Capturing a Commander's decision making style

Eugene Santos Jr[a], Hien Nguyen[b], Jacob Russell[a], Keumjoo Kim[a], Luke Veenhuis[b], Ramnjit Boparai[b], Thomas Kristoffer Stautland[b]

[a] Thayer Engineering School, Dartmouth College, 8000 Cummings Hall. Hanover, NH 03755

[b]Dept. of Computer Science, UW-Whitewater. 800 W. Main Street. Whitewater, WI 53190

ABSTRACT

A Commander's decision making style represents how he weighs his choices and evaluates possible solutions with regards to his goals. Specifically, in the naval warfare domain, it relates the way he processes a large amount of information in dynamic, uncertain environments, allocates resources, and chooses appropriate actions to pursue. In this paper, we describe an approach to capture a Commander's decision style by creating a cognitive model that captures his decision-making process and evaluate this model using a set of scenarios using an online naval warfare simulation game. In this model, we use the Commander's past behaviors and generalize Commander's actions across multiple problems and multiple decision making sequences in order to recommend actions to a Commander in a manner that he may have taken. Our approach builds upon the Double Transition Model to represent the Commander's focus and beliefs to estimate his cognitive state. Each cognitive state reflects a stage in a Commander's decision making process, each action reflects the tasks that he has taken to move himself closer to a final decision, and the reward reflects how close he is to achieving his goal. We then use inverse reinforcement learning to compute a reward for each of the Commander's actions. These rewards and cognitive states are used to compare between different styles of decision making. We construct a set of scenarios in the game where rational, intuitive and spontaneous decision making styles will be evaluated.

Keywords: Decision making process, Double Transition Model, inverse reinforcement learning, inverse optimal control, cognitive states

1. INTRODUCTION

The decision-making process is one of the key factors to successfully commanding a battle[1, 2]. This process requires that Commanders assess the relevancy of retrieved information such as intelligence reports, field reports, and inter-agency communications. They need to teach their subordinates to focus on the important information through training and actions. Their experience, knowledge, creative thinking, biases, and beliefs are used to make tactical, operational, and strategic decisions. Each of these factors contributes to the formation of a Commander's unique decision making style which could significantly affects the outcome of battles. Therefore, capturing the differences in Commanders' styles through modeling their decision-making processes is beneficial for the Commanders themselves to help reinforcing good decisions and learning from bad decisions. Additionally, this is quite helpful for training junior Commanders or staff to make decisions in warfare situations. However, this process is very complicated, especially in a warfare setting when making a timely decision under extreme pressure and limited resources is required. This is a challenging research problem that has attracted some interest from the cognitive science community and decision modeling (DM) community (e.g [3, 4, 5, 6, 7, 8, 9]). Normally, these styles are identified by conducting a questionnaire survey with users either before or after making a decision. Unfortunately, this does not capture and integrate the decision maker's cognitive process *when* he made decisions. We need a computational framework in which a user's decision making process is captured and his decision-making style is inferred.

In this paper, we set out to capture a Commander's decision making style by modeling his decision-making process using a computational model in which each state reflects a stage in his mind on the way to arriving at a final decision and each action reflects a possible move to execute a decision. We model the Commander's decision-making process over time using Double Transition Models[10,11] (DTM). A DTM can be used to derive a dynamic Markov Decision Process (*d*MDP) in which each state reflects a cognitive state of a decision maker in a combat setting and each action represents the tasks and decisions that he makes during that battle while the rewards represent the effect of a decision's outcomes. The differences between decision making styles could be highlighted through analyses of the graph structure of DTMs and the changes in the reward functions over time. The novelty of our approach lies in the development and use of a computational model to model, quantify, and recognize different styles in a decision making process.

Updated 3/20/14

We demonstrate the evaluation of our approach through two assessments. The first assessment focuses on *how well* we recognize different decision making styles by analyzing the graph structures of their DTMs and the second assessment focused on exploring *how significant* the rewards inferred from their DTMs are. For both assessments, we use hypothetical user profiles and real users. For hypothetical users, we create three different profiles which corresponding to different styles: spontaneous (novice), intuitive (learner), and rational (planner). With real users, we have two Commanders with two styles: spontaneous and intuitive playing a naval warfare game called SteelOcean. We measure the differences in the DTM structures using a few different measures including the density of the graph, average outgoing edges and average incoming edges. Additionally, we compute the correlations between the reward functions over a set of cognitive states with the game rewards over the period of an online naval wargame and use them as a similarity measure between Commanders.

This paper is organized as follows. We begin by reviewing key related work with regards to cognitive styles in decision making and provide background on DTMs. Next, we describe how to leverage DTMs with *d*MDP through inverse reinforcement learning (IRL). We then describe the two assessments, with a detailed discussion of the results. Finally, we present our conclusions and future work.

2. RELATED WORK

The novelty of our approach is to capture a decision-making style using a computational model that keeps track of traces of Commander's actions as well as his interest in the information he processes. In this section, we highlight some related work in the area of decision making styles and provide background information on our computational model called DTM.

2.1 Decision making style

Decision making style has been defined as "the learned habitual response pattern exhibited by an individual when confronted with a decision situation"[5]. Five decision making styles have been in studied in the cognitive science community over the years including rational (reason-based), intuitive (heuristic-based), dependent, avoidant and spontaneous[3,4,7]. Rational decision making is the process of considering all alternatives and logically makes a choice while intuitive decision making is based on heuristics, previous experience, motion to make decision. Dependent decision making is to rely on others' recommendations. Avoidant decision making is a lack of decision making. Finally, spontaneous decision making is a desire for quick decision making with response to the immediate need[3,4]. In combat situations, there is a naturalist decision making style which refers to the use of previous experience to make critical decision under a lot of pressure and time constraints[5]. This is closely related to the intuitive decision making style mentioned above. Decision making style is a very important factor to assess the quality of a decision and identify the errors in judgement[5]. It is also related to a broader concept that has been studied thoroughly which is cognitive style. A study[7] has explored a specific relationship between cognitive styles and decision making style such as "*People who are more extroverted in personality are more likely to have intuitive cognitive style, while those who are more introverted in personality are more likely to have analytic cognitive style"*. Even though that study has not confirmed this relationship, it doesn't reject it and lays the groundwork for further study. In recent years researchers have produced many other works that study decision making style in different domains (including pharmaceutical training[12], construction work[13], and work life balance[14]) have shown that determining one's decision making style is a crucial step to evaluate and distinguish between good and poor decision makers and helps improve the quality of decision making process. In this work, we aim to determine different decision making style by using a computational framework to capture his/her entire decision making process.

