab tests using a toy game, we believe that our analyses and experiments provide reasons to believe that our Pairing Pictures can be effective as a support system for weapon-target pairing in time-sensitive targeting. This is because the system gives users an informative and interactive display of system results (*what*) as well as reasons (*why*), as discussed elsewhere (Burns 2004). We also believe that the underlying approach of structure mapping can be effectively extended to other command and control problems, as discussed below.

5. Other Systems

As discussed above, the basic approach of structure mapping starts by analyzing the computational structure of a problem – and ends by representing this conceptual structure in the graphical structure of a visual display. The design process is still as much art as science because a given problem structure may be mapped to different visual displays. But a formal focus on problem structure does offer some science to guide designs – and this is the main value of a structure mapping approach.

In a nutshell, the idea is to identify *conceptual* attributes and relations in the structure of C2 problem, and then to illustrate them with *graphical* attributes and relations in the structure of a visual display. For the case of weapon-target pairing, the *conceptual attributes* are assets and targets, and the *conceptual relations* are the value function and pairing possibilities. The *graphical attributes* in our system design are colors and sizes (of bars), and the *graphical relations* are the matrix and the bar heights inside the cells.

As one guideline, the mapping should be both *conventional* and *consistent*. By conventional we mean that the graphics should exploit existing conventions, like the military convention for Red and Blue. By consistent, we mean that graphical features should be used to distinguish between conceptual features. For example, our Pairing Picture (Figure 1) uses vertical bars to represent expected utilities, where vertical is a graphical feature. Then to represent priorities, which are distinctly different from expected utilities, we use horizontal lines in our display of Adjustable Assumptions (Figure 4). The point is that the two sorts of bars lie in conceptually different dimensions and so they should be mapped to graphically different dimensions, like vertical versus horizontal. The structure that is common to both priorities and expected utilities, which is represented by the colors (Red and Blue), applies to both the horizontal and vertical bars.

As another guideline for system design, the display should be both *informative* and *interactive* (Burns 2004). A display is informative if its graphical structure is analogous to the conceptual structure of the problem and solution, as discussed above. However it is also helpful if the display is interactive because this allows the user to explore and digest the structure in an active manner, for example by developing manual solutions that can be compared to system solutions. Other interactive features of our design include the graphic display of Adjustable Assumptions and ability to perform sensitivity studies that can then be compared in Solution Summaries.

Another example of a system designed with structure mapping is a colored calculator to assist users in Bayesian inference (Burns, in press) – which is a key task in data fusion for target identification and other combat/intel problems, as discussed elsewhere (Burns 2004). Our support system for this *probability problem*, called Bayesian Boxes, was developed along the same guidelines as Pairing Pictures (discussed above) – by mapping the conceptual structure of the probability problem and its solution to the graphical structure of an informative and interactive display (Burns 2006). The system was shown in experiments to improve human judgments in Bayesian inference (Burns in press), and this lends further credence to structure mapping as a useful approach for designing decision support systems.

6. Conclusion

In this paper we proposed a formal approach to system design based on the cognitive-scientific theory of structure mapping. We applied this approach to the C2 problem of weapon-target pairing in time-sensitive targeting, and designed a system of *Pairing Pictures* that gives targeteers visual advice on *what* they should do, as well as visual reasons *why* they should do it. This design was shown to have advantages over previous designs that did not use structure mapping or any other cognitive theory to guide their development. We believe that our structure mapping approach is applicable to other C2 problems, and we briefly discussed another system called *Bayesian Boxes* that has been successfully designed and tested by the same approach. In current work we are extending the approach to additional problems, in order to further refine and apply structure mapping to the design of decision support systems in command and control applications.

7. References

Bertsekas, D. (1992). “Auction Algorithms for Network Flow Problems: A Tutorial”. *Computational Optimization and Applications* 1, pp. 7-66.

Burns, K. (in press). “Dealing with Probabilities: On Improving Inferences with Bayesian Boxes”. In Hoffman, R. (Ed.) *Expertise Out of Context*. Mahwah, NJ: Lawrence Erlbaum.

Burns, K. (2006). “Bayesian Inference in Disputed Authorship: A Case Study of Cognitive Errors and a New System for Decision Support”. *Information Sciences*.

Burns, K. (2004). “Painting Pictures to Augment Advice”. *Proceedings of the 7th International Conference on Advanced Visual Interfaces*, Gallipoli, Italy, May 25-28.

Cummings, M., & Mitchell, P. (2005). “Managing Multiple UAVs through a Timeline Display”. *Proceedings of AIAA Info Tech*, Arlington, VA.

Gentner, D. (1983). “Structure Mapping: A Theoretical Framework for Analogy”. *Cognitive Science* 7, 155-170.

Gentner, D., & Markman, A. (1997). “Structure Mapping in Analogy and Similarity”. *American Psychologist* 52(1), pp. 45-56.

Pedersen, D., Van Zandt, J., Vogel, A., & Williamson, M. (1999). “Decision Support Systems Engineering for Time Critical Targeting”. *Proceedings of Command and Control Research and Technology Symposium*, Newport, RI.

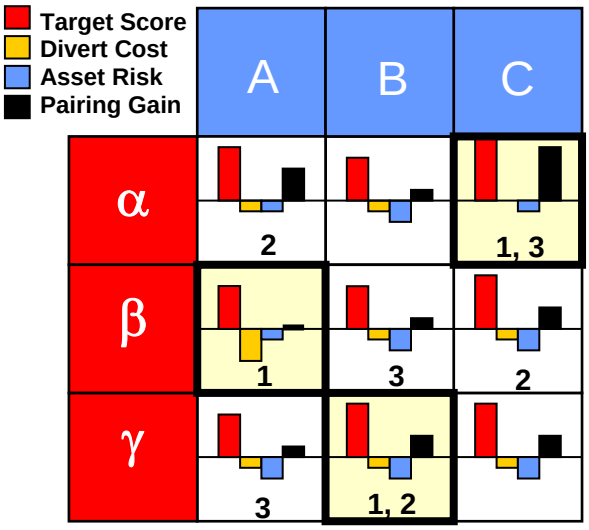


Figure 1: A support system called Pairing Pictures.

