An Instance of What?
Determining Cues, Attributes, and Goals in a Virtual Task Environment

Bradley J. Best
Nathan Gerhart
Adaptive Cognitive Systems, LLC
1709 Alpine Avenue
Boulder, CO 80304
303-413-3472
{bjbest, ngerhart} @ adcogsys.com

Keywords:
Virtual Environment, Avatar, Spatial Representations, Instance-Based Learning, Cue Learning

ABSTRACT: *The construction of instance-based cognitive models is often advocated as a way to reduce the complexity of modeling. Through an in-depth case study of modeling driving and adversarial behavior in a virtual environment, we show that the representation of the attributes and cues that make up instances can be enormously complex. Eye tracking and Cognitive Task Analysis are employed as methods to identify the key attributes, and their utility is demonstrated through automated goal recognition and driving performance in a virtual domain.*

1. Introduction

Cognitive models are often developed in one of two ways: 1) through extensive task analysis and knowledge engineering to produce rule-based systems, or 2) by inducing performance models from the data, using paradigms known as case-based reasoning, instance-based modeling (Aha, 1991), or learning from examples (Simon & Zhu, 1988; Simon & Gobet, 1996). Recently, the ACT-R cognitive architecture (Anderson et. al., 2004) has been used as a platform to develop models using this methodology (Taatgen & Wallach, 2002; Gonzalez, Lerch, & Lebiere, 2003; Best & Lovett, 2006). Equipped with a bootstrapping mechanism (e.g., the ability to randomly guess on initial trials), an instance-based ACT-R model can operate in an environment, learning as it performs the task (e.g., Best, Dixon, & Speed, 2008).

Our aim was to evaluate the feasibility of using instance-based modeling to induce a cognitive model from a recorded performance of a subject matter expert (SME), as demonstrated in a virtual environment. The SME piloted a virtual ground vehicle, a tank with an independent turret and main chassis, and performed a variety of tasks that included combat, seeking cover, scouting terrain, and driving a twisty forested path. We translated the continuous behavior of an SME performing a task in a virtual environment into a set of discrete instances to create the learning examples. These instances could include any of a number of ground truth attributes from the simulation environment, or derived attributes. Extracting and determining which to use is the significant technical challenge that this report will focus on.

2. Task Environments

Virtual environments can place severe demands upon computational resources, making it challenging to develop cognitive models capable of acting within real-time constraints (Best & Lebiere, 2006). In particular, agent behavior in the environment becomes unacceptable using cycle times of more than 200ms, while less than 100ms tends to produce smoother behavior, and a response of less than 50ms is required to achieve targeting behavior, for example. This cycle time is somewhat independent of the data sampling rate, which needs to be frequent enough to capture snapshots of the performance, but need not necessarily be as rapid as the cycle time for performance.

We worked with the dTank virtual environment (Ritter, 2008; Figure 1) and the Unreal Tournament game engine for task performance and data collection.

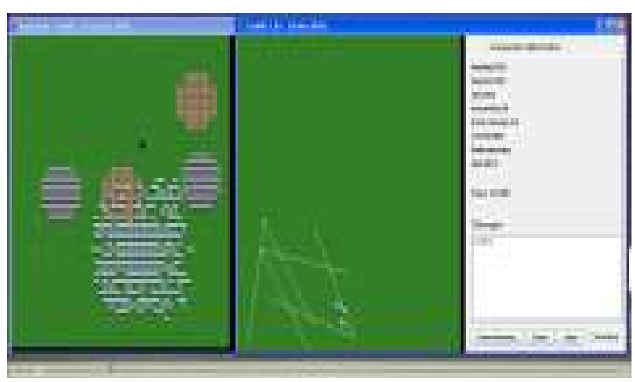


Figure 1: A screen capture of both the omniscient (left) and commander views (middle) of a dTank battle

3. Collecting and Segmenting Data from Human Task Performance

We collected continuous data in two different ways: 1) we collected behavioral streams from the SME during battle simulations in the dTank domain, and 2) we collected a series of scripted behaviors in which the actions were more constrained. For both of these situations, the data were automatically segmented offline into instances based on the dTank message frequency of 5Hz. This frequency interacts with the rate of responses when they are discrete – at a very high sampling frequency, almost no conditions would result in a response. To account for this, responses that occurred between sample points (during the 200ms interval) were pushed forward to the next sample point.