2.2 Double Transition Model (DTM)

In this paper, we model a Commander's decision-making processes using a Double Transition Model (DTM), originally proposed in[10,11] as a way to describe human opinion formation processes through computational simulation. The DTM is based on a graphical representation of human cognitive states during their decision-making processes, in which a node is composed of two subgraphs, a Query Transition Graph (QTG) and a Memory Transition Graph (MTG). The QTG is used for representing the dynamic changes of the Commander's interest/focus. Therefore, each node in the QTG represents a single query in the Commander's mind and is represented by a vector $[X, ?, b_1, c_2, \ldots]$ where $X \in U$ denotes the target random variable of interest, an element of the set of all random variables representing features, ? denotes the unknown instantiation of the variable $A \in U$, and b_1 and c_2 denote an instantiation of variables $B \in U$ and $C \in U$. The

MTG is a graphical representation of the underlying knowledge the Commander has in his mind, and is based on a Bayesian Knowledge Base (BKB)[15].

Figure 1 shows an example of the DTM, where a Commander may have perceived new information about the location of enemy battleships. The Commander could undergo a memory transition from enemy battleship 'concealed' to 'visible' based on the change of his perception through MTGs. Asking himself a question of "Is there enemy battleship around?", the Commander takes an action to destroy the enemy battleship when he believes it is the right action to take at that time.

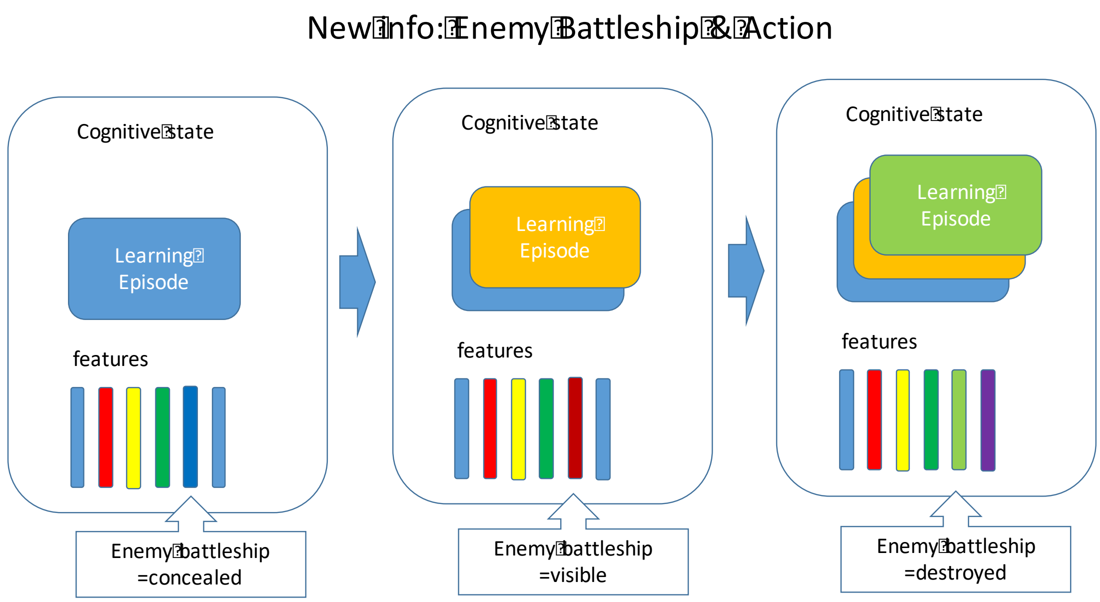

Figure 1. An example of a DTM framework.

The random variables (such as 'Action' and 'enemy battleship') and their relationships in the DTM and its underlying QTG and MTG are unknown at first. They must be identified and extracted from the Commander's information stream in order to incorporate them into the MTG representing the Commander's memory. Each node within the DTM represents the Commander's cognitive state (via the MTG) and what he focuses on (via the QTG). Transitions between states represent decisions made and actions taken. The goal in the DTM is to reconstruct the sequence of decisions that the Commander made in order to predict future decisions by leveraging repeating patterns. The DTM has been applied to represent a Commander's decision making process in our previous work[15].

3. OUR APPROACH

We capture a Commander's decision making style by building a computational model to represent his decision-making process. The core of our approach is the DTM[11] constructed from traces of the Commander's episodic information-processing processes and his actions with respect to his subordinates over time. Each Commander has his own DTM which describes his internal cognitive states and is determined by using external relevant information sources related to his decision-making process. Our model allows us to follow a Commander's actions and to explain his decisions based on information relevant to those decisions. We extracted information about a Commander's decision making styles using factors describing the DTM graph and the distribution of rewards over time. The original DTM framework[10,11] had used reinforcement learning to learn an optimal policy from an environment that has an observable reward value reflecting how good or bad an activity is in reaching an optimal decision. In complex battlefield scenarios, the true reward value may not be available and it is necessary to approximate it. An approximation function can be learned via inverse reinforcement learning[17].