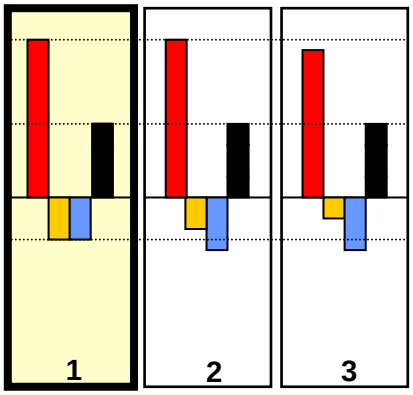


Figure 2: A Solution Summary for the Pairing Picture shown in Figure 1.

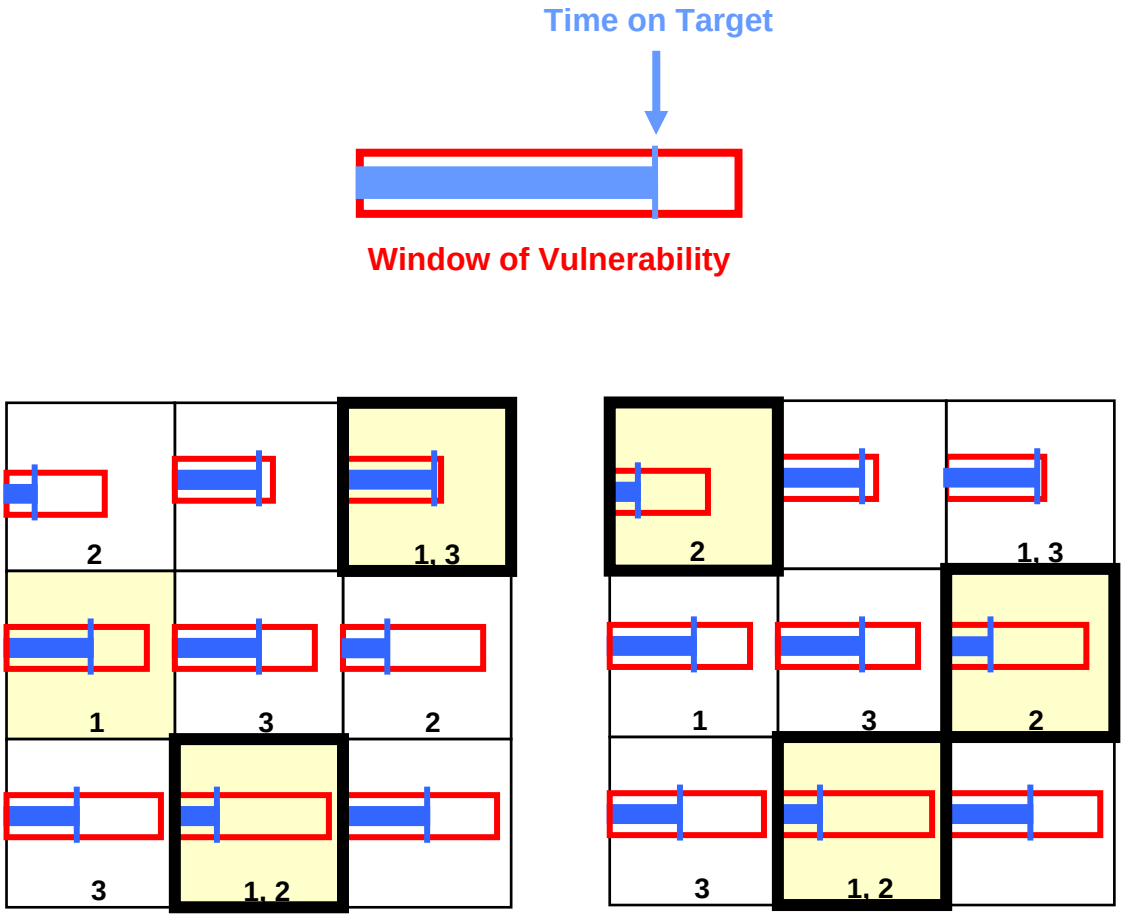


Figure 3: A comparison of timing windows for two solutions, number 1 (on left) and number 2 (on right).

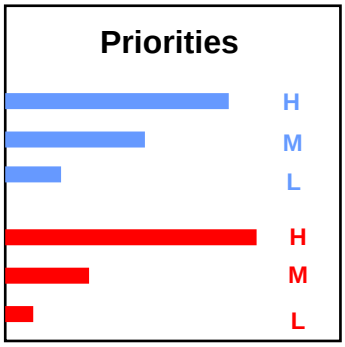
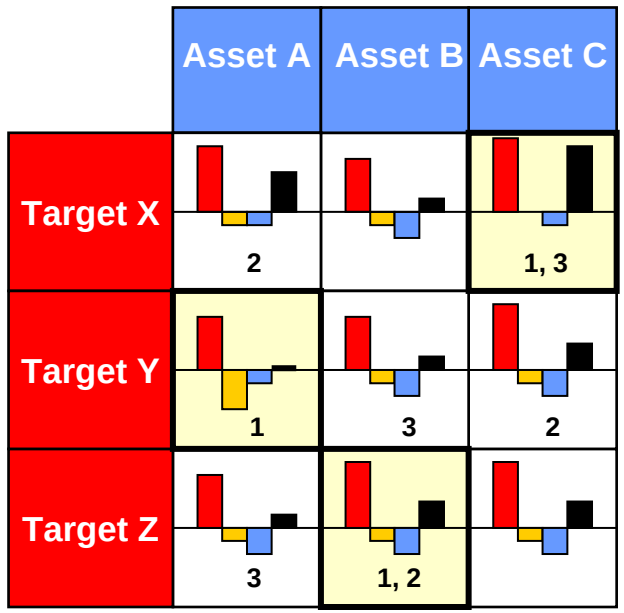


