# A Framework for Dynamic Context-Centric Commander Decision Support

Eugene Santos Jr[1], Hien Nguyen[2], Jacob Russell[1], Joshua Day[2], Scott Pegleow[2], Alexis Greenstreet[2]

[1]Thayer School of Engineering
Dartmouth College
8000 Cummings Hall,
Hanover, NH 03755

[2] Department of Computer Science
University of Wisconsin-Whitewater
800 W. Main Street
Whitewater, WI 53190

## ABSTRACT

This paper addresses the fundamental research question: "How can we determine the sequential decision-making process inside a decision maker's mind?" We construct a dynamic Markov Decision Process using a Double Transition Model (DTM). The DTM is a cognitive model decomposing the decision-making process into episodic tasks that are extracted from a stream of incoming information. In a DTM, each state reflects a stage en route to a decision, and each action reflects a possible move from collecting data to hypothesizing and inferencing. The reward reflects how close a stage is to the final decision. We demonstrate this process through a proof-of-concept DTM using a hypothetical scenario for Typhoon Haiyan in the Philippines (2013). The DTM constructed from this scenario enables a Commander to reason about damaged areas, death tolls, and assistance methods while allowing his actions to be captured and used to explain why and how each decision is made.

## INTRODUCTION

The key for the successful development of a proactive decision support system (PDS) is to determine *what* the Commander wants and *why* from his goals, actions taken, and any relevant information. This process requires the system to understand the decision-making goals of the Commander in order to anticipate the Commander's needs while making decisions. In a PDS, the context that is central to the different decision trajectories is constructed to support the Commander. Unfortunately, the existing technologies at the center of a PDS do not fully address this problem, namely Decision Modeling (DM) and User Modeling (UM). UM techniques focus on the activities, information seeking, and cognitive behaviors of the user, whereas DM techniques focus on how decisions are made. Even though all of these foci affect the decision-making process, up to this point they have not been integrated and their interactions have not been considered in a unified framework. This leads to a considerable gap between the two research areas of DM and UM. In this paper, we set out to address this gap, which leads to the fundamental research question, "How can we determine the sequential decision-making process in the Commander's mind?" We propose to model the Commander's decision-making process over time by using Double Transition Models (DTM) (Yu, 2013; Yu and Santos, 2016). A DTM can be used to derive a Markov Decision Process (MDP) in which each state reflects a stage on the way to a decision, and each action reflects a possible move, from collecting data to hypothesizing or from hypothesizing to inferencing. Finally, the reward reflects how close a particular stage is to the final decision. The novelty of our approach is that we model the process of decision-making as a sequence of events that includes information-seeking actions, which are themselves decisions that are being taken. With a model of this sequential and episodic decision-making process, the information-seeking behavior itself can now be placed into context– that is, the UM is put in the context of DM, and DM's need to account for UM is fulfilled.

We demonstrate this process through a proof-of-concept DTM using information from a hypothetical disaster rescue and relief effort scenario for Typhoon Haiyan in the Philippines (2013). The DTM constructed from this scenario allows Commanders to reason about damaged areas, death tolls, and assistance methods, and also allows his actions to be tracked and used to explain why and how each decision is made. This framework addresses the gap between DM and UM at the center of PDS by using contextual information about what the Commander's decision-making process looks like, and provides a concrete structure for explaining his decisions in the future.

This paper is organized as follows. We begin by reviewing key related work with regards to constructing this framework. Next, we describe our approach. We then describe the scenario, followed by a brief discussion. Finally, we present our conclusions and future work.

## RELATED WORK

The novelty of our approach is to construct a *dynamic* model that integrates a user's interests and keeps track of sequences of actions in his decision-making process. We use User Modeling and dynamic Markov Decision Processes (*d*MDP) in our approach. Related work for these areas is presented in this section.