3.1 Architecture

A Commander's DTM is built from scratch using the actions that he has taken, the contents of the reports that he has read and found relevant as well as the communication that he has with his subordinates over time (as shown in Figure 2). Each of these artifacts are grouped by timeline and pre-processed to extract what we refer to as a "feature". A feature is a distinctive attribute of the action space and the target domain where a Commander is currently making decision. A feature is assigned a specific value which represents the observed status of that feature at a particulate point in his decision-making process. Features could be pre-determined manually or could be automatically extracted using algorithms. Our previous work[15] showed the process of extracting a set of features from documents automatically using document graphs and Latent Dirichlet Allocation[18] (LDA). These features are utilized in our DTM. We derive a dynamic Markov Decision Process (*d*MDP) from the DTM. A MDP[18] is a commonly used model of the decision-making process in uncertain environments. The difference between our model and the existing MDP approaches is that we use *d*MDP to model actions and states which may have never been seen before and cannot be predicted in advance. In this *d*MDP, it is possible to estimate a reward function that describes the Commander's unique style and method for making decisions. We have determined a solution to this problem by estimating the optimal solution using Inverse Reinforcement Learning. Developed at Stanford University[17], Inverse Reinforcement Learning (IRL) replaces a well-defined but often overly simplistic reward function with a policy that is based on optimal observed behavior, such as that of a Commander with years of experience to draw upon along with knowledge of the common behavior of enemy forces in a variety of situations. In this section, we describe the DTM Builder and the process of mapping to *d*MDP and using IRL to solve for rewards.

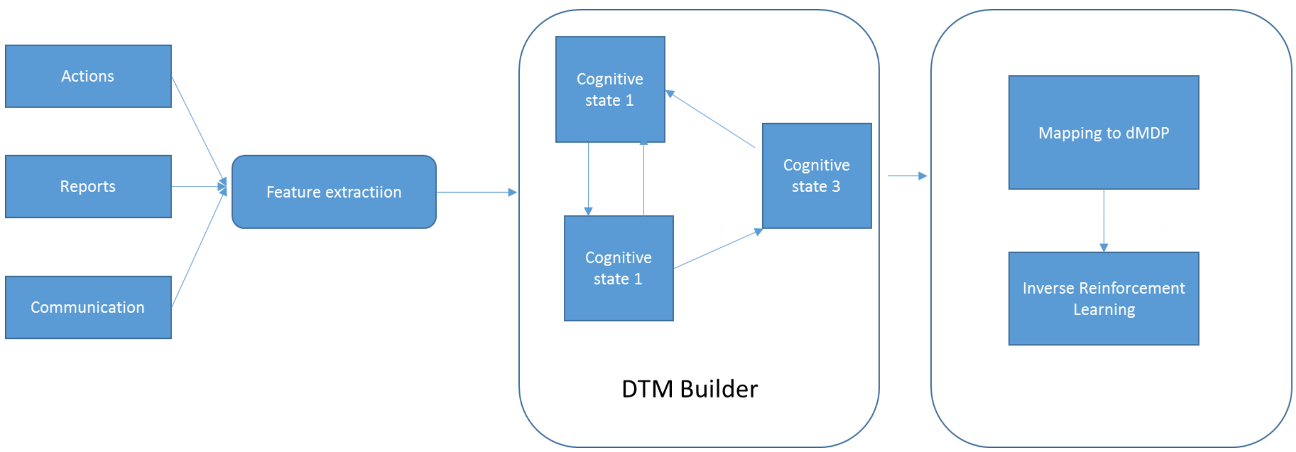

Figure 2: Overall architecture of our approach

3.2 DTM Builder

The DTM Builder constructs a DTM from a set of traces of a Commander's behavior. These traces are sequences of observations about the world and which action he took, transitioning the Commander to another set of observations. For example, the observations might be that there is a fog area that may be concealing a ship, so we perform an action to check the fog are, resulting in another observation that there is no ship.

To build the DTM, we take the traces of actions and observations to construct a cognitive state for each observation and take the observed actions to construct an action edge which transitions from one cognitive state to another cognitive state. We iteratively update the DTM as we see new traces.

The cognitive states, themselves, are a cross product of the memory and query transition graphs and therefore represent what the Commander believes and what is important at a snapshot of time. Since the interest of the Commander is not directly observable, we make an initial assumption that the Commander is interested in all known features at any given point in time. The observables are mapped onto attributes for features F_i forming a feature vector X such that the size of X

is less than or equal to the number of features. In the general DTM formulation, we have another vector Q such that $|Q| = |X|$ represents the focus of the Commander. $X \times Q$ forms the basis of the cognitive state S.

Next, we represent a vector of all of these actions as $a_i = [a_{i1}, a_{i2}, \ldots, a_{in}] \in A$. These actions will be interleaved with observables about the world. The Commander may be performing multiple actions simultaneously, such as moving multiple ships at the same time.

These actions and cognitive states form the basis of traces T (sometimes called trajectories or demonstrations in the MDP and IRL literature[19]). Traces are composed of a sequence of individual transitions between states. The transitions inside of each trace form a 3-tuple (s_i, a_j, s_i') where $s_i, s_i' \in S$ are states and $a_j \in A$ are actions and the full 3-tuple is a transition from state s_i to state $s_i{}'$ through taking action a_j. The DTM builder aggregates counts of each 3-tuple to compute probabilities for a $S \times A \times S$ transition matrix where each cell is the probability of transitioning from s_1 to s_2 when taking action a_i.

This process is updated as we see new traces. New traces may have additional features which were not previously seen; as such, the underlying distribution of the 3-tuple probabilities is only guaranteed to be stationary at an exact point. In other words, we can learn a reward function to solve for the reward assigned to each cognitive state at a given point in time within the DTM but there may be nonlinear changes with the addition of traces or even just new features. Therefore, we want to map to a simpler model to solve at each snapshot. We can map the DTM to a MDP at any given point in time, but a single MDP is not sufficient to model how the Commander's behavior changes as they learn from old traces. Since the process of decision making may have information come in and out of relevancy, information and states come in and out of memory in the DTM in an episodic manner. Since we are interested in modeling not just the current belief about the Commander's actions, and instead focus on his decision-making process, we extend MDPs to account for dynamic information, hence dynamic Markov Decision Processes (*d*MDPs).