Figure 4: An illustration of Adjustable Assumptions for asset (Blue) and target (Red) priorities, showing the relative magnitude of High (H), Medium (M) and Low (L).

Structure Mapping in Visual Displays for Decision Support

Kevin Burns
Craig Bonaceto



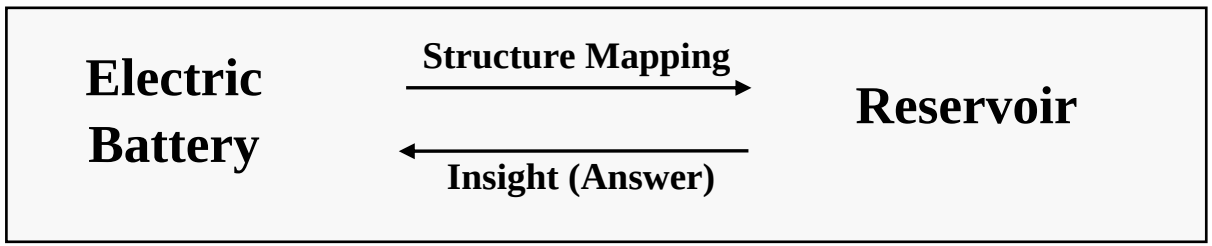
MITRE

The Problem

- Computerized decision support systems for C2 are often designed without adequate attention to the psychology by which decisions and other human inferences are made.
- Operators may not use these systems, and performance may actually be degraded by such systems.
- In this briefing, we:
  - Present an approach to decision support design founded on a cognitive theory of analogy called Structure Mapping.
  - Analyze the C2 problem of time-sensitive targeting and present a support system designed in accordance with Structure Mapping.
  - Summarize guidelines for generalizing the approach to other systems.

A Solution in Structure Mapping

- In Structure Mapping, a body of knowledge in one domain is mapped to a body knowledge in another domain based on common structures shared by the two domains.
- Example: “An electric battery is like a reservoir.”



- We are interested in constructing visualizations that support C2 decisions by mapping the structure of a C2 problem to the graphical features of a visual display.

Structure Mapping: The Missing Link

C2 Problem ⟷ **C2 Picture**

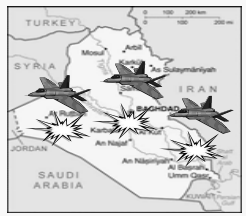
Structure Mapping →

← Insight (Answer)

Weapon-Target Pairing

Standard Mapping →

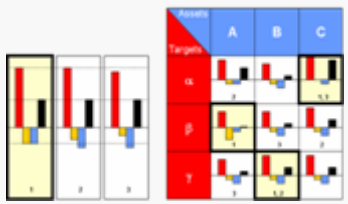
← / **A Missing Link**



Weapon-Target Pairing

Structure Mapping →

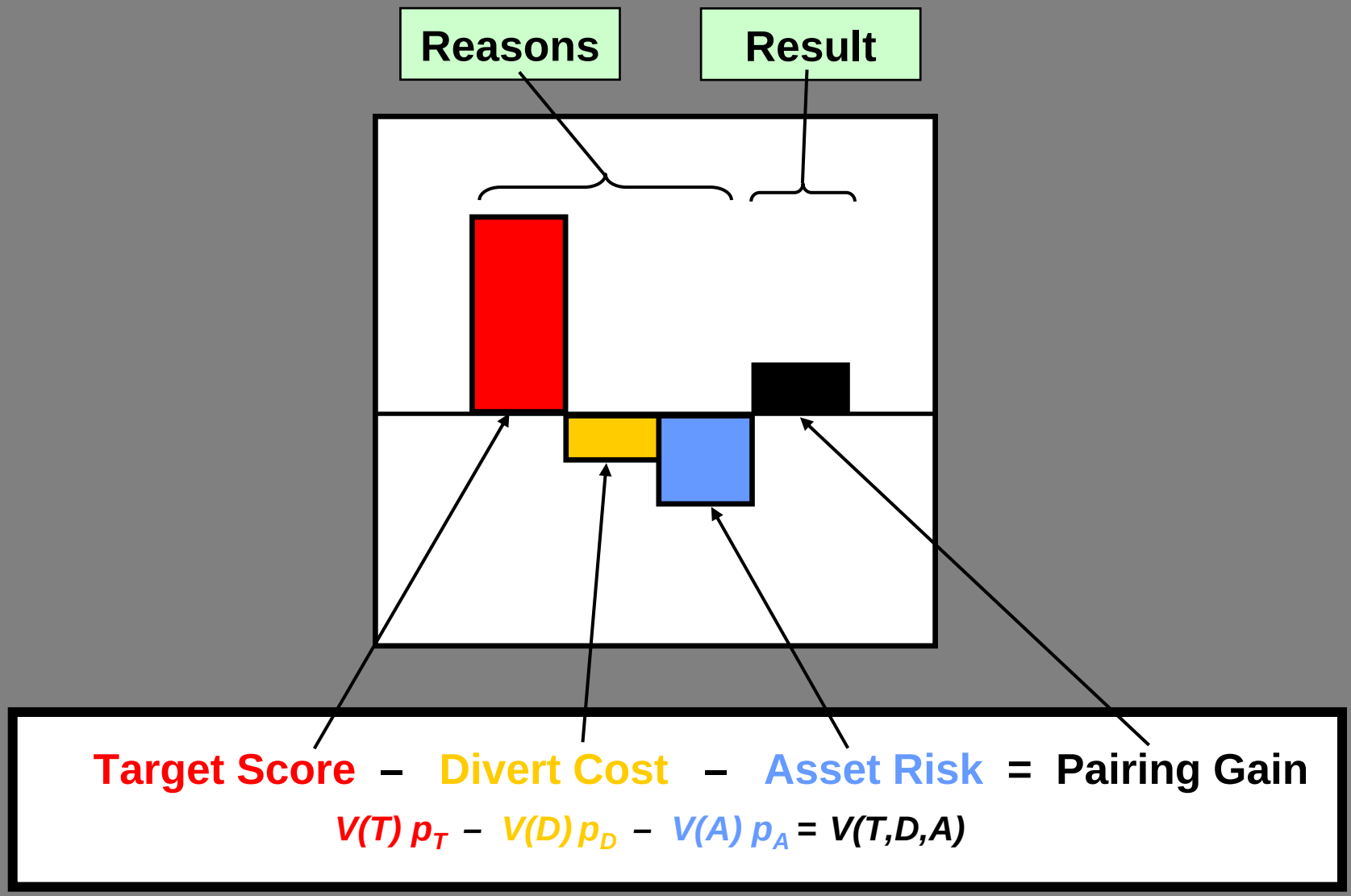
← Insight (Answer)