User Modeling techniques have been used to enhance the major tasks involved in the cognitive process of decision making (Wang and Ruhe, 2007). The major foci in this area include determining a user's goals, identifying and quantifying a user's choices, evaluating choices, and making decisions. Furthermore, a tremendous amount of effort has been spent on understanding a user's goal or intent in a Web search or a general information-seeking task through information retrieval and knowledge representation techniques (e.g., Nguyen and Santos, 2013; Santos and Nguyen, 2009; Chuklin et al., 2013). User studies about personality, cognitive styles, and domain expertise, and even the emotion and motivation of the user, have been conducted in the cognitive psychology community to provide evidence of their respective effects on the decision-making process (e.g., Poore et al., 2014; Scribner, 2015; Drew et al., 2015). User models are also used to build adaptive eLearning applications in which a student's profile drives his navigation through domain knowledge (see (Truong, 2016) for a comprehensive review on eLearning systems). Unfortunately, there is a considerable gap between current user studies in cognitive engineering and computational models like the ones we are developing due to limitations on knowledge elicitation, representation tools, and differences in evaluation approaches. Our approach enables us to model the Commander's behaviors over time and *learn* from his sequential actions and decisions to proactively assist him in future decision-making tasks.

In the DM community, the Markov Decision Process (MDP) (Bellman, 1957) is a commonly used model of the decision-making process in uncertain environments. A MDP is composed of a set of states representing the world ($S$), a set of actions ($A$) that can be taken, a reward function ($R(s, a)$), and a set of possible transitions ($T: S \times A \rightarrow P(S)$) mapping state and action pairs onto another state with a probability. The goal is to find a set of actions to take in each state in order to maximize the reward (an optimal policy). Reinforcement learning is often applied to solve MDPs where the transition probabilities are unknown by leveraging traces of a policy (Sutton and Barto, 1998; Ng and Russell, 2000). Among other areas, MDPs have been applied to such military applications as air campaign planning (Meuleau et al., 1998); for a detailed review of MDPs, please see Feinberg and Schwartz (2012). At the current time MDPs have not been used to model individuals, but rather to identify policies that do not require individualization. Additionally, the set of states in MDPs are always defined a priori. Unfortunately, predefined states are not possible in temporal and dynamic scenarios where concepts and transitions between these concepts may not exist until a later point in time. Therefore, our approach requires us to model actions and states which may have never been seen before and cannot be predicted in advance.

# APPROACH

Inferring *what* information the Commander wants and *why* he wants it are the keys to a successful PDS, as identifying and explaining a Commander's objectives and intent help fill the gap between UM and DM. The interaction between what a user wants and why he wants it cannot be separated because the motivation is a reason for taking actions to achieve something desired. Therefore, a unified cognitive decision model must have UM to identify the actions that the Commander takes and DM to explain why he made the decision. Simply put, actions taken by the DM impact representations, biases, and preferences in the UM, and in turn, preferences, biases, and representations in the UM impact which actions are preferred in the DM. In our cognitive decision model, context *represents* a Commander's internal cognitive state, and is *determined* from relevant, externally available information that is central to a Commander's decision-making process. The information includes which documents he retrieves and processes, which actions he has taken, and what goals he is currently pursuing. Our model enables us to explain why a Commander is making a certain decision by leveraging traces that contain a sequence of actions and relevant information to those actions.

## Architecture

The basis of our approach is to construct a Double Transition Model (DTM) (Yu and Santos, 2016) using traces of the Commander's information streams, including incoming reports, documents, and requests over time. The DTM describes the Commander's cognitive state and what he focuses on throughout the history of his decision-making process. It will not prescribe which actions the Commander should take next. In order to prescribe which actions the Commander should take next, we need to derive a dynamic MDP (*d*MDP) from the DTM. In this *d*MDP, it is possible to estimate a reward function that describes the Commander's unique style and method for making decisions. In the next section we describe the construction of the DTM, the conversion of the Commander's information stream to features accessible by the DTM, and the mapping of the DTM onto a *d*MDP.