3.3 Mapping from DTM to *d*MDP

The DTM connects the Commander's actions to his cognitive states and even models how this changes over time, but it does not know whether the individual states are preferred or not. MDPs, on the other hand, have well known algorithms to learn this through inverse reinforcement learning[16]. To leverage these algorithms, we would like to map our DTM onto a MDP at any point in time so that we could update the approximation of preference for cognitive states. Unfortunately, MDPs have fixed state and action spaces at the time of learning which is not feasible to know ahead of time when modeling unseen behavior. In our cognitive architecture, we must be able to incorporate new information that a Commander may not have even thought about previously and we want to model the process of the changes to the *d*MDP as well. Therefore, a MDP that represents the DTM must be rebuilt every time we have new actions, states, or even changes in probabilities. We refer to the MDP that we map the DTM onto as a dynamic Markov Decision Process (*d*MDP) because we will eventually be analyzing what changed, why it changed, and what the impact of the change was over the evolution of the MDP.

Like the DTM, the *d*MDP is composed of states and actions that are mapped as a bi-jection onto each other. Therefore, the *d*MDP has the same states S, actions A, and connectivity defined by the 3-tuples (s_i, a_i, s_i') with the corresponding DTM. The probabilities of each 3-tuple are defined by the transition matrix from the DTM, and hence the only factor that the *d*MDP needs to provide is a function to compute the reward of a state in the *d*MDP and consequently, because of bi-jectivity of states and cognitive states, the reward of a cognitive state. The *d*MDP has no access to the features, instead each state already has a unique mapping onto feature instantiations from within the DTM.

We also need a mapping from traces onto states in the *d*MDP when there may not be an exact match between the observations and an individual cognitive state. This comes up when we compare reward values for traces across multiple DTMs. In this case, we do not want to modify the DTM because then we change the underlying reward function in a (possibly) nonlinear manner. Instead, we map observables from the trace onto their closest states and actions in the DTM. Since there is a direct mapping within the DTM to the underlying *d*MDP, we can resolve states within the *d*MDP as well.

We use Hamming distance between the feature values at each state and the features in the underlying trace to find the closest state.

Once we have mapped the DTM onto the *d*MDP, we can perform inverse reinforcement learning on the *d*MDP to learn the reward values for the states in the *d*MDP, and thus also for the cognitive states in the DTM.

3.4 Using IRL to compute rewards

The underlying *d*MDP generated from the DTM does not initially have any rewards for any states. The DTM is supposed to learn the rewards that the Commander may have assigned to each cognitive state. In this manner, the model is supposed to generalize and identify the preferences of each Commander and how he likes to solve the problems that he is solving. The mapping of the DTM onto a MDP at any given point in time lets us leverage the existing work in inverse reinforcement learning to estimate the rewards.

We tried the linear programming approach proposed in the seminal inverse reinforcement learning work proposed by Ng and Russell[17]. The sparsity of many real-world problems appears to lead to a distribution of rewards that is heavily skewed towards 0 with this formulation. Rewards in our evaluation below only had peaks at the maximum reward value (set at 500), ~1/5th of the maximum reward value (100), and $0 \pm 1e-12$.

We then tried the maximum entropy inverse reinforcement learning formulation proposed by Ziebart et al.[21], using an implementation freely available on GitHub[22]. The distribution of rewards on cognitive states had similar peaks, but was distributed more evenly away from zero.

4. EVALUATION

The objectives of our evaluation are twofold: first, we would like to assess whether we could distinguish different decision making styles by comparing the graphical structures of different DTMs; second, we want to study how significant the reward values are by exploring the reward distribution and by comparing the correlation of reward functions to the game rewards. For both assessments, we use hypothetical user profiles and real users playing an online naval warfare game.

4.1 Capturing decision making styles with hypothetical users

Our hypothetical users employ a set of simplified scenarios based loosely on the game called Steel Ocean[1] which is a World War 2 Naval combat simulator where players take command of one of over 100 warships and battle in vast ocean arenas. In these scenarios, the Commander has an aircraft carrier that can deploy scout planes and two battleships. The enemy forces can have a submarine and a battleship, though the number of ships is not required to be fixed. There are four smoke/fog areas in which the enemy forces can be concealed. We simulated three decision making styles: rational, intuitive, and spontaneous by creating three typical profiles: planner, learner, and novice, respectively. The planner makes his decision rationally. That means he carefully considered his choices based on his knowledge and training and chose the optimal choice at any given time. For example, he could move multiple ships at a time to check out each fog area to minimize the risk of losing the fleet own ships while simultaneously maximizes the chance to destroy the enemy ships. The learner makes decisions intuitively. He uses his past experience to guide his current direction. For instance, he learned from his previous mistakes as well as success to decide what actions to take. The novice makes his decisions based on reacting to the current situation, in a spontaneous fashion. For example, he sends a single ship to check out each smoke at a time or decides to engage with the enemy once the enemy engages him. With planner and novice decision makers, we made up 84 different learning episodes, each of which contains 10 moves. Each move describes what happens at a specific time, such as at time 2, we send a ship to scout enemy while the enemy ship appears to be idle. For learner we made up 74 such episodes with 10 moves each.

We used the following measures to assess the graph structure of each DTM: the number of states, the number of actions, density of the graph, average of state to action (out-degree), average of action to state (in-degree). The density of the graph is computed as follows: $d = \frac{|E|}{|V|*(|V|-1)}$ with *E* being the number of actions and *V* being the number of states. Average of state to action is defined as the average of outgoing edges for each state while average of action to state is defined as

[1] SteelOcean is available at http://store.steampowered.com/app/390670/

the average of incoming edges for each state. As shown in Table 1, the number of states and actions for novice are the largest and those for planner are the smallest while those of learner fall in between. This could be explained as the novice seems to respond to a task as it appeared without using much of his previous experience because he has a limited amount of experience to recall. Therefore, he would have more cognitive states than the other two decision styles. The planner appears to be effective in his use of previous knowledge and therefore, his graph contains less states but is more dense compared to novice decision makers. That means his cognitive states seem to be repeated. In other words, he re-used his experience to apply to the new but related situations.