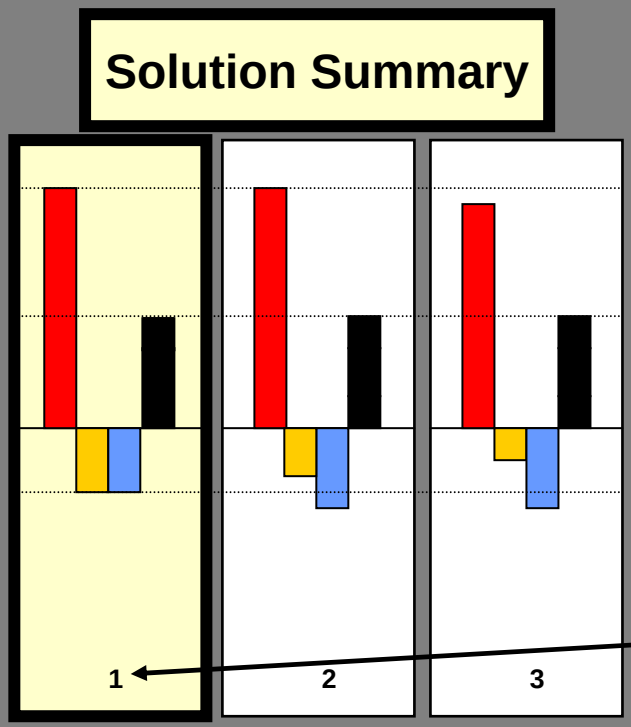
The Example: Weapon-Target Pairing for Time-Sensitive Targeting

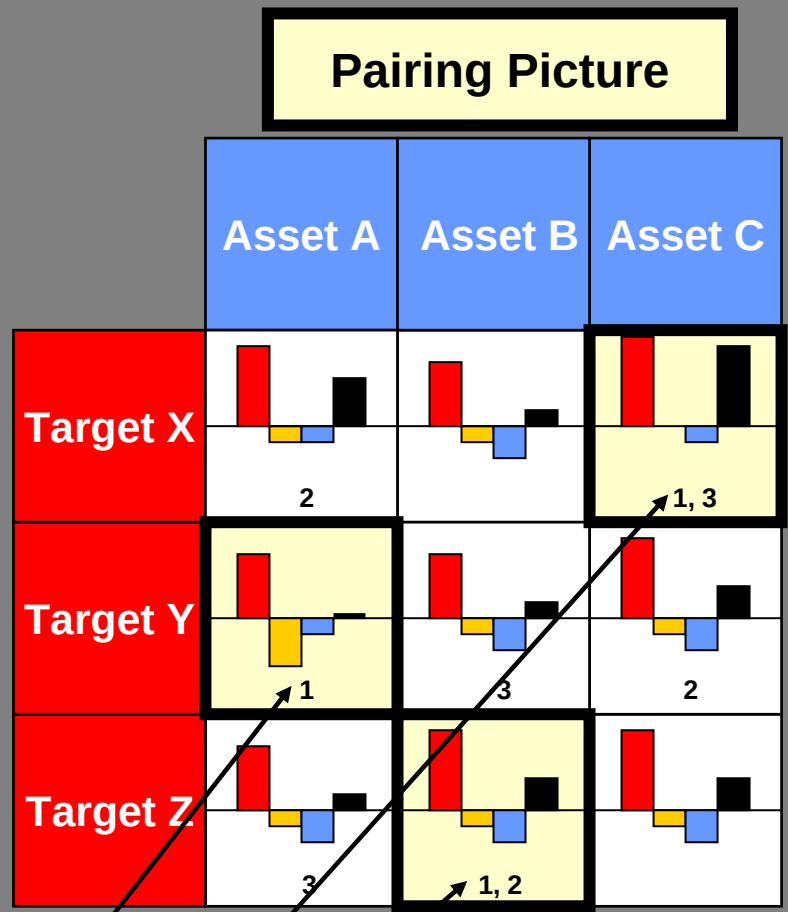
- **The C2 Problem: Which weapons (assets) should be diverted from their originally scheduled targets to attack time-sensitive targets?**
- **A Solution:**
  - **Given asset/target priorities and loss/kill probabilities, compute the value of pairing each asset against each target.**
  - **Compute a set of asset/target pairings with maximum expected value.**
- **The Structure Mapping Problem: How to construct a visual display to show what assets should be assigned to what targets and why?**

Mapping the Value Function



Mapping the Problem and Solution Spaces





Numbers in the Problem Space matrix correspond to the solution they're part of.