## Double Transition Model (DTM)

The Double Transition Model (DTM) was originally created for modeling the formation of human opinion (Yu and Santos, 2016). In this paper, we use the DTM to model the human decision-making process, since opinion formation also involves decision-making processes. Each state in a DTM is the cross-product of two subgraphs, a Query Transition Graph (QTG) and a Memory Transition Graph (MTG).

Each node in the QTG represents a single query or a question that the Commander focuses on. Each query is represented as a vector $[X, ?, c_1, d_2, \dots]$ where $X \in U$ represents the target random variable of interest, $U$ represents the entire universe of random variables, $? \in U$ represents that the instantiation of a variable is unknown, and $c_1$ and $d_2$ represent the states that the random variables $C, D \in U$ take on. The set of all possible queries is denoted by $\nabla$. A function $F_q: q_1 \rightarrow q_2$ where $q_1, q_2 \in \nabla$ is called the query transformation function (*qtf*), which tells how to transition between states in the QTG. Each task has its own query space $\varphi(t_i) = \langle q(t_i), T_q(t_i) \rangle$ where $q(t_i)$ is a finite set of queries and $T_q(t_i)$ is a finite set of *qtfs*.

**Def 1. (Yu and Santos, 2016):** A *QTG* $\boldsymbol{Q}$ is an undirected graph $(\boldsymbol{V^Q}, \boldsymbol{E^Q})$where $\boldsymbol{V^Q}$ is a finite subset of $\bigcup_{\boldsymbol{i}} \boldsymbol{q}(\boldsymbol{t_i})$ and $(\boldsymbol{v_1^Q}, \boldsymbol{v_2^Q}) \in \boldsymbol{E^Q}$ only if $\boldsymbol{F_q}(\boldsymbol{v_1^Q}) = \boldsymbol{v_2^Q}$ for some $\boldsymbol{F_q} \in \boldsymbol{T_q}(\boldsymbol{t_i})$ and $\boldsymbol{i}$.

Query transformations represent the changes in the Commander's focus. The formulation of the query lends a potential structure for how the Commander represents information in his underlying knowledge base. Queries may be input by the commander, or inferred by assuming a base structure and generating queries to match the structure.

The MTG stores the underlying knowledge of the Commander and how its features relate in a probabilistic network called a Bayesian Knowledge Base (BKB) (Santos and Santos, 1999). Let $\Xi$ be the space of probabilistic networks over $U$ that form the memory of a Commander. A memory transformation function (*mtf*) $F_m: m_1 \to \mathrm{m}_2$ for $m_1, m_2 \in \Xi$ represents how memory changes, either by changing the parameters or structure of the underlying knowledge base. Let $\Gamma = (K, T_m)$ be the memory space where $K$ is a finite set of probabilistic networks and $T_m$ is a finite set of *mtf*s operating on $K$.

**Def 2. (Yu and Santos, 2016):** A *MTG* $\boldsymbol{K}$ is an undirected graph $(\boldsymbol{V^K}, \boldsymbol{E^K})$ where $\boldsymbol{V^K}$ is a finite set from $\boldsymbol{K}$ and $(\boldsymbol{v_1^K}, \boldsymbol{v_2^K}) \in \boldsymbol{E^K}$ only if $\boldsymbol{F_m}(\boldsymbol{v_1^K}) = \boldsymbol{v_2^K}$ for some $\boldsymbol{F_m} \in \boldsymbol{T_m}$.

MTGs and QTGs combine to form the DTM. In the DTM a transformation from one state to another represents a change in what the Commander is interested in or what changes occurred in his cognitive state.