Style	States	Actions	Density	Avg.Outgoing	Avg.Incoming
Learner (Intuitive)	61	34	0.0094258	1.106614545	1.191934
Novice (Spontaneous)	69	40	0.0085307	1.1598958	1.238642
Planner (Rational)	41	12	0.0075724	1.04095066	1.1828077

Table 1: Capturing hypothetical users' decision making styles

4.2 Reward distribution for hypothetical users

In addition to the analysis of graph structure, we wanted to understand what the reward distribution looks like for each decision-making style and how to compare the decision making styles of the different Commanders based on reward information. In order words, we would like to study how significant the rewards are. The distribution of the rewards for cognitive states seems to be clustered mostly between zero and two for all of the different Commanders that we have (as shown in Figure 3). We note that the reward values themselves are not necessarily directly comparable between different DTMs because the rewards are learned via maximum entropy inverse reinforcement learning[21] and therefore the scale of the rewards may not be the same.

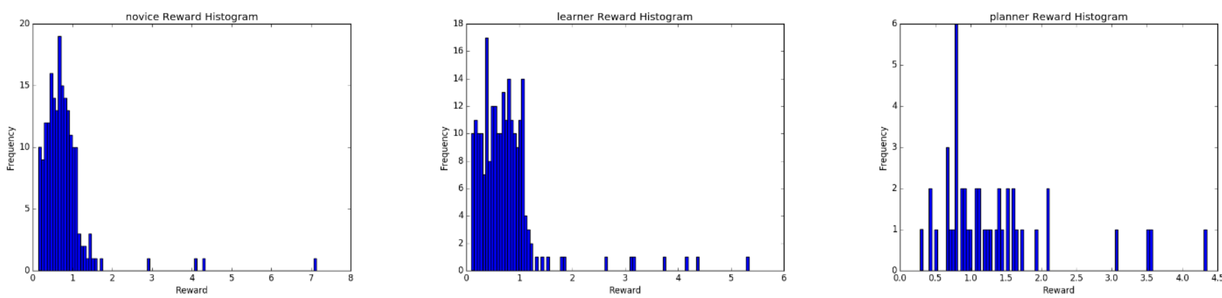

Figure 3: Histogram of rewards learned on cognitive states in the DTMs for different decision styles

While the reward values, themselves, appear to be uninteresting, they are important to view because they are not distributed normally which has an impact on how we can do our analysis. So instead of comparing the rewards directly to get an understanding of how the decision making styles differ and how they are the same, we compare the aggregate reward value for each of the individual traces. This is intuitive because inverse reinforcement learning does not optimize the rewards themselves, rather, it attempts to optimize the preferences evidenced by the traces.

In order to have a ground truth for comparison, we used the same synthetic scenarios with Commanders who have different decision-making styles (i.e learner, novice, and planner as previously defined in 4.1). We look to quantify how well each decision-making style actually performs in the synthetic task under the assumption that the actors who plan more, will have better performance. We use a game score as a measure of performance. In other words, we would like to know which DTM best predicts the game rewards. At learning time, the DTMs did not have access to the game rewards and the scale of the game rewards is likely to differ from the scale of the learned rewards. The hypothesis is that the planner (rational) decision maker who decided ahead of time what moves to make would have the best correlation with the game rewards, followed by the learner who plays intuitively, and finally followed by the novice who is spontaneously making moves. For our analysis, we compute Spearman's ρ coefficient because it does not require the relationship between the points to be linear and does not need to compare the magnitude. Spearman's correlation is computed as $\rho = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}}$, where rg_X is the rank orderings of the values in X, and the same for Y.

Decision Style	# Samples	Spearman ρ coefficient	Spearman p-value
Planner (rational)	19	0.977375648	6.60E-13
Learner (intuitive)	55	0.956522574	4.95E-30
Novice (spontaneous)	36	0.81681177	1.24E-09

Table 2: Correlations between rewards and game rewards for each decision-making style

As seen in Table 2, the ordering of the rewards for the planner is the most correlated with the actual reward from the game. It is surprising, that the rewards from the learner also match so closely. The rewards for the novice being so high, demonstrate that the synthetic game was easy to win in general.

Now we have demonstrated that the synthetic Commanders match our expectation for how similar they are to the game score, but we still want a method for comparing different DTMs, or finding a probability that a certain trace came from a particular DTM. To measure the similarity between DTMs, considering the reward values not being distributed normally and not necessarily having the same scale, we computed the sum of the rewards of each cognitive state matched within a trace and multiply by their probability. Given a set of actions A, a set of states S, and a trace $T = \{(s_1, a_1, s_2), (s_2, a_2, s_3), \dots\}$ consisting of triples with overlapping states, we compute a product $\prod_{a \in A} \Pr(a)^{\cdot} \sum_{s \in T} \mathrm{R}(s)$ where $\Pr(a)$ is the probability of the action in the DTM counted from all of the traces and $R(s)$ is the reward of the state learned from IRL. This gives us a single number for each trace measuring how much the Commander prefers that trace. Then we can run the same trace through another Commander's DTM, computing the reward the other Commander would assign to the same trace. Finally, we can compare the ranks of the preferences for traces that each Commander gives to the other Commander's traces. In this way, we can determine whether each Commander believes that each trace is preferable over the other traces.

Predicting Style	Predicted Style	# samples	Spearman ρ coefficient	Spearman p-value
Planner	Learner	55	-0.4068	0.0021
Planner	Novice	36	0.1534	0.3716
Learner	Planner	19	0.7386	0.0003
Learner	Novice	36	0.7979	5.61E-09
Novice	Planner	19	0.3351	0.1607
Novice	Learner	55	0.8481	3.07E-16

Table 3: Correlations using decision-making styles to predict rewards for traces of other decision making styles

The results (as shown in Table 3) are not symmetric because the predicting style and predicted style have different DTMs and therefore assign different rewards to the traces from the other DTM. Generally, if there is a high correlation between the orderings both ways, there is high agreement on the ordering. As expected, the novice and planner have Spearman coefficients closest to zero and insignificant p-values, implying that they are the least similar. The learner has high correlation with the novice, and medium correlation with the planner. It was expected that the learner would be somewhere between the novice and the planner. The number of traces from the novice that appeared in the learner is larger than the number of traces in the planner, therefore the learner expected to have higher correlation with the novice than with the planner, as we have seen.