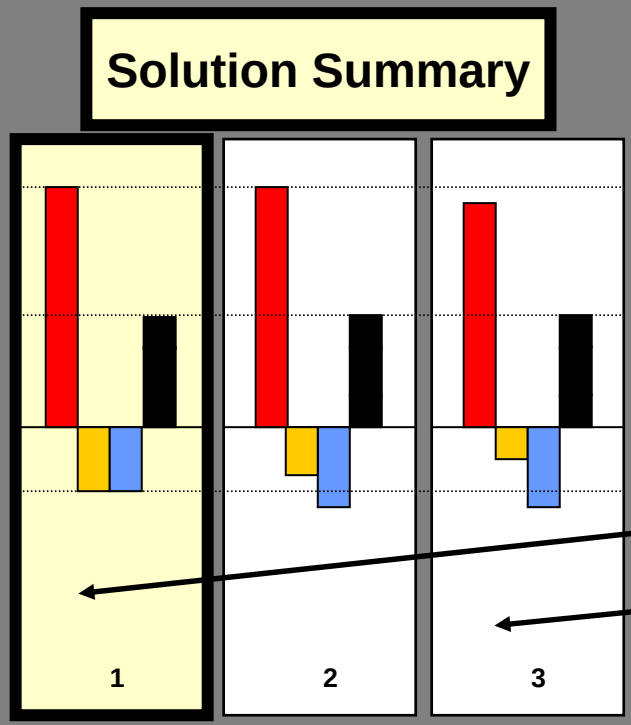
Is This a Better System?

Question: But why is this system better than just giving operators the “optimal” answer?

Answer: Operators realize there are many assumptions and uncertainties behind system solutions, and there may be many sub-optimal solutions that are almost as good as the so-called optimal solution.

Let’s see some examples…

Commander's Guidance: "Minimize Divert Cost!"



Solutions 1 and 3 have the same overall score (black bar) but different divert costs and asset risks. Solution 1 is actually sub-optimal if the commander's guidance includes a desire to minimize divert cost.

I Know Something You Don't Know

Target Score (+)
Divert Cost (-)
Asset Risk (-)
Pairing Gain

An operator may know that Asset C's effectiveness against Target X is sensitive to weather. Thus, he may choose to avoid any solution that includes this cell.

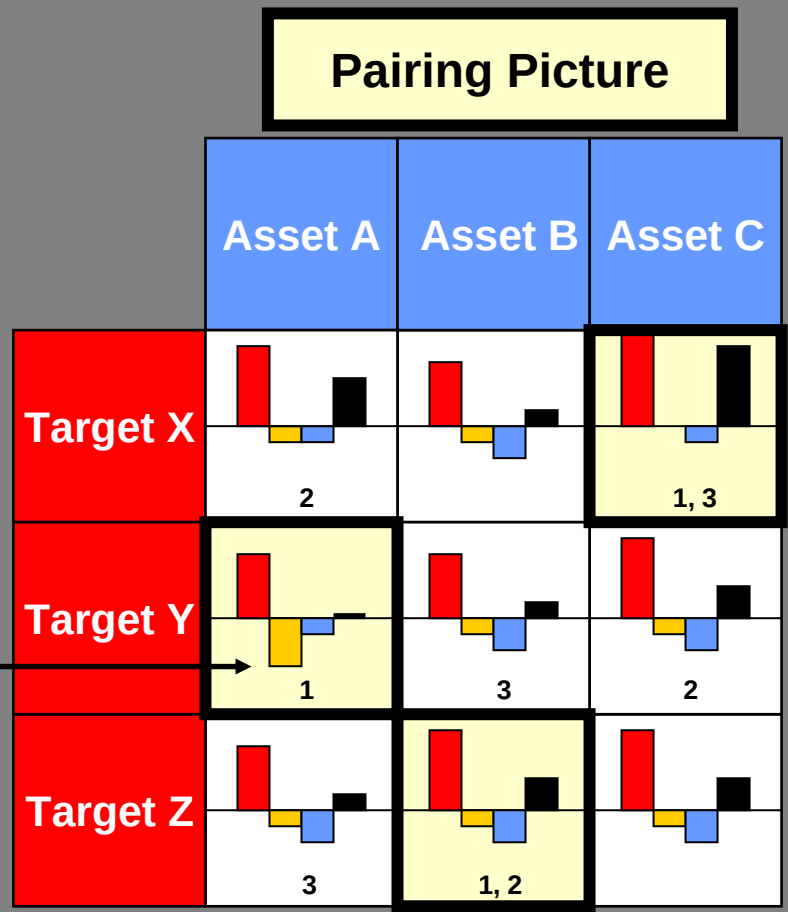
Pairing Picture

	Asset A	Asset B	Asset C
Target X	2		1, 3
Target Y	1	3	2
Target Z	3	1, 2	

Do I Really Want to Re-task Maverick?