**Def 3. (Yu and Santos, 2016):** A *DTM* $\boldsymbol{D}$ induced by QTG $\boldsymbol{Q}$ and MTG $\boldsymbol{K}$ is the undirected graph $(\boldsymbol{V^D}, \boldsymbol{E^D})$ where $\boldsymbol{V^D} = \boldsymbol{V^Q} \times \boldsymbol{V^K}$ and there is an edge between $\boldsymbol{v_1^D} = (\boldsymbol{v_1^Q}, \boldsymbol{v_1^K})$ and $\boldsymbol{v_2^D} = (\boldsymbol{v_2^Q}, \boldsymbol{v_2^K})$ if and only if (1) $\boldsymbol{v_1^Q} = \boldsymbol{v_2^Q}$ or $(\boldsymbol{v_1^Q}, \boldsymbol{v_2^Q}) \in \boldsymbol{E^Q}$ and (2) $\boldsymbol{v_1^K} = \boldsymbol{v_2^K}$ or $(\boldsymbol{v_1^K}, \boldsymbol{v_2^K}) \in \boldsymbol{E^K}$.

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 thus to reconstruct the sequence of decisions that the Commander made in order to predict future decisions by leveraging repeating patterns.

As an example of the DTM, a Commander may have received new information that an Earthquake hit Bohol in the Philippines. The Commander could undergo a memory transformation by adding a feature representing the possibility that Bohol may be damaged into his underlying MTG that represents the Commander's knowledge. Then he could undergo a query transformation, asking a question such as "Is Bohol Damaged?" This process can be seen in Figure 1.

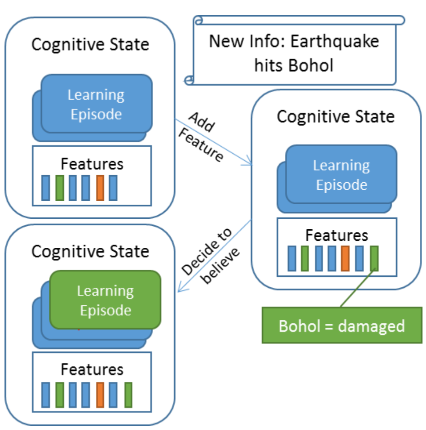


*Figure 1. An example of the DTM framework.*

The random variables (such as 'Bohol' and 'Earthquake') and their relationships in the DTM and its underlying QTG and MTG are unknown at first and must be identified and extracted from the Commander's information stream in order to incorporate them into the MTG representing the Commander's memory.

## Information preprocessing

Our model is built from a set of relevant features that are defined as distinctive attributes for the Commander's decision-making process extracted from his information stream. In order to extract the features automatically from a source of incoming unstructured documents we convert the text into two structured formats: Document Graphs (DG) (Santos and Nguyen, 2009) and vectors containing the main topics of these documents.

A Document Graph is a graph consisting of concept nodes and relations between them. Concept nodes represent entities, and relations are either "isa" for subsumption relations between entities or "related_to" for any other relationship between entities. A sample DG for "A magnitude 7.2 earthquake hit Central Visayas on Tuesday" is shown in Figure 2.

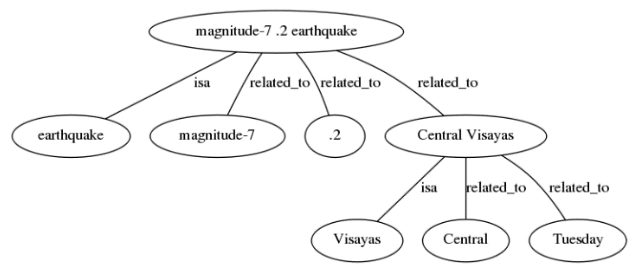


*Figure 2. Sample Document Graph*

We also use Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to generate topics from the corpus of relevant documents compiled in the Commander's information stream. Topics are vectors of semantically related words. The topics and terms in each topic give us *general* summaries of the main topics from these incoming documents while document graphs fill in the detailed information of main concepts. We combine both LDA and DGs into our feature extraction algorithm.