4.3 Capturing decision making styles with real users

In addition to evaluate different decision making styles with hypothetical users, we managed to use a team of three humans to play Steel Ocean game. We use this game because it exists in a domain with many readily available examples of Naval Commanders with more concrete traces. These traces allow us to model multiple Commanders' traces and have a platform to understand how they differ, why they differ, and what uniquely composes a Commander's style over a set of tasks while still having control of how the tasks overlap. The process of this evaluation is shown in Figure 4.

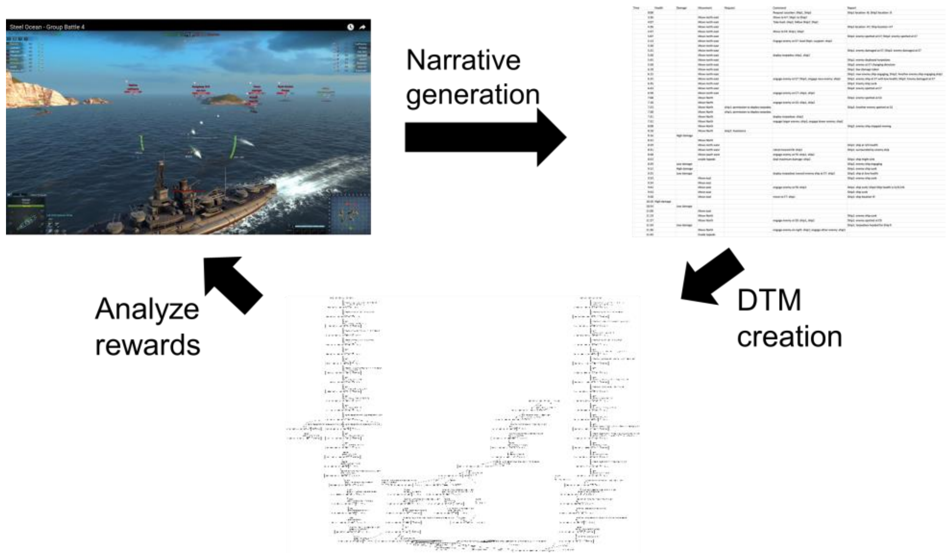

Figure 4: Process of evaluating with real users

We created and recorded videos of four scenarios, each of which was played by three student volunteers who have some experience playing wargames online. Each scenario describes a fleet battle between two teams: our team and the enemy. We have three fundamental pieces in each scenario: A Commander's ship, a Support ship and an Attacker ship. A Commander ship makes decisions involving the actions that the Attacker or the Support ship take. The Attacker ship is responsible for carrying out an action and the Support ship assists the Attacker ship in carrying out these actions. These pieces in each battle follow different formations in scouting the enemy and carrying out offensive and defensive actions. The formations are straight line, scattered line, triangle and D-formation as guided by United State Joint Force Command[2]. We selected two students to act as Commanders so each Commander has a chance to command two battles. The goal of the mission is to destroy as much as possible the resources of the enemy while trying to protect our own resources. The first two battles were recorded one week before the last two battles. The two Commanders have different decision making styles as it turns out: intuitive vs. spontaneous. Commander 1 (intuitive) tends to connect his experience with his decisions while Commander 2 (spontaneous) tends to react quickly to the tasks at hand without considering the alternatives. After recording these scenarios, we sample the movements and situations of each battle every 15 seconds and convert this data to a comma delimited file (*.csv). Each comma delimited file is a trace which, in turn, is used to create a portion of the DTM. Figures 5 and 6 each show a trace for Commander 1 represented graphically as the DTM it would produce on its own. Figures 7 and 8 each show a trace for Commander 2, again represented as the DTM that it would generate on its own. (Note that our primary focus for these figures is on the graph structure of the DTMs.) The details of the graph analysis of the DTM for each Commander is shown in Table 4. The DTMs for Table 4 include the union of both Figures 5 and 6 for Commander 1 and the union of Figures 7 and 8 for Commander 2.