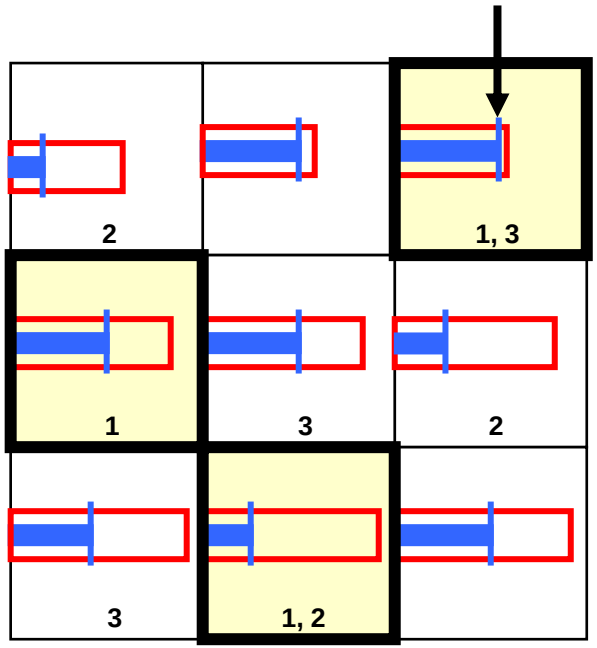
Target Score (+)
Divert Cost (-)
Asset Risk (-)
Pairing Gain

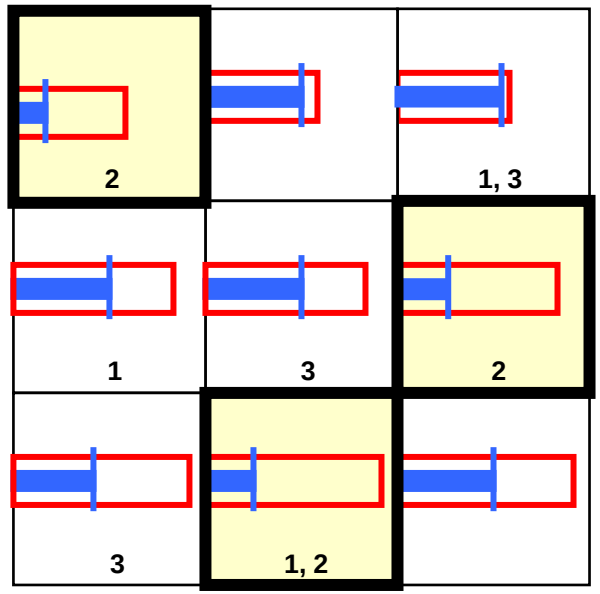
Most of the gain in Solution 1 comes from pairing Asset C against Target X and Asset B against Target Z. Even though this cell is part of the optimal solution, an operator might choose not to send Asset A to Target Y since the pairing gain is nearly 0.



Will Iceman Get There in Time?