## Feature extraction

A Commander's decision and action space is uncertain and constantly changing. Making decisions in such a space requires automatic identification and extraction of features that are relevant to current and future decisions. The value of a feature is a specific, valid, and observed state of that attribute. Therefore, extracting feature-value pairs from unstructured and structured information is a crucial part of this model. A flowchart of extracting features and values from unstructured incoming documents is shown in Figure 3. These features are utilized in the *d*MDP, the DTM, and the underlying BKBs.

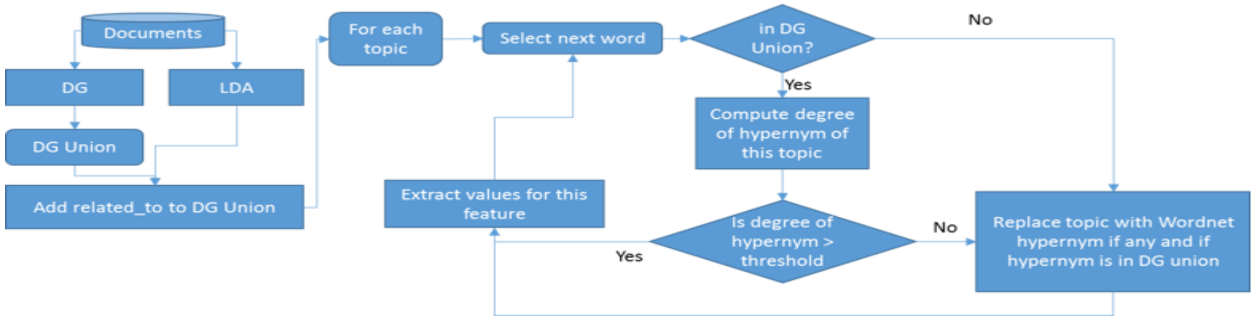


*Figure 3. Feature extraction flowchart.*

## Bayesian Knowledge Bases

In order to be able to answer queries in the DTM, and thus make individual decisions or choose actions, we must be able to reason over the feature space. A Bayesian Knowledge Base (BKB) (Santos and Santos, 1999) provides a convenient method to reason over the feature space using if-then rules with associated probabilities. The structures and probabilities must both be learned from the documents and traces of the Commander's actions. The QTG in the DTM provides the structures. For a given query such as, "[$X = Bohol, Earthquake\ Magnitude = 5.2, ?$]", where the features were extracted from text, we use the uniform distribution to select probabilities and get the BKB depicted in Figure 4. Note that this is only a single possible state represented by the query, and all features that exist for an unknown variable (denoted as "?" in the above query) (e.g. "Attack = Moro National Liberation Front") could have been selected instead (including no instantiation, as pictured), changing the structure of the BKB in figure 4.

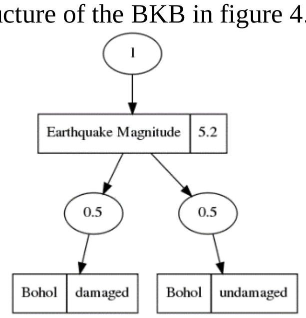


*Figure 4. BKB for query [X=Bohol, Earthquake Magnitude=5.2,?]*

A formed query is looking for a relationship among features. Choosing to include features in a query implies that there is an expected probabilistic relation between the target and the features. We both extract the queries from the Commander's information-seeking actions and also build the memories from documents in the Commander's history. The ability to transition between cognitive states and also keep all subsequences as episodes in the QTG and MTG allows us to identify more probable memory structures that are consistent with the Commander's behavior because we can simply select the best subsequent decision when we derive the *d*MDP.