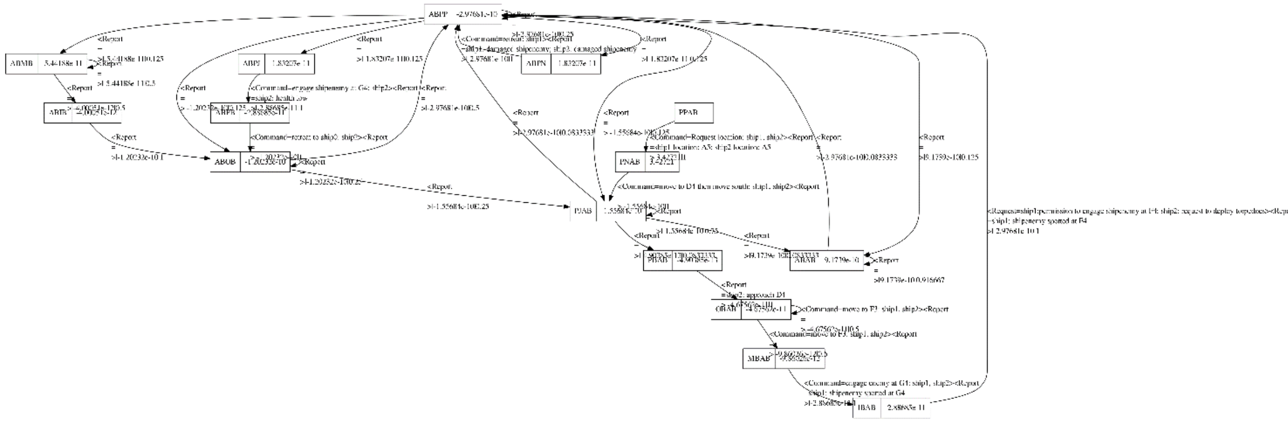

Figure 5: Commander 1's DTM for his first battle

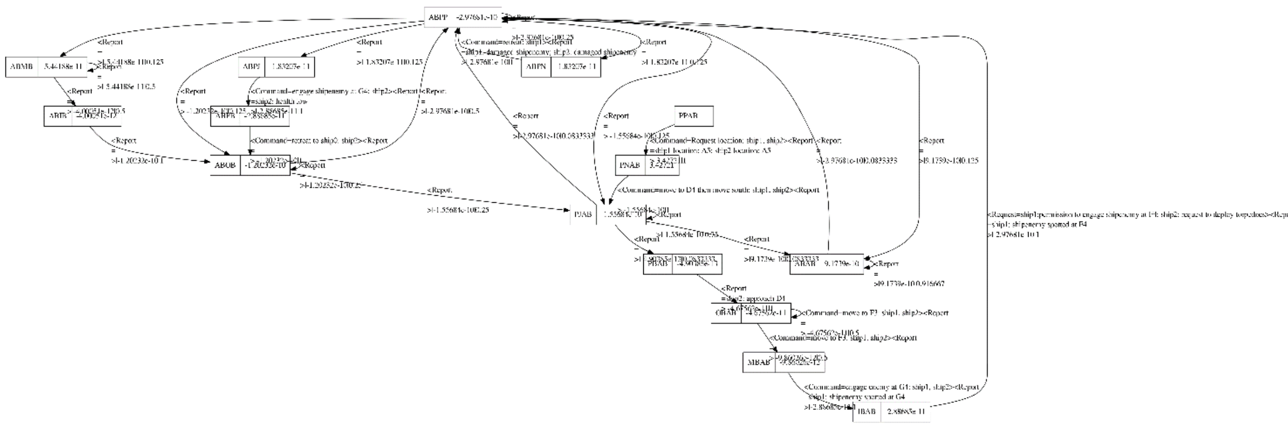

Figure 6: Commander 1's DTM for his second battle

Style	States	Actions	Density	Avg.Outgoin	Avg.Incoming
Commander 1(Intuitive)	25	23	0.08087	1.63908	1.63908
Commander 2(Spontaneous)	36	32	0.030612	1.06	1.06

Table 4: Capturing real users' decision making styles

As we could see from the figures, decision making styles were fairly consistent among Commanders. Despite commanding two different battles, both Commanders maintained their styles. The number of average states and actions of Commander 1 were significantly less than those of Commander 2 and the structures of their DTMs showed that the Commander 2's DTMs were more linear than the DTMs of Commander 1. For Commander 1, his two battles had lengths of 10 minutes 39 seconds and 6 minutes 29 seconds while for Commander 2, his two battles had lengths of 11 minutes 15 seconds and 5 minutes 32 second respectively. We further checked with the video recordings to confirm that the Commander 2 had used his previous experience extensively in games which resulted in a fewer cognitive states because he was able to recognize similar situations in games. Whereas the Commander 2 was more reactive in games which resulted in more states because he was unable to relate with his previous experience.

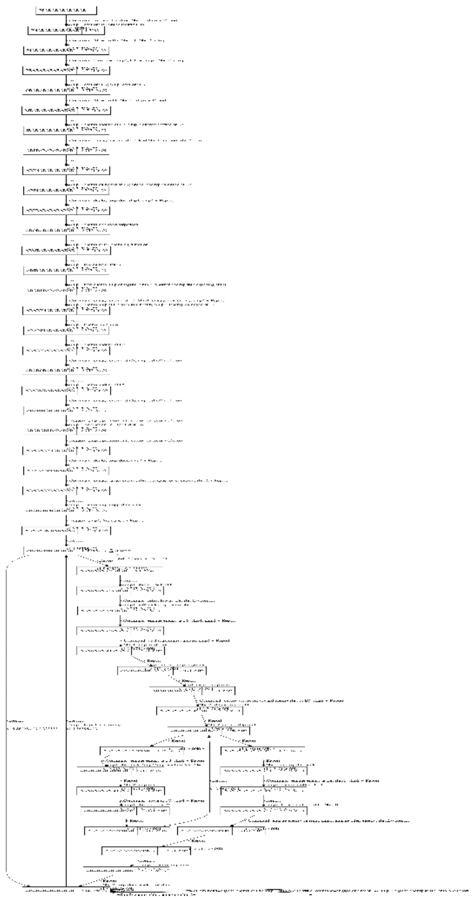

Figure 7: Commander 2's DTM for his first battle

Figure 8: Commander 2's DTM for his second battle.

4.4 Reward distribution with real users.

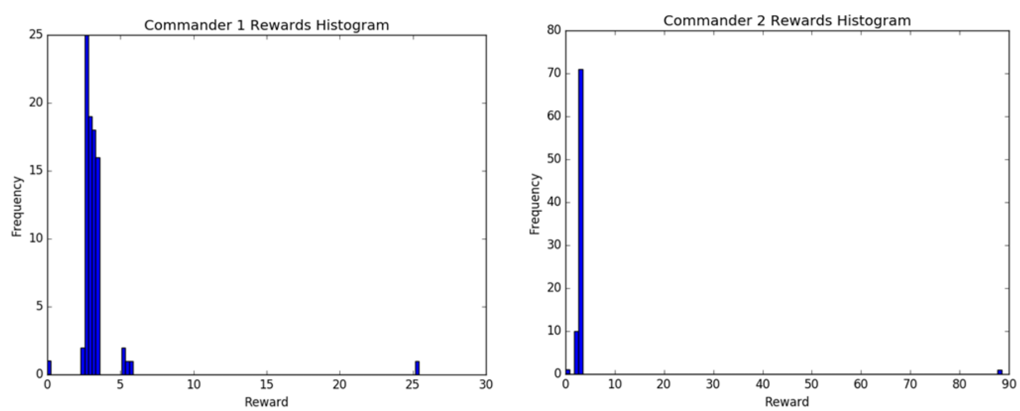

Figure 9: Reward distribution for real users

Similarly to what we have done with hypothetical users, we computed the reward distribution for both Commander 1 and Commander 2 using the same procedure addressed in Section 4.2. As we could see in Figure 8, their rewards for each cognitive state seems to be clustered around zero for Commander 2 while they are clustered around the midpoint between zero and 5 for Commander 1. We have only two traces for each Commander over their two battles and unfortunately, it's not enough data points to judge whether the rewards are significant or not. We are unable to compute the game reward and DTMs correlations due to the lack of data points. This will be addressed in our future work.