Solution 1 has a cell with close timing

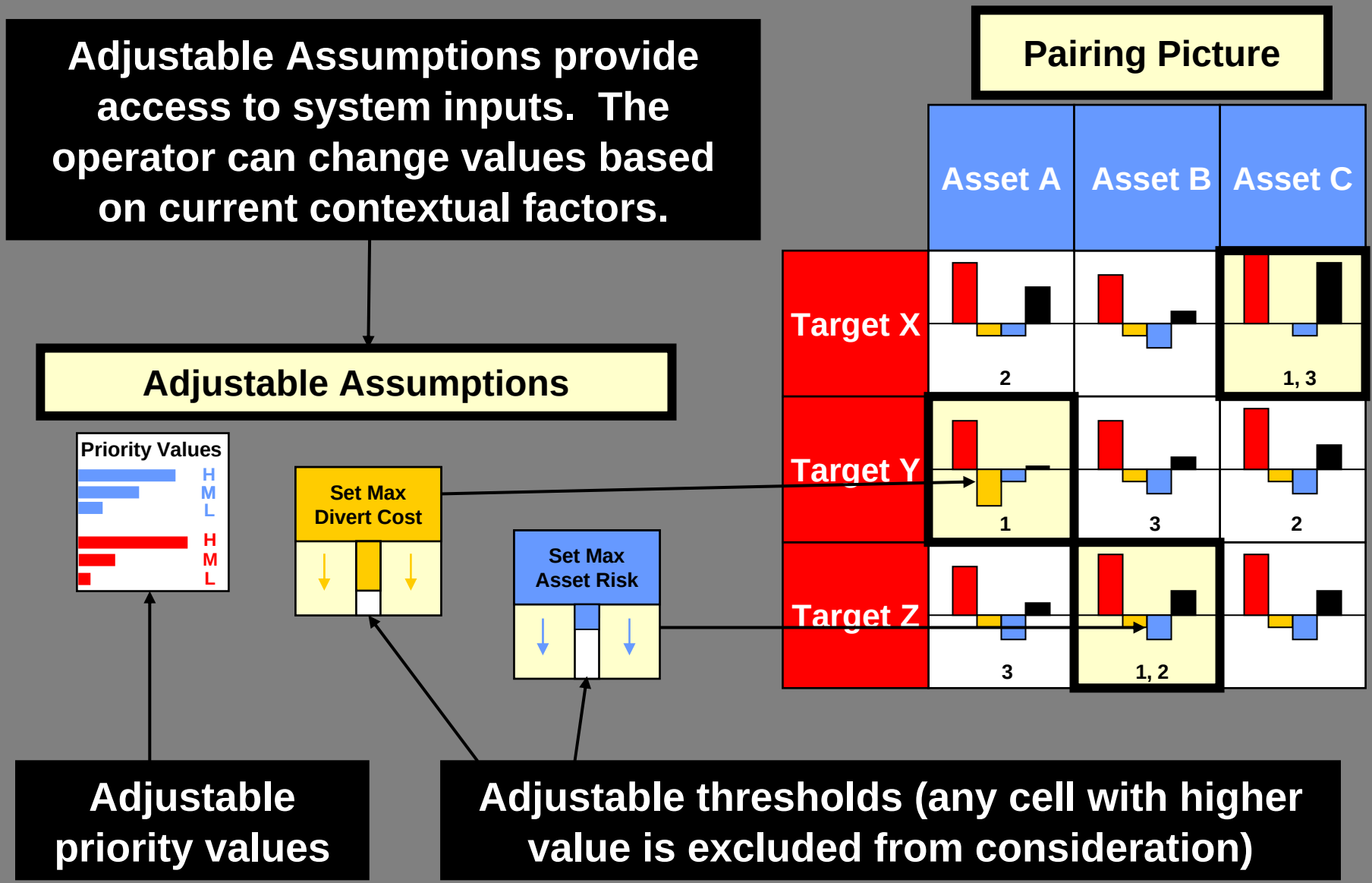




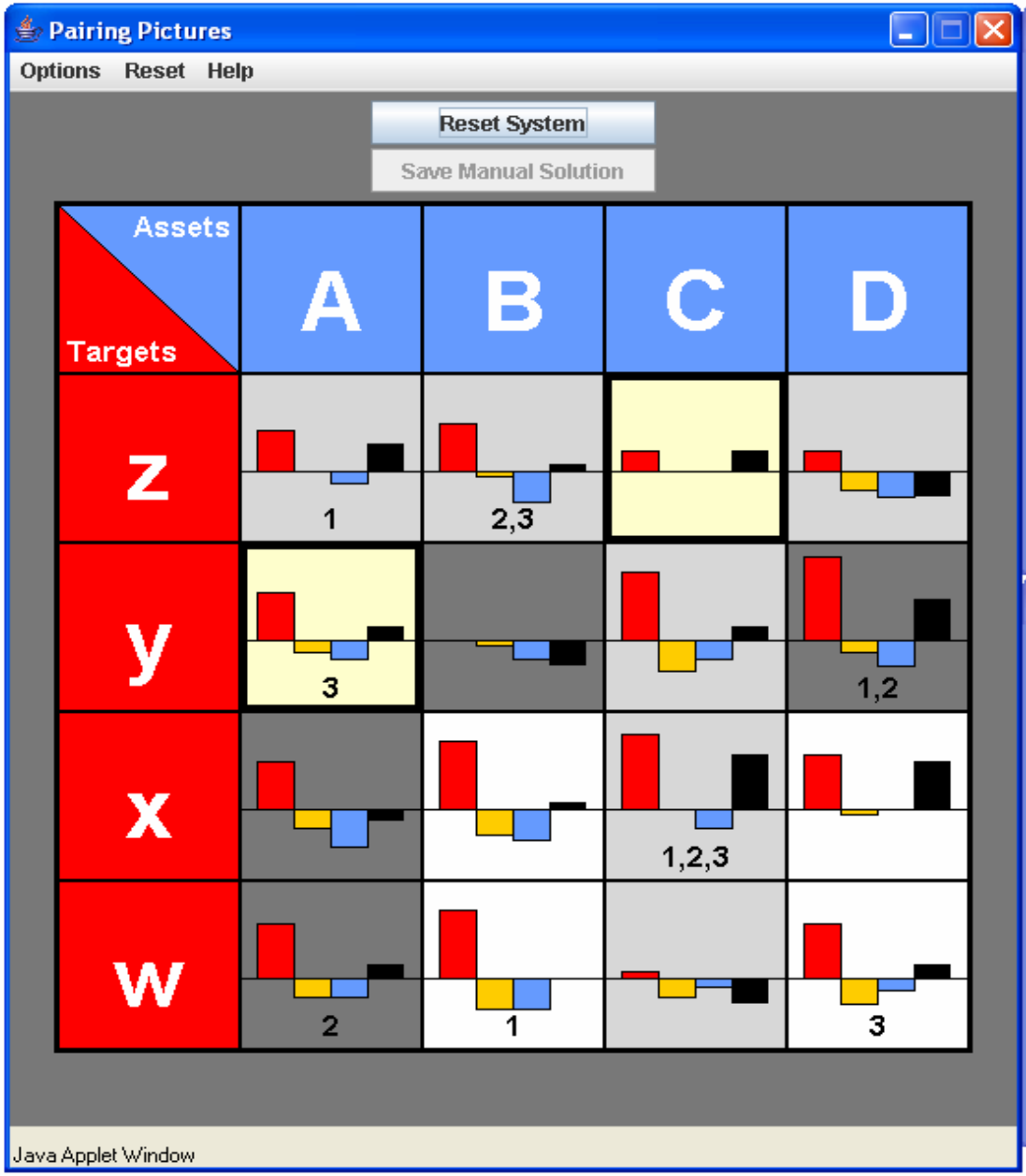
Solution 2 might be better, because it's more robust.