## Dynamic Markov Decision Processes

In the paper by Yu and Santos (2016), the authors derived a mapping from the DTM, goals, and actions onto a MDP for an individual opinion formation task. Their process maximizes a goal function and reward function while minimizing the change to the knowledge base of a decision maker. In our case, as new information comes in and a Commander makes decisions, his feature space must be robust to inaccuracy and information changes. We instead use a *d*MDP to account for this dynamic feature space that is presented to a Commander. The MDP is constructed and solved for each of the finite feature sets at a given point in time. The probabilities of transitions in the *d*MDP mirror the probabilities of transitions within the DTM, which were sampled from the Commander's own history. The Commander's history is based on his decisions, and the reward function maximizes the policy that best matches his decisions while simultaneously maximizing the consistency of the memory transition graph. This reward function is the key insight into modeling a Commander's decision-making behaviors because traces of his decisions encompass his unique decision-making style. Our future goal is to solve the *d*MDP to predict both the decisions that the Commander is likely to encounter and which choice he is likely to make.

## SCENARIO DISCUSSION

To demonstrate our approach, we chose the disaster rescue and relief effort for Typhoon Haiyan in the Philippines in 2013. We collected and compiled relevant documents about this disaster, including press releases and Department of Defense reports (including US Marine reports) and arranged them into a timeline. Since we have to synthesize the Commander's timeline, the documents have been filtered so that only the material highly correlated with the tasks at hand are delivered to our system. We argue that this is realistic because the documents that a Commander selects would be expected to be correlated with the task at hand. We also assume that the tasks related to the decisions made by Commanders and the Commander's cognitive states are changing over time. The features will be automatically extracted from semi-structured documents and used to construct the DTM. A memory transition graph of the DTM contains a set of nodes, each of which is described by a BKB created from the features obtained at each time slice (as shown in Figure 5(a)). Similarly, the QTG is created from features extracted based on a Commander's requests over time (shown in Figure 5(b)). The random variables $A, B, C$, and $D$ represent the features while $a_i, b_i, c_i$, and $d_i$ represent the corresponding values. This DTM allows users to ask questions that are defined over the features in the DTM. Questions may relate to different probabilities acquired through reasoning in the underlying BKBs. For example, the BKB may reveal such questions as "Which region is more damaged, Tacloban or the Six Islands?", "Which region has more damage among those where more than 1,000 people were killed?", or "Is Bohol still damaged from the Earthquake?" We refer to the features that we believe to be true as evidence. These questions are answered through solving the *d*MDP that ultimately learns to create a policy rewarding the decisions that the Commander made and recommends transitions in the DTM. At this time, a belief update is performed on a BKB at each node computing posterior distributions of the features. For example, through the BKBs we are able to find that the answer to the first of the aforementioned questions that the six islands had a higher probability of being damaged at time $t_1$ but had a relative lower probability of damage at time $t_4$ as incoming information changed and updated the response of the system. Testing with our scenario validates that we can model a user's actions and behaviors in a decision-making process. Finally, while we do not currently have learning modules in the model to estimate a user's cognitive styles and patterns in the decision-making process for prediction, we will develop this ability in the future.

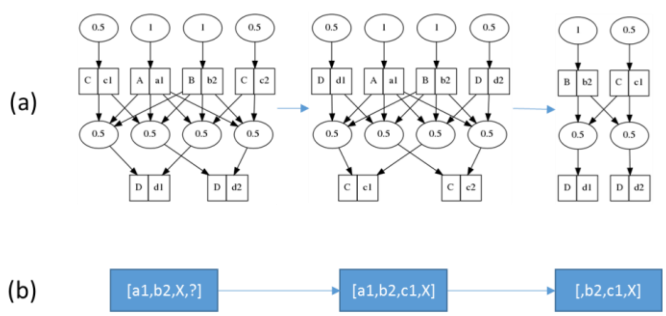


*Figure 5 (a) Memory Transition Graph (b) Query Transition Graph*

## CONCLUSION

We have demonstrated a framework for providing proactive decision support to Commanders in an uncertain, changing, temporal information space. Our framework bridges a gap between the user modeling and decision making communities by capturing the context of a Commander's decision-making process and reasoning over the feature spaces. This provides a concrete architecture for future work including explaining why a specific decision is made or incorporating cognitive styles into the decision-making process.