5. CONCLUSION

We have applied our DTM approach to capture a Commander's decision making process and evaluated the DTMs to distinguish different decision making styles from different Commanders. Our approach allowed us to look at not only the context of the Commander's decision that could potentially infer why he makes that specific decision but also the distribution of rewards that go along with each cognitive state where the decision is made and the relative ordering of preferences for each of the traces. This provides a concrete background to evaluate the quality of any decision or trace by using the contextual information associated with cognitive states and offers an effective tool to see how influence a Commander's decision making styles would affect the quality of his decisions. By capturing a Commander's decision making process, we can distinguish different decision styles and study the distribution of rewards over time. Additionally, we could compute the correlation between game rewards and rewards inferred by inverse reinforcement learning and compute correlation between different Commanders' DTMs. These results could be used for training purposes when a senior DTM is used to guide training process for a junior staff member.

Although we have laid concrete ground work for the development and assessment of this framework, there remains many areas of interesting research to explore. First, we would like to study further how the graph structure of our DTMs changes as users change their decision styles over time. This means that, in a battle, a Commander may have a sequence of actions where he is acting spontaneously but also has a sequence of actions where he is acting intuitively. A DTM could capture these changes and that could be used to distinguish dynamic decision styles over time. Second, we also want to strengthen our evaluation with real users by comparing our decision style classification with the traditional approach that uses a questionnaire survey to determine a user's decision style from among the general decision styles. Lastly, we would like to find out whether the reward distribution plays a significant role in reinforcing good decisions, enabling us to learn from bad decisions.

6. ACKNOWLEDGEMENT

Thank you to UWW student volunteers Luke De Guelle and Devan Wolfmeyer for participating in our scenario creation. This work was sponsored in part by ONR Grant No. N00014-15-1-2154 and DURIP Grant No. N00014-15-1-2514.

7. REFERENCES

[1] International Committee of the Red Cross. [Decision-Making Process in military combat operations]. ICRC, Geneva, Switzerland (2013).
[2] United States Joint Forces Command. [Commander's Handbook for Joint Battle Damage Assessment] (2004). Retrieved from http://www.dtic.mil/doctrine/doctrine/jwfc/hbk_jbda.pdf
[3] Scott, S. G, & Bruce, R. A. "Decision-making style: The development and assessment of a new measure." Educational and Psychological Measurement, 55(5), 818-831 (1995).
[4] Bavol'ár J. and Orosová, O. "Decision-making styles and their associations with decision-making competencies and mental health Judgment and Decision Making," Vol. 10, No. 1, January 2015, pp. 11. 5–122 (2015).
[5] Flin, R. "Decision making in crises: the Piper Alpha disaster." 103-116. In MANAGING CRISES: Threats, Dilemmas, Opportunities (2001).
[6] Wood, N. "Individual differences in decision-making styles as predictors of good decision making." Master thesis. Bowling Green State University (2012).
[7] Syagga, L. A. "Intuitive Cognitive Style and Biases in Decision Making." Master thesis. Eastern Mediterranean University (2012).
[8] Wang, Y., & Ruhe, G. "The Cognitive Process of Decision Making." International Journal of Cognitive Informatics and Natural Intelligence (IJCINI), 1(2), 73-85 (2007).
[9] Kaur, N. and Sood S. K. "Cognitive decision making in smart industry." Computers in Industry 74 151-161 (2015).

[10] Yu, F. “A Framework of Computational Opinions” (Doctoral Dissertation), Dartmouth College, Hanover, NH (2013).
[11] Yu, F. & Santos, E. Jr. “On Modeling the Interplay between Opinion Change and Formation,” in Proceedings of the 29th International FLAIRS Conference, Key Largo, FL 140-145 (2016).
[12] McLaughlin, J. E., Cox, W. C., Williams, C. R., and Shepherd, G. 2014. “Rational and experiential decision-making preferences of third-year student pharmacists.” American journal of pharmaceutical education 78.6 (2014): 120.
[13] Esa, M., Alias A., and Samad, Z. A. “Project Managers' Cognitive Style in Decision Making: A Perspective from Construction Industry." International Journal of Psychological Studies 6.2 (2014): 65.
[14] Michailidis, E. and Banks, A. P “The relationship between burnout and risk-taking in workplace decision-making and decision-making style.” Work & Stress Vol. 30, Issue. 3 (2016).
[15] Santos, E., Jr., & Santos, E. S. A framework for building knowledge-bases under uncertainty. *Journal of Experimental & Theoretical Artificial Intelligence*, *11*(2), 265-286 (1999).
[16] Santos, E. Jr., Nguyen, H., Russell, J., Day, J., Pegelow, S., Greenstreet, A. “A Framework for Dynamic Context-Centric Commander Decision Support.” Proceedings of the Human Factors and Ergonomics Society Annual Meeting (2016).
[17] Ng, A. Y., and Russell, S. J. "Algorithms for inverse reinforcement learning." Icml. (2000).
[18] Blei, D. M., Ng, A. Y., & Jordan, M. I. “Latent dirichlet allocation.” The Journal of Machine Learning Research, 3, 993-1022 (2003).
[19] Bellman, R. E. “A Markovian Decision Process.” Ind. Univ. Math. J. 6(4), 679–684
[20] Abbeel, P. and Ng, A.Y., 2004, July. “Apprenticeship learning via inverse reinforcement learning.” In Proceedings of the twenty-first international conference on Machine learning (p. 1). ACM (2004).
[21] Ziebart, B. D., et al. "Maximum Entropy Inverse Reinforcement Learning." AAAI. Vol. 8. (2008).
[22] https://github.com/MatthewJA/Inverse-Reinforcement-Learning