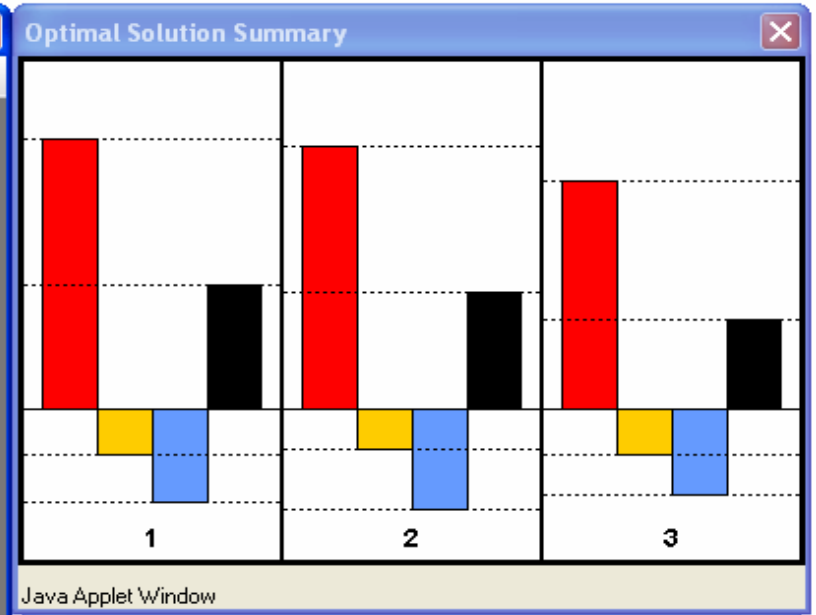
What are the Underlying Assumptions?

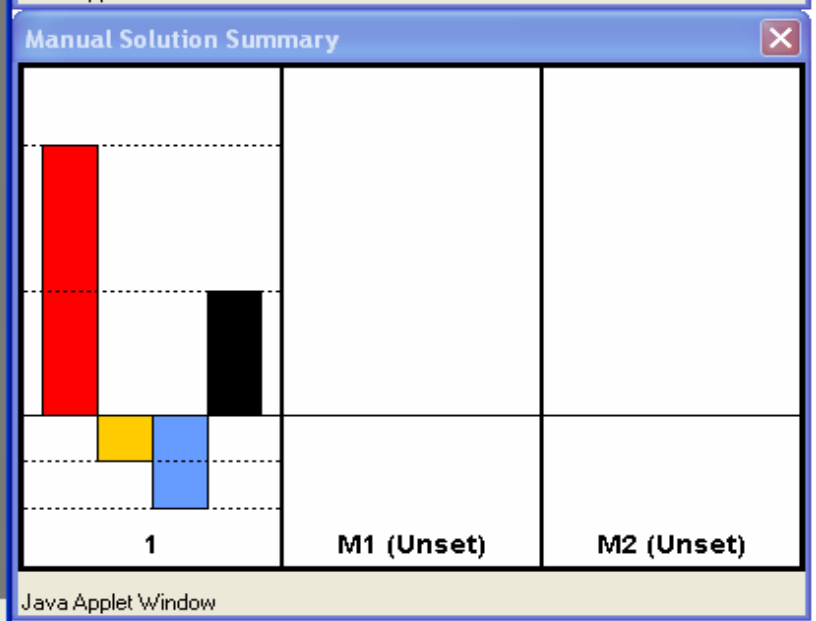
Adjustable Assumptions provide access to system inputs. The operator can change values based on current contextual factors.



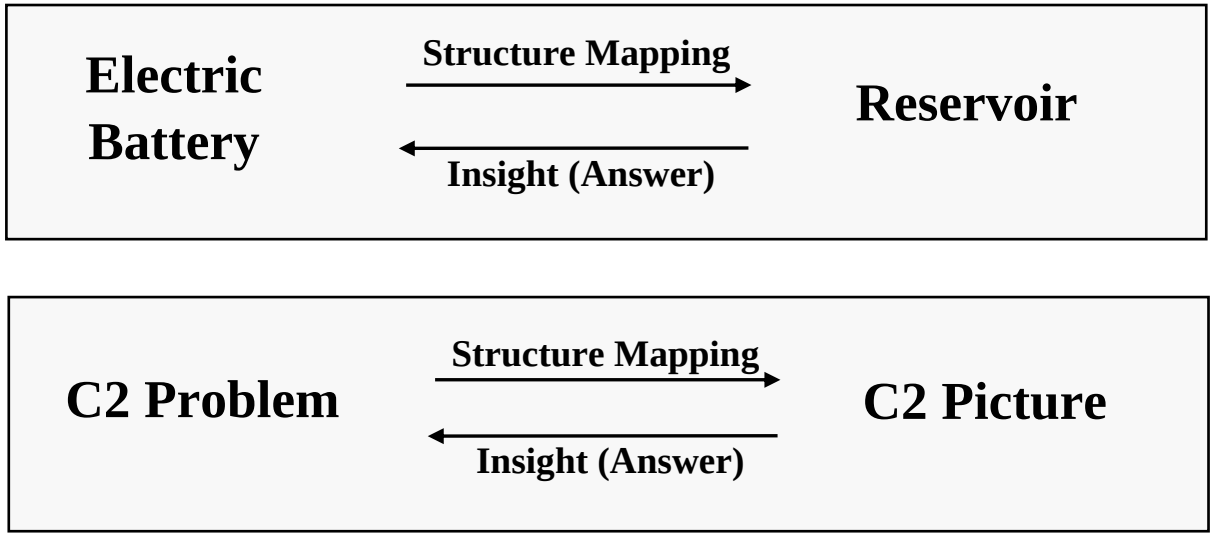
Taking Control: Manual Solutions







Applying Structure Mapping to Other Systems



- The goal of Structure Mapping is to identify *conceptual aspects* of the problem and illustrate them with *graphical attributes* of a display.
- The graphical structure of the display should be analogous to the conceptual structure of the problem and solution.
- The technique may be applied to a wide range of systems in environments that require human judgments and decisions.

Questions?

- **A working prototype of the system is available online at:**

http://mentalmodels.mitre.org/Hyperlinks/Time_Is_Critical.htm