A Mental Simulation-Based Decision-Making Architecture Applied to Ground Combat

Dietmar Kunde, PhD
LTC, German Army
Federal Office of the Bundeswehr for Information Management and Information Technology
Koblenz, Germany
49 - 261- 896 8257
DietmarKunde@Bundeswehr.org

Christian J. Darken, PhD
US Naval Postgraduate School
Monterey, CA
831- 656-2095
cjdarken at nps dot edu

Keywords:
Mental Simulation, Naturalistic Decision Making, Event Prediction, Combat Simulation, Combat XXI.

ABSTRACT*: At last year's BRIMS conference, we described a model of mental simulation based on statistical event prediction (Kunde and Darken, 2005). In this paper, we describe a new decision-making architecture based on our mental simulation model. We have developed and tested the model using a scenario built in COMBAT XXI, where the model is used to make fire/hold fire decisions. While the choice of what is to be predicted and the basis for the prediction are chosen by a human modeler, the details of the predictive models are constructed by machine learning based on actual simulation data. Three different predictive models are used to support the decision, one for target richness, one for the effects of obscuring terrain, and one for losses. The outputs of the predictions are integrated by a decision component, which is currently implemented by a decision tree. Preliminary experimental results indicate that the predictive ability of the model and the resulting firing behavior are similar to human performance.*

1. Introduction

For many years now, models of naturalistic decision making have inspired insightful analyses of human and human/machine systems, leading to improvements in human interface and overall system design (Klein et. al. 2003, Miller, 2005, NDM, 2005). The most well-known naturalistic decision making theory is arguably Gary Klein's Recognition-Primed Decision (RPD) model (Klein, 1999). We are very interested in models like RPD and what they imply for how realistic human behavior representations for simulations could function. In particular, our research has focused on a salient component of the RPD model, mental simulation. In brief, mental simulation is the human ability to anticipate the consequences of courses of action in order to select the best one. We have previously advanced a model of mental simulation based on statistical event prediction (Kunde and Darken, 2005). In this paper, we describe how this mental simulation model can be incorporated into a complete architecture for decision making.

We have developed a decision-making architecture as a framework for applying mental simulation in a combat simulation environment. This approach, based on statistical models, shows that simulated entities that are capable of "looking ahead" into the near future perform more realistically than those that do not include knowledge of the past, but only use information from the present (Kunde, 2005). The look ahead consists of the prediction of likely next events over various time scales. Our implementation of a mental simulation component projects the past into the future using no more than three variables like people usually do (Klein, 1999). In the case of the example application considered here, the predicted events include changes in the target richness of the environment and the impact of the terrain in the near future. Losses for friendly and opposing forces over a somewhat longer period of time are also predicted. We consider the resulting behavior "more realistic" because the entities reason and have expectations about a larger number of relevant factors, are able to adjust to sudden changes in the environment (e.g. react to an enemy that is currently not visible), and are able to use information or knowledge gained during a considered period, that is, a simulation run. Knowledge gain, also called learning, improves the overall performance of the software agent.

In the next sections of the paper we put our work in the context of other ongoing efforts and describe the developed architecture for our mental simulation model. In the subsequent sections we show in detail how we approach the empirical terrain evaluation and how the model actually renders the decisions. In closing we compare the behavior of the model to that of human subjects in a preliminary experiment.

2. Related Work

To date, researchers have made the following attempts to model recognition-primed decision-making. The work closest to this project was done by Sokolowski (2002). He implemented a model for the recognition-primed decision-making of a Joint Task Force commander in an operational military scenario using a multi agent system approach. With this computational system, Sokolowski could mimic the cognitive process. However, he didn't focus on mental simulation and stated specifically that "the mental-simulation process will most likely need to be enhanced to better replicate the role of mental simulation within RPD" (Sokolowski, 2002). Warwick et al., (2001) approached their modeling of RPD by encoding the long-term memory (LTM) of decision makers. They modeled LTM in a data structure by storing individual decision-making experiences as a two-dimensional array. When new situations occur, they are compared with experiences stored in the LTM. Computing a "similarity value" yields a measure of comparability in order to recognize a usable experience and the appropriate course of action. Although it seems to show promise as a model of parts of RPD, the mental simulation part has yet to be designed (Warwick, 2002).