Solving the models in this framework allows us to reconstruct the most likely decision process the Commander took to reach a conclusion. One of the challenges in the evaluation of this framework is limited availability of data that tracks the process of Commanders' behaviors. We are in the process of constructing multiple synthetic Commander's action sequences using data from real-world events to further validate the model. Our future work includes the use of inverse reinforcement learning to learn the Commander's policy and reward function simultaneously. This will enable us to predict a Commander's decisions and actions through a model of his underlying process of decision making in order to provide relevant information and proposed courses of action even before being asked. Future work also involves quantifying the growth rates of the model.

## ACKNOWLEDGEMENTS

This work was sponsored in part by ONR Grant No. N00014-15-1-2154 and DURIP Grant No. N00014-15-1-2514.

## REFERENCES

Bellman, R. E. (1957). A Markovian Decision Process. *Ind. Univ. Math. J. 6*(4), 679–684.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993-1022.

Chuklin, A., Serdyukov, P., & De Rijke, M. (2013). Using intent information to model user behavior in diversified search. In *Proceedings of the 35th European conference on Advances in Information Retrieval (ECIR'13),* Pavel Serdyukov, Pavel Braslavski, Sergei O. Kuznetsov, Jaap Kamps, and Stefan Rüger (Eds.). Springer-Verlag, Berlin, Heidelberg, 1-13.

Drews, F., Siebeneck, L., & Cova T. (2015). Information Search and Decision Making in Computer-Based Wildfire Simulations. *Journal of Cognitive Engineering and Decision Making*. 9: 229-240.

Feinberg, E. A., & Shwartz, A. (Eds.). (2012). *Handbook of Markov decision processes: methods and applications* (40). Springer.

Meuleau, N., Hauskrecht, M., Kim, K. E., Peshkin, L., Kaelbling, L. P., Dean, T. L., & Boutilier, C. (1998). Solving very large weakly coupled Markov decision processes. In *Proceedings of the fifteenth national/tenth conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence (AAAI '98/IAAI '98)* (pp. 165-172).

Ng, A. Y., & Russell, S. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00), Pat Langley (Ed.).* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 663-670. ).

Nguyen H. & Santos E., Jr. (2013). Hybrid User Model for Capturing a User's Information Seeking Intent. In *Multimedia Services in Intelligent Environments. 24*, pp 31-63.

Poore, J. C., Forlines, C. L., Miller, S. M., Regan, J. R., & Irvine, J. M. (2014). Personality, cognitive style, motivation, and aptitude predict systematic trends in analytic forecasting behavior. *Journal of cognitive engineering and decision making*, *8*(4), 374-393.

Santos E., Jr., & Nguyen, H. (2009). Modeling Users for Adaptive Information Retrieval by Capturing User Intent. In *Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling*. IGI Global.

Santos, E., Jr., & Santos, E. S. (1999). A framework for building knowledge-bases under uncertainty. *Journal of Experimental & Theoretical Artificial Intelligence*, *11*(2), 265-286.

Scribner, D. R. (2015). Predictors of Shoot–Don't Shoot Decision-Making Performance An Examination of Cognitive and Emotional Factors. *Journal of Cognitive Engineering and Decision Making*.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT press.

Truong, H. M. (2016) Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities, Computers in Human Behavior, Vol. 55, Part B, Pages 1185-1193, ISSN 0747-5632.

Wang, Y., & Ruhe, G. (2007). The Cognitive Process of Decision Making. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, *1*(2), 73-85.

Yu, F. (2013). *A Framework of Computational Opinions* (Doctoral Dissertation), Dartmouth College, Hanover, NH.

Yu, F. & Santos, E., Jr. (2016). On Modeling the Interplay between Opinion Change and Formation, in *Proceedings of the 29th International FLAIRS Conference*, Key Largo, FL (pp. 140-145).