The majority of data collected were complete battle simulations in which the SME participated in a 1,000 second simulation against three enemies (two tanks and one officer on foot). The SME behavior was analyzed into distinct goals and behavioral sequences through a Cognitive Task Analysis (CTA). We initially attempted to segment the battles into higher-level categories of either goals or behavioral sequences, but this rapidly became unrealistic; the time to hand-annotate 5,000 lines (per simulation) created a severe bottleneck in the data collection process. Ultimately, we decided to segment the behavior before data collection (details of one such segmentation – following a tree-lined path – is presented in section 6.2).

4. Determining the Task Structure

We asked the SME to play a round of dTank battles, which were recorded for later analysis. During these assessment battles, the SME was only able to see the typical 'commander view' window. However, for the purposes of illustration and analysis, the commander view and the 'omniscient view' were both recorded (see Figure 1). The omniscient view was obscured from the SME's sight during the battle trials. Each battle was segmented into discrete 'behavioral chunks', where each such behavioral chunk is identified by the elapsed battle time at which it began. This breakdown of each battle into discrete behavioral chunks was then later refined in the expert review and assessment of SME battle performance. A portion of an example CTA, from the beginning of Battle 1 until the first time the SME fires, appears in Table 1. It corresponds to Figure 1, a screen capture (with both omniscient and commander views) taken at the onset of behavioral chunk beginning 43200 milliseconds into the battle (the commander view is on the right, and includes the 'turret whiskers' of the two enemies and the human-controlled tank).

Time (ms)	Summary
0	BEGIN BATTLE
0	scout perimeter
3600	see Opponent
7000	start for cover, keep scouting
7600	see 2nd Opponent
9000	head for cover
12600	they see me, keep an eye out as seek cover
19400	good cover, wait to be loaded, monitor Opponent position
32400	It's been a while, seek Opponent location
43200	see Opponents again, they're still close together
45000	stay out of sight
54600	nearly loaded; take another peek; they don't see me
60400	loaded, select nearest Opponent as target
61200	FIRE

Table 1: Protocol Summary for the first 61200 milliseconds from Battle1.

A video screen capture of the SME battle performance was recorded for later analysis. Following the argument that expertise leads to simple yet sophisticated systems of representation, we revisited the video battles and attempted to categorize the dTank SME "behavioral segments" in a way that expressed the SME's battle representation not at the key-press level but at the intention level. For example, the SME would often remain as concealed as possible while continuing to peek out at an opponent he was stalking. This approach led to the identification of 7 discrete Tactic Categories that were then used to classify the SME behavior: 1) Fire At Opponent, 2) Monitor Opponent Position, 3) Scout Perimeter, 4) Seek Cover, 5) Seek Opponent Location, 6) Stay Out Of View, and 7) Target Opponent. These seven categories were determined based on a single rater's perception of the SME behavior and comments during the CTA. The SME reported engaging in multiple battle "tasks" simultaneously. With that in mind, we also explored categorizing each behavioral segment by at least one and as many as three hierarchically ordered Tactic Categories: Primary Tactic, Secondary Tactic and Tertiary Tactic.

Four complete SME battles were coded using the set of 7 Tactic Categories. Each battle consisted of approximately 50 individual behavioral segments, totaling 206 individual segments across all battles. Although the complete set of possible Tactic Category combinations (given a minimum of 1 and a maximum of 3 categories per chunk) is 259, only 27 actually came up in these 4 battles. Interestingly, only 5 of the 27 unique Tactic combinations were used in all 4 battles; 5 were used in 3 of the 4 battles; 7 were used in only 2 battles, and 10 (a full third) were unique to a single battle. Four battles is a relatively small data set, yet this analysis indicates that this methodology is in fact sufficiently robust to capture both the commonalities and the unique situations that appear across different battles in a meaningful yet succinct manner. We used the resulting task structure to annotate instances and determine if it was possible to predict goals from available cues (see results below).

5. Determining the Cue/Attribute Structure

The virtual tank commander should, given a set of circumstances, produce behavior that approximates that of the SME in the same circumstances. This depends on identifying the context: What is the SME seeing and attending to within the battle environment that elicits that specific behavior? One obvious source of context is the raw information provided by the dTank and Unreal environments. That information can easily be included as input into a virtual commander, including, for example, Speed, Heading, Opponent Type (Officer versus Tank), and Time. However, the CTA revealed that the SME filters and interprets environmental features in a way that far exceeds the simple consumption of raw data. Examples of these less direct – but at least equally important (as reported by the SME in the CTA) – attributes include proximity to cover from fire, and time until opponent is loaded. These 'cognitive' cues must be derived from the (simulation environment) ground truth.

5.1 Spatial Representation in Virtual Environments

An important aspect of context recognition is the ability to recognize specific terrain configurations that might help determine the appropriate course of action. One such attribute that the SME routinely utilized is the proximity to cover. To adequately simulate human behavior, a model must interact with the virtual environment in much the same way that the human does; this may entail some form of SME-like visual parsing of the environment by the synthetic commander. The next sections detail our efforts to understand and quantify the SME's visual parsing of the dTank environment and cues

This is of particular interest in that large areas of the terrain maps remain underspecified or unknown to the tank commander (whether SME or virtual agent) during battle. This is rather different from the majority of typical training environments where much of what is present is known to the operator or agent. Use of the dTank environment in this project therefore afforded a unique opportunity to investigate such situations that are relatively rare in test environments but relatively common in the real world. We believed that an investigation of the SME's representation of the dTank battle space would be an essential component underlying the development of a virtual tank commander. To that end, we proceeded with two tracks of investigation into the SME's visual representation of the dTank environment: Test Terrain Map Assessment, and Eye-tracking dTank trials.

5.2 SME Assessment of Test Terrain dTank Maps

The CTA revealed that the SME was very sensitive to the specific configuration of the terrain both leading up to and during engagements with opponents. This suggested that it might be necessary for the model to make SME-like assessments of the terrain. To enable this, we produced a set of 21 test terrain maps (see Figure 2) and asked the SME to assess the omniscient view of those terrain maps.

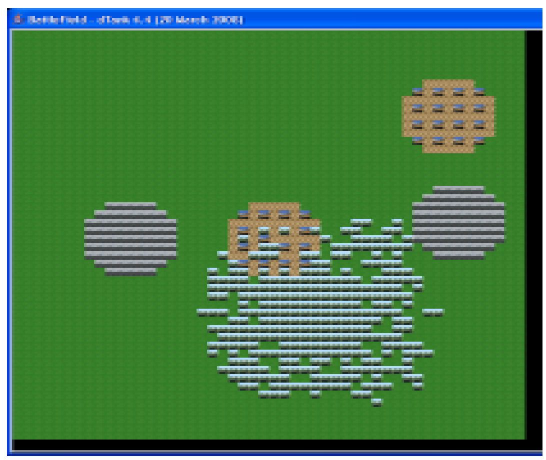


Figure 2: Example test terrain map

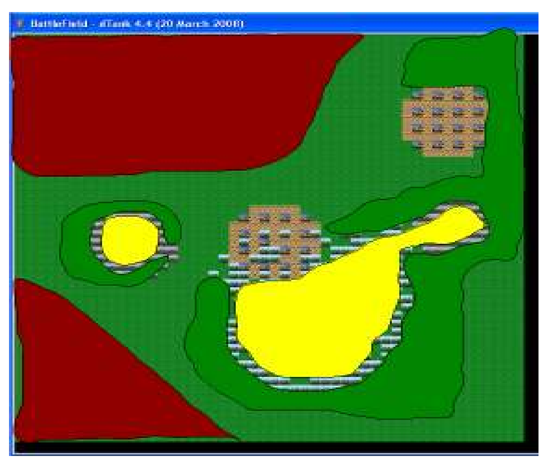


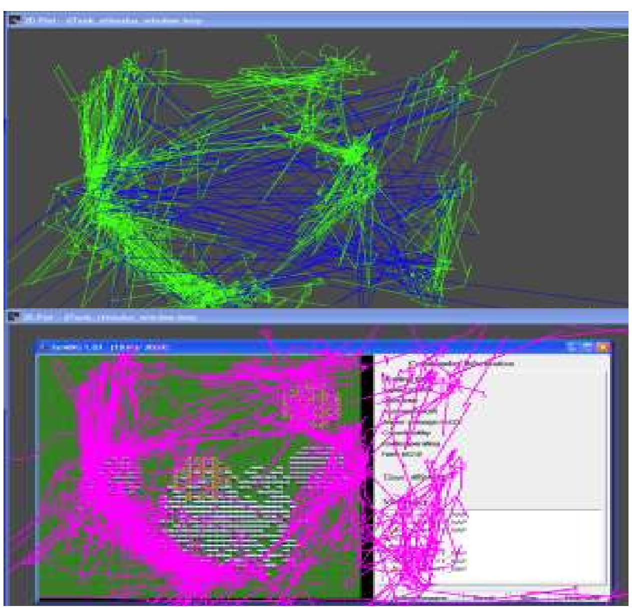
Figure 3: Map annotated by the SME

The SME verbally indicated that any areas he had indicated as 'high potential threat' (YELLOW) would also be 'strategically weak areas of engagement during battle'

(RED). However, not all RED (strategically weak areas) are also 'high potential threat' (YELLOW) areas. The SME reported that in the absence of a specific battle (including tank position) configuration, the areas of 'low confidence regarding opponent tank location' (MAGENTA) would be redundant with areas of 'high potential threat'. Additional information would be necessary to disambiguate these two categories and to specify the areas of 'high confidence regarding opponent tank location' (BLACK). Figure 3 shows where the SME annotated a map and verbally reported that he was being “extremely careful” with the placement of the green, 'strategically strong areas of engagement during battle' to hug the periphery of the low hills area. This is in marked contrast to the placement of the 'strategically strong' areas around the towns. We investigated this information in the eye-tracking analysis of the SME battle behavior.

5.3 Eye-tracking dTank Trials

We had the SME perform a series of dTank battle trials during which we recorded his eye movements, using terrain maps drawn from the map annotation activity described above. The omniscient view terrain map and the SME's eye movements during one battle appear in Figure 4a below. In this display, saccades (eye movements that exceed a threshold velocity of .2) appear in blue and fixations appear in green. Figure 4b displays a superposition of fixations (in pink; the saccade traces removed for ease of interpretation) on the omniscient version of the terrain map. Note that the SME never sees this omniscient view of the terrain map during the battle.



Figures 4 a and b: Example SME eye trace. 4a displays both the saccades and the fixations; 4b displays only the fixations and is superimposed upon the omniscient map

The SME spends a small amount of time looking at the battle statistics (i.e., heading, speed, time, etc.), while the vast majority of the SME's gaze is directed at the location of the opponents and at the terrain immediately in the vicinity of the SME's tank. The fact that the SME spends much of his time regarding the immediate vicinity is not surprising: that is the only visual information about the terrain that is present to the human commander (see Figure 1). However, it is very interesting that the SME spends so much of his time gazing at the location of the opponents, even if the opponent is not currently visible.

To quantitatively report how much time was spent looking at the different regions of the terrain map, we generated a 'weather map' version of the eye tracking results. In this display, the terrain map has been segmented into a 10x10 grid, and the total fixation time for each of the 100 terrain areas has been color coded. The total fixation times associated with each color is given in Table 2 below ('warmer' colors represent a disproportionate amount of time spent focusing on that map region), while the corresponding frequency map is shown in Figure 5.

Color	Total Fixation (ms)
Red	800 <
Orange	200 – 799
Yellow	100 – 199
Green	25 – 99
Aqua	10 – 24
Blue	1 – 9
Gray	< 1

Table 2: Color coding of total fixation times as displayed in the eye tracking 'weather maps'

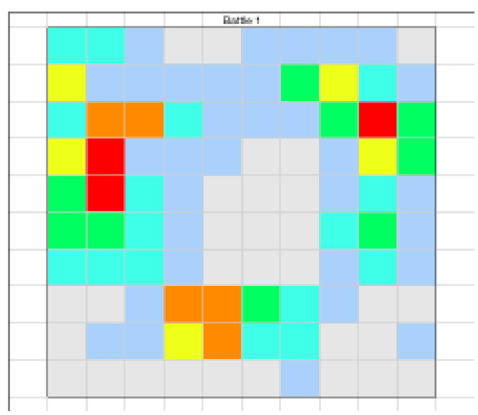


Figure 5: The total fixation time in the 100 regions of the Battle 1 terrain map.

The comparison of the different visual perceptions of the terrain map by the SME reveals that the SME's eye movement during the battle is strongly influenced by the location of the opponents during the battle. This is not surprising – the goal of the battle is not to scout the terrain, but to seek and engage opponents. This analysis further reveals that the SME spends a great deal of time looking at the opponent – sometimes even when there is nothing to see. At this stage of the battle, the SME has destroyed 2 of the 3 opponents and has located the final opponent tank – it is in the middle of the trees. The vast majority of the time, the SME can see absolutely nothing of the middle of those trees – yet, as displayed in the associated 'weather map', the SME spends an extraordinary amount of time focusing on that region of the map, looking where he thinks the opponent is, even when the opponent is out of view.

The SME's visual fixations are less concerned with studying terrain features, but instead trying to locate and keep track of the opponents. The SME is very aware of the terrain features, and uses them to his advantage during battles, but the knowledge he is maintaining is less spatial and more environmental. This leads to the consideration of what environmental information the SME is detecting, representing and utilizing during the course of a battle, and how to best represent this information in attributes.

5.4 Map Annotation

During the Cognitive Task Analysis (CTA), it became apparent that the SME was not merely paying attention to and making decisions based on the raw environment information he was getting from the game. Rather than keeping track specifically of where woods, buildings, hills, and grass were located, he was maintaining a representation of which areas of the map were safe to be in, which provided a good place to hide and which were dangerous. However, our initial models only used the "ground-truth" or raw environment information that they could parse directly from the environment. Thus, we turn to determining how to interpret and represent the environmental information similar to the SME.

Using a handful of maps annotated by the SME as reference, we automated and approximated the process. As Figure 3 depicts, the entire map has not been annotated by the SME; we defined unannotated areas of grass as "risky" and unannotated areas of terrain as "cover". We also broke up the categories of safe (green) and deep-cover (yellow) into three categories – safe, cover, and deep-cover. The resulting five categories give the agent a comprehensive but abstract representation of the entire dTank environment. Grass is coded based on proximity to terrain as safe (within 2 tiles), risky (2 to 6 tiles away), or dangerous (more than 6 tiles away). Similarly, terrain was classified based on proximity to grass as safe (within 1 tile), cover (within 2 tiles), or deep-cover (more than 2 tiles away).

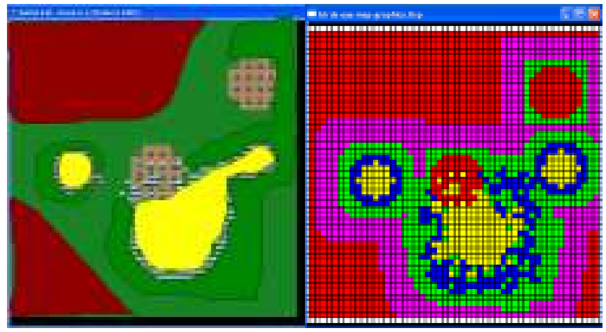


Figure 6: The map from Figure 3, as annotated by the SME (left) and the same map, annotated automatically (right). Red (dangerous), green (safe), and yellow (cover with poor visibility) represent the same type of terrain in both maps. Blue designates terrain that is considered in cover while pink designates risky terrain.

Figure 6 shows a comparison of the map annotated by the SME and the map annotated by our automated system. As discussed above, there are more categories of terrain in the automated version; however, the two maps are qualitatively very similar. Both show the same "dangerous" and "deep-cover" areas, even preserving the slightly fuzzy boundaries between territory considered safe and cover. The main difference to note is the red circle in the upper-right corner of the map that was annotated automatically. The entire town has been colored in red based on the SME comment that he tried to stay away from towns because they offered a false cover. We designate towns as "dangerous" to convey this message to the agent.

5.5 Egocentric Spatial Representation

As part of our attribute enumeration, we identified a significant gap in the information conveyed in the attributes as directly extracted from the environment. While the human SME was able to identify objects on the map and navigate around them, the environment attributes did not convey the spatial reasoning that humans take for granted (i.e., the disconnected nature of the attributes prevented any sort of navigation with respect to obstacles). To remedy this, we developed an egocentric spatial representation, dividing the area surrounding the tank into bins of varying width, and adding several attributes that provided information as to the identity of and distance to the objects surrounding the tank. We chose to employ finer granularity towards the front of the tank and rougher granularity towards the back in an effort to capture the fact that humans generally encode a greater level of detail about the environment in front of them and pay less attention to what is behind them (see Figure 7).

The distance to the closest terrain object (woods, hill, wall, etc.) in each bin is returned as an attribute. We also identify the closest and farthest of the objects and provide the angles to those objects as attributes. The addition of these navigational attributes allowed us to successfully create a model that drives a tank through a curvy path surrounded by trees – even if the path is different than any path used in the data collection (see below). We have also implemented the same type of binning using the cognitive attributes discussed in the next section.

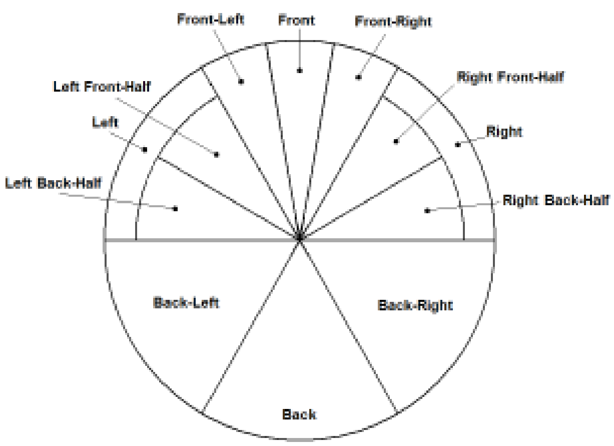


Figure 7: spatial bins used with dTank for determining the closest-object attributes

6. Validation in a Virtual Environmental

We initially used ACT-R for the instance-based model, but due to difficulties in achieving satisfactory real-time performance (response times within 200ms), we tested the representations above using our in-house instance-based rule engine and toolkit, CIBRE (Best & Gerhart, 2011) for both the entire battle and several sub-task paradigms.

6.1 Using Cues to Predict Goals

Given the cues and goals listed above, we evaluated whether the cues comprised a sufficient set of predictors for the goals (assuming that they were, in fact, predictable). Table 4 shows the C4.5 derived classifier (Quinlan, 1993) was able to correctly identify both primary goals and combination goals an overwhelming majority of the time, compared to the probability of correctly guessing (Table 3). Note that predicting the specific action to take was less successful; we were unable to replicate human performance using the entire battle paradigm.

	All Games	Testing Set	Game 1	Game 2	Game 3	Game 4
Action Taken	14.92%	15.23%	14.50%	14.57%	15.09%	15.08%
Primary Goal	19.82%	19.44%	19.93%	19.79%	22.93%	24.48%
Concat Goals	7.77%	7.62%	7.25%	7.52%	7.60%	11.38%

Table 3: The probability of correctly guessing.

	All Games	Testing Set	Game 1	Game 2	Game 3	Game 4
Action Taken	76.22%	62.43%	26.20%	25.87%	17.94%	18.80%
Primary Goal	95.41%	95.65%	35.56%	34.06%	36.26%	11.81%
Concat Goals	93.61%	93.41%	14.85%	9.65%	20.94%	5.65%

Table 4: Response accuracy using 1500 random instances, using the CIBRE model

6.2 Using Instances: the Path Driving Model

We developed a model of path driving using the CIBRE instance-based rule engine (Best & Gerhart, 2011) to simulate SME behavior. The SME completed the path five times, avoiding obstacles (trees) along the way, and this data was used to train the CIBRE agent. We compared each recorded position of the CIBRE agent against the closest recorded SME position from among the five paths. A plot of both of these positions overlayed on the map is presented in Figure 8. CIBRE's average deviation from the closest SME path was 1.95 meters with a standard deviation of 1.81 meters and a maximum distance of 13.95 meters. Given that the path is generally about 20 meters wide, expanding to 60 meters wide in some places, this represents close agreement between the training data and CIBRE's behavior.

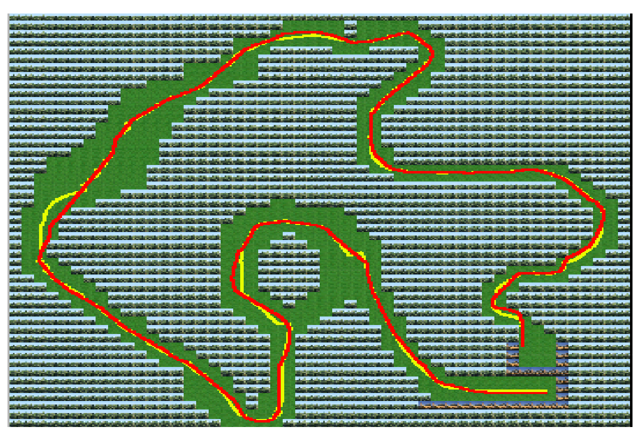


Figure 8: CIBRE (red) vs SME (yellow) path tracings in dTank

Using the methodology described by Best and Lebiere (2006), and Best, Gerhart, and Lebiere (2010), we then transferred the dTank model to a second virtual environment, UT2004, to validate behavior in the new environment. We also collected instances of our SME driving the path in the new environment and trained a new cognitive agent. The results from both models are shown in Figure 9. The image on the left shows the path as driven by CIBRE in UT2004 trained on data collected in dTank (in red). The image on the right shows the path as driven by CIBRE trained on data collected in UT2004 (in blue). The SME performance in UT2004 is on both panels (in yellow). The same analysis described above was performed on both models. The differences between the ported dTank model and the closest SME path in UT2004 (mean 6.73, standard deviation 6.56, maximum 29.93) were similar to the differences between the UT2004 model and the closest SME path in UT2004 (mean 7.20, standard deviation 6.79, maximum 30.68). While the differences are larger than those reported for the dTank model in dTank, they are still within the confines of the path, indicating some success in porting cognitive models between virtual environments using abstract egocentric spatial representations.

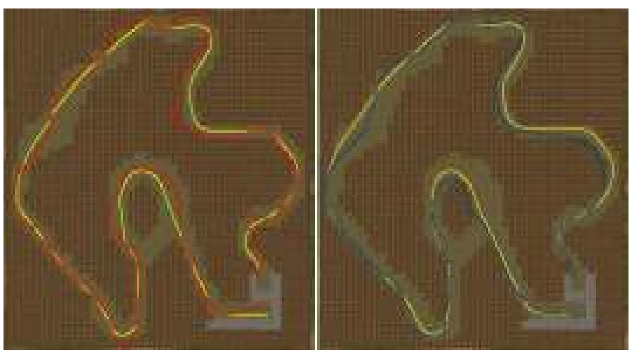


Figure 9: Left: CIBRE model ported from dTank (red) vs. closest SME (yellow) path. Right: CIBRE model trained in UT2004 (blue) vs. the closest SME (yellow).

Figure 10: A CIBRE model following the tree-lined path

Our analysis has shown that while the cognitive model does not follow the SME's behavior exactly, it does a very good job of approximating it, even when ported to a new environment, as long as the same information can be retrieved from both environments.

7. Conclusions

The analysis of the SME behavior told an interesting story: contrary to our original hypothesis, the SME's visual representation of the virtual environment was deceptively simple; sophisticated, but simple. This result is not surprising: experts in a given field are often distinguished from novices in that they are able to recognize and interact with larger groupings – or “chunks” – of information than novices are able to do.

Careful analysis of the SME's discussion of his battle processes revealed that the SME was taking discrete dTank information, such as the current time and the presence of trees, and turning them into meaningful battle cues, such as ' opponent will be loaded soon' and 'the Opponent can see – and possibly fire – at me'. On the basis of this insight, we developed a series of environmental cues or attributes that are computationally derived directly from the discrete dTank information that is available to the SME during a battle. We extracted a total of 113 attributes, and the simple path model uses approximately 10,000 instances. We were unable to generate real-time (within 200ms) predictions using ACT-R, so we used CIBRE for the performance in the virtual environment. However we have found that execution of many of the tactics requires only a subset of these attributes. For example, seeking cover requires knowledge of the terrain with respect to the tank whereas firing at an enemy tank requires knowledge of that enemy's location.

The Cognitive Task Analysis provided a great deal of insight into what the SME is attending to within the environment and how the SME uses that information to perform the overall task of playing dTank. Specifically, this activity led to the production of both Attributes and Tactical Categories that were specific to the dTank domain. These Attributes and Tactical Categories comprise the features and responses that collectively make up the instances upon which the instance-based learning system relies.

One issue with the data collected in the entire battle phase was that there was a large portion of time in which the appropriate behavior was to do nothing. These “empty” sequences of behavior were difficult to reproduce, and caused an overestimation of how much behavioral data we had actually collected. For these reasons, we had the SME perform a series of pre-determined "vignettes" consisting of high-level actions identified during the CTA.

Approximately one third of the data we collected were pre-segmented behavioral sequences during which the SME performed a simpler task several times (e.g., following an opponent, aiming and shooting, following the tree-lined path, etc.). These data streams were much more compact, and were easier to replicate in real-time. Most likely, the increased repetitions of behavioral sequences actually resulted in more useable data per model, despite having many times fewer instances than the models created from entire simulations. The improved real-time replication may also be explained by the fact that the behavioral streams were in separate models, so there was no interference between goals (i.e., given a particular situation the appropriate action might depend on the goal to be achieved.

In the end, we were able to induce a model of performance that approximated human behavior directly from the data. However, this was only possible through the combination of 'cognitive attributes', such as safe and risky territory, and an egocentric spatial representation – our attempts at leveraging ground truth directly were consistently unsuccessful. Thus, knowledge engineering was replaced, by necessity, with attribute engineering.

8. References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y . (2004). *An integrated theory of the mind*. Psychological Review 111, (4). 1036-1060.

Best, B. J., Dixon, K. R. & Speed, A. (2008). Modeling an Unstructured Driving Domain: A Comparison of Two Cognitive Frameworks. In *Proceedings of the 2008 Conference on Behavioral Representation in Modeling and Simulation*.

Best, B. J. & Lebiere, C. (2006). Cognitive agents interacting in real and virtual worlds. In R. Sun (ed.), Cognition and Multi-Agent Interaction: From *Cognitive Modeling to Social Simulation.* Cambridge University Press; New York, NY, 186-218.

Best, B. J. & Lovett, M. (2006). Inducing a cognitive model from examples provided by an optimal algorithm. In *Proceedings of the Seventh International Conference on Cognitive Modeling* (pp. 56-61). Trieste, Italy.

Bellman, R. E. (1957). Dynamic programming. Princeton University Press.

Deng, K., & Moore, A. W (1995.) Multiresolution Instance based Learning. In *Proceedings of IJCAI*, 1995. Morgan Kaufmann

Douglass, S., Ball, J., & Rodgers, S. (2009). Large declarative memories in ACT-R. In *Proceedings of the 9th International Conference of Cognitive Modeling* (paper 234), Manchester, United Kingdom.

Gonzalez, C., Lerch, F. J., & Lebiere, C. (2003). Instance-based learning in real-time dynamic decision making. *Cognitive Science* 27 (4), 591-635.

Ritter, F.E. (2008). dTank: A lightweight synthetic environment for teaching and theoretical research in the Human Behavioral Synthetic Research Environments (HB-SRE) Symposium. In *Proceedings of the 17th Conference on Behavior Representation in Modeling and Simulation*, 196-199. 08-BRIMS-034. U. of Central Florida: Orlando, FL.

Simon, H. A., & Gobet, F. (1996). Templates in chess memory: A mechanism for recalling several boards. *Cognitive Science*, 31, 1–40.

Simon, H. A., & Zhu, X. (1988). Learning mathematics from examples and by doing. In *Models of Thought* (Vol. II, pp. 167–184). New Haven, London: Yale University Press.

Taatgen, N. A. & Wallach, D. (2002). Whether skill acquisition is rule or instance based is determined by the structure of the task. *Cognitive Science Quarterly*, 2(2), 163-204.

Author Biographies

BRAD BEST is a Principal Scientist at Adaptive Cognitive Systems, LLC, in Boulder, CO., where he focuses on cognitive modeling of adaptive behavior in complex environments, especially those that have significant spatial and temporal aspects. His current research interests include integrating perception with decision making in robotic and virtual agents and the development of methods for analyzing, understanding and visualizing model behavior in these environments.

NATHAN GERHART is a Research Programmer at Adaptive Cognitive Systems, LLC, in Boulder, CO. He is the resident subject matter expert for performance in virtual environments and is currently writing software to assist with the exploration and optimization of parametrized models. His current research interests include integrating haptic control with virtual environments and using mobile platforms to enhance endurance training.