3. The Architecture

The general framework of the model developed is depicted in Figure 1. The entire system consists of four components: the environment, which covers mainly the simulation system, the situational awareness component, which ensures an up-to-date situational picture at the decision time, the mental simulator, which predicts and assesses, and the decision component, which evaluates the influencing factors and actually renders the decision.

3.1 Example Scenario

In order to conduct experiments and to present results we designed an example scenario with two modifications which is a further development of the scenario used last year (Kunde and Darken, 2005).
The forces depicted in the simulation were a blue and a red tank platoon. The red tanks would always start out following a predetermined path through blue's kill zone, but would attempt to flank blue's position if blue revealed himself too early. Only blue used our decision-making architecture. Red's behavior was generated in a relatively simple and conventional manner.

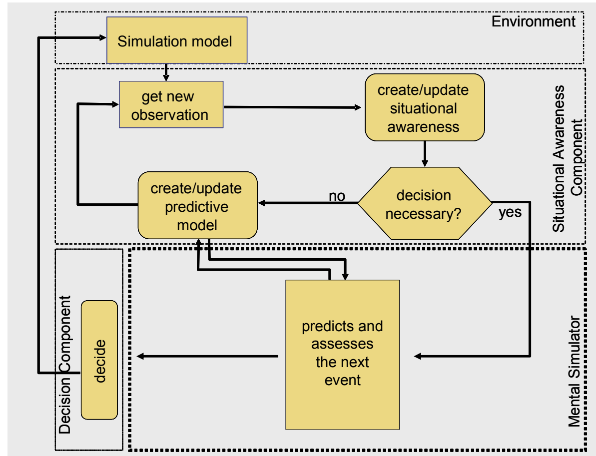

Figure 1. The general architecture.

3.2 Simulation Environment

The simulation environment is the driving component. It contains the simulation system that can run on the same computer or can be networked. For the general use it does not matter, as long as the output of the system contains the required data for the other components. That sounds totally obvious, however this is not always the case and the simulation system might need to be adjusted. One occurrence of that case could be related to the way detections are handled in a combat simulation system. The target acquisition algorithms yield detections of entities, but in contrast to a human observer on the battlefield they normally do not provide the event when a spotted unit goes out of sight. This can only be deduced when in the next observation-sweep the specific entity does not show up on the "detection list" any more. But then it is still unknown at what specific time and at what specific location this occurred. Another consideration could be the case in which aggregated units are used. The attrition of aggregated units is normally computed by Lanchester Equations. The target detection and acquisition does not provide information about individual tanks. There exist combat simulation models where the resolution is not on the entity level, like in Vector in Commander (VIC, 2005). That does not exclude aggregated models from being used in this research. However, the decision-making process will not be more detailed than the model's resolution level.

3.3 Situational Awareness Component

The situational awareness component takes the output of the simulation and builds up its own internal perception of the world (Sutton and Barto, 1981). For our ground combat scenario, it creates estimates about the enemy formations, speed and directions. The situation awareness component exists to provide the mental simulator with a description of the current situation (Kunde and Darken, 2005). If the mental simulator is actively learning its predictive model, the percepts/observations necessary for training the model are also sent to the mental simulator. If there is a finished predictive model pre-loaded, only an update of the current situation would occur. In the abstract view of Figure 1, the situational awareness component is not limited to ground combat situations alone. It is applicable to all cases of simulation where a more sophisticated awareness is required than is directly available from the simulation system. This might include appropriate knowledge in a 3D-environment about the value, benefit, or meaning that "people" seen in the virtual environment have due to their spatial relationships. This might mean to know that I can watch a certain portion of a building and others see a different portion, but overall I know what portion of the building in total can be surveyed.

3.4 Mental Simulator

The mental simulator, the central component of the architecture, makes the difference between our simulation system and all other combat simulation systems. It uses the knowledge gained in the past, predicts the next probable event, puts this estimated event into the context of the anticipated situation and has knowledge about potential outcomes in terms of blue and red losses.

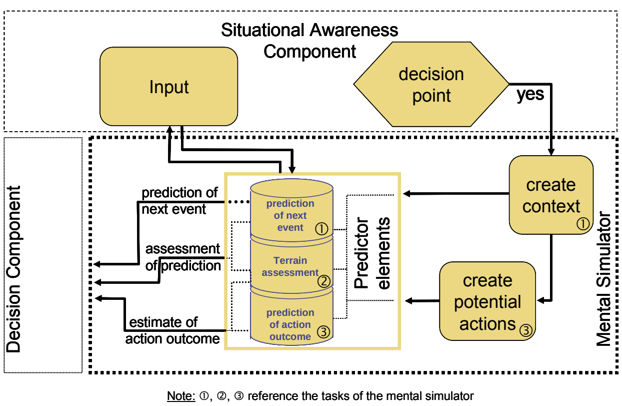

Figure 2. Detail of the mental simulator as applied to our test scenario.

A detailed view of the mental simulator as applied to the fire/hold fire decision is depicted in Figure 2. The circled numbers in the figure depict the three predictors contained in this component:

1. To retrieve a context from the situational awareness component, and to estimate the next probable observation and the average (typically we use the median) time when this event will occur (Kunde, 2005);
2. To predict the terrain quality in the near future; and
3. To create potential actions and estimate their outcomes.

Each predictor has a model structure chosen by a human modeler with internal parameters that are populated in a machine learning process based on simulation data. The learning process represents the prior experience of the blue commander. While in principle additional learning could occur concurrently with the active use (record runs) of the model, for experimental convenience learning was done off-line on special simulation runs designed to maximize learning.

Predictor 1

The context binds the variables of the maximum number of tanks per formation and determines whether, for the upcoming decisions, several formations are currently observed. In the case of an observation of tanks from multiple formations, the one that is the greatest threat is selected to engage first. In the current implementation, the threat is proportional to the distance, which is part of the provided context. There is also a need to assess whether the red tanks can detect the blue tanks. This assessment can't be done due to model deficiencies in the test bed.. The decision component later on decides on the threat evaluation done in the mental simulator. The predictive model we used for this scenario is a Markov Chain. In order to ensure that events with a lower probability will also sometimes be predicted we used a Monte Carlo simulation for sampling the values from the probability distribution as estimates (Kunde and Darken, 2005).

Predictor 2

In the test bed, Combat XXI, we used detections that were determined with the ACQUIRE algorithm (see next section). We model the likelihood that a tank will go out of sight in the near future by examining the terrain empirically in a preprocess. This extends the terrain consideration beyond merely having a line-of-sight feature. The ACQUIRE algorithm uses various parameters to determine whether a specific sensor detects a specific target. In extension to that our model, the terrain assessment given the presence of a line-of-sight, enables entities to estimate how likely it will be to detect a target in a certain terrain before the target actually arrives. To a certain degree, that ability to predict likelihood, or probability, mimics the anticipation of "undetection." This capability is important when modeling, for example, the behavior of human tank gunners in a "duel" situation, in which they monitor targets before shooting them. In known constructive combat simulation environments to date, it is not implemented to consider how long the target might be visible, since the observations occur in a manner similar to a radar sweep of a certain sector (see 4.1). But with the terrain assessment performed in our model, an agent can anticipate when targets will go out of sight and take appropriate action, rather than the agent simply recognizing eventually that the targets are gone.

Predictor 3

The mental simulator is also tasked to create potential actions. The current implementation is coded to create two potential actions: 1) to initiate the firing process as soon as a target becomes visible on the battlefield and 2) not to start engagement, respectively to hold fire until the next observation occurs.

In case 1, the risk of outflanking arises and the likelihood the likelihood of not seeing all tanks increases. In case 2, when all or most of the red tanks are visible, the likelihood that they will try to outflank the blue tanks is relatively small. In a real-world situation, they would most probably move in ways to avoid cross-movements relative to the enemy, and try to engage as quickly as possible. In Combat XXI, these two actions were simulated with the Run Manager, and the output was determined with respect to blue losses, red losses, and the "starting state," which is the number of red tanks seen in the first observation.

3.5 Decision Component

In the current implementation, the decision component requires three inputs:

- a prediction of the next event likely to occur (predicted change in target richness),
- an assessment of the prediction with respect to expected terrain influence, and
- an assessment of the likely blue and red losses given each possible action.

In other words, the decision component takes the predicted number of tanks to see in the next observation, retrieves a median time for this event to occur, and estimates the expected location of the tanks to be seen. The new location estimate is calculated geometrically based on the estimated speed and direction. For this estimated location there is a terrain cell attribute that indicates how likely it is that an observation will occur in this location. The terrain cell attribute is defined in terms of the number of detections in a preliminary run. The terrain attribute, explained in 4.2, will be used to assess the prediction. The probable blue and red losses for each possible action are retrieved from the database. In preliminary runs in similar scenarios, the dependence of losses on whether the platoon fired immediately or delayed the firing and also on the number of tanks seen in the initial observation was determined.

In a real combat environment, a commander observing a tank can continue to look at the tank as long as the same line of sight also continues. Seeing the movement of the tank or anticipating the path in the future, he can determine that the tank might go out of sight in a certain amount of time given obstacles, terrain features, etc. He can determine a certain amount of wait time in which he has to fire before he "loses" the target. Contrary to that is the situation in a constructive simulation environment. There are events at a particular point in time that determine that certain detections have been made. But, in general, there is not information available as to how long the observed entities will be visible. Although in some cases the system developers accepted that there is a need also for undetection information, no such implementation has yet been accomplished. Therefore, we developed a method to get information as to when a tank would probably go out of sight. This can be seen as an upper bound on how long to wait until the enemy tank is engaged in order to enrich the environment with targets and to avoid that the currently undetected members of the formation change their path.

4 Empirical Terrain Assessment

A novel aspect of our model is that our decision making model is provided with an empirical assessment of the impact of terrain on the continued visibility of the targets. By "empirical" we mean that the terrain judgments are not based directly on the ground truth about the terrain, but are instead based on experience.

4.1 ACQUIRE Algorithm

The U.S. Army's current standard algorithm for target acquisition is the ACQUIRE model. The ACQUIRE algorithm is a common search-and-target-acquisition algorithm used in many army force-on-force models (Cioppa et al., 2003). The ACQUIRE algorithm predicts target acquisition performance for imaging systems that operate in the visible, near-infrared, and infrared spectral bands. Therefore, it covers all sensors that occur in our currently implemented scenarios. According to the user's guide, the ACQUIRE algorithm

> predicts the expected proportion of an ensemble of trained military observers who can discriminate a target of a given size and temperature difference with the background, under specified atmospheric conditions (ACQUIRE Range Performance Model for Target Acquisition Systems, 1995).

The ACQUIRE algorithm uses a Field of View (FoV) and a Field of Regard (FoR) nomenclature. Field of View is the horizontal and vertical angle that the sensor looks at, plus a scaling factor that is not of further interest to our simulation. The ACQUIRE algorithm is applied independently for each FoV. Before a FoV can be revisited, the entire Field of Regard must have been scanned: thus, the bigger the number of FoVs per FoR, the longer it is before any one FoV can be revisited.

4.2 Terrain Attributes

We introduced the term terrain attributes. A terrain attribute is an index that determines whether a particular terrain cell can be categorized as having either a "good" or a "bad" rate of detectability. Our terrain of interest, that is, the site where we expect decisions to occur, is divided into 100 x 100 m cells. In each cell, approximately four to six tanks were randomly distributed. No entity (i.e., tank) was moving, but the target acquisition algorithm was made active. Then the simulation is turned on and the detections, which occur over time, are recorded. The graph in Figure 3 shows how the detections in a particular repetition occurred over time. It can be seen that the longer the scan time the more tanks get detected. To determine a reasonable cell attribute, we conducted 50 runs of the combat simulation model with a scan time that was similar to the actual simulation run. In our scenario we had scan times per FoV that were normal distributed over a mean of 3.5 seconds.

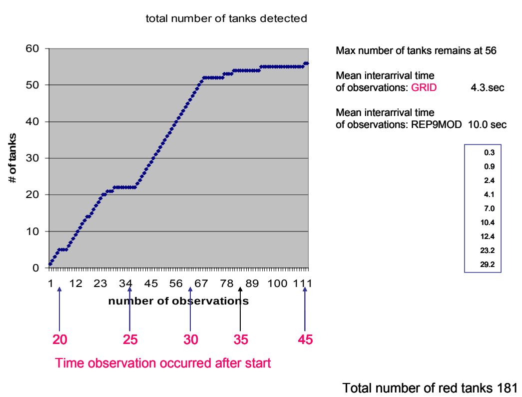

Figure 3. Total detections over time.

Figure 4 depicts one example of the terrain assessment: the cell with the coordinates (59200, 23100), which contains six tanks. Their numbers are listed at the right. The ACQUIRE algorithm detected only one tank. A cell was attributed as "good," in terms of its detectability, when more than 50 percent of its tanks were detected. In this case, the cell was attributed as "bad."

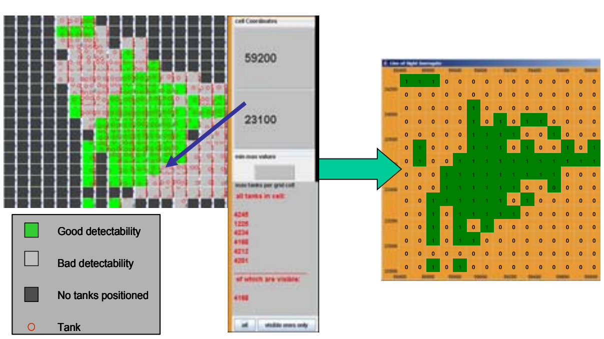

Figure 4. The determination of terrain attributes

Since the detection is stochastic, the attributes for the terrain cells are aggregated. Figure 5 shows the variation among various runs.

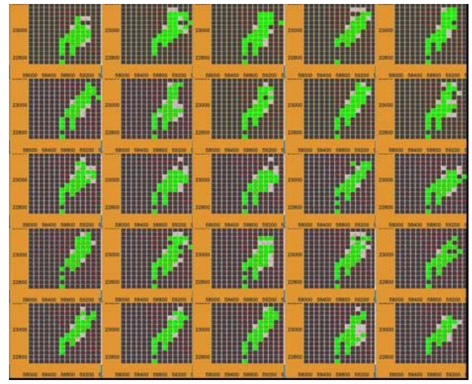

Figure 5. Variation in the number of tanks detected over 25 runs.

A terrain attribute also depends on the location of both the observer and the target. In our example, none of the tanks, including the observer's is moving over the course of a single run. Therefore, we also conducted several runs in which the target tanks randomly changed position within their 100 x 100 m cells. The number of tanks per cell, however, was kept constant. The mean of the six runs conducted was forty-seven detected tanks, plus or minus seven tanks, in a 1.4 x 1.4 km square. The number of cells containing tanks and having line of sight to the observer tanks is 121. Thus, the variation per cell is in average less than half a tank. A terrain cell earned the attribute "good," indicated by green in Figure 4 , when in 90% of the cases half or more of the tanks were detected. In all other cases the terrain cells were attributed as "bad." When there was no detection at all, the cell was colored dark gray; in the remaining cases, light gray.

We also evaluated how well this approach performs. In order to do so, we looked at 10 replications of a single scenario involving a group of moving hostile tanks. We examined all consecutive observations that had a change of terrain cell attributes associated with them. That means, when an observation i occurred in a "good" terrain cell and the observation i+1 occurred in a "bad" terrain cell then the number of tanks in each observation are recorded. This was done in the same way when the terrain cell attribute change occurred from "bad" to "good." When no change occurred, nothing was recorded. At the end of the scenario replications the mean values for changes from "good" to "bad" and from "bad" to "good" were determined and put into Figure 6. This figure displays the mean values of differences in number of tanks observed vertically. The x-axis displays the replications.

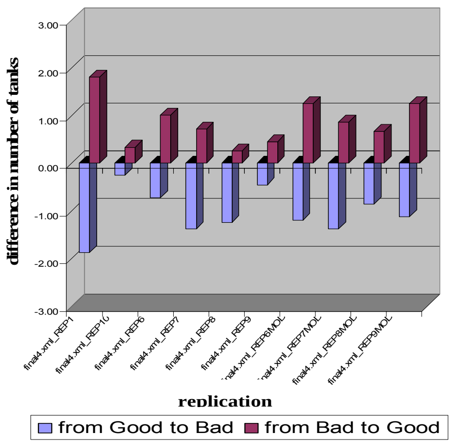

Figure 6. Changes in number of tanks observed around a terrain cell attribute change

Above the zero line are the values for changes from "bad" to "good" and below vice versa. It is apparent that all means are either above or below the zero line indicating that it can be assumed when going, for example, from a "good" to a "bad" terrain cell in average less tanks can be expected in the next observations. We then also truncated the data (due to space constraints not displayed). Truncation was done when the first damage occurred. The resulting chart showed more clearly the difference between good and bad terrain cells because the maximum number of tanks observable decreases after damage occurred.

5 Decision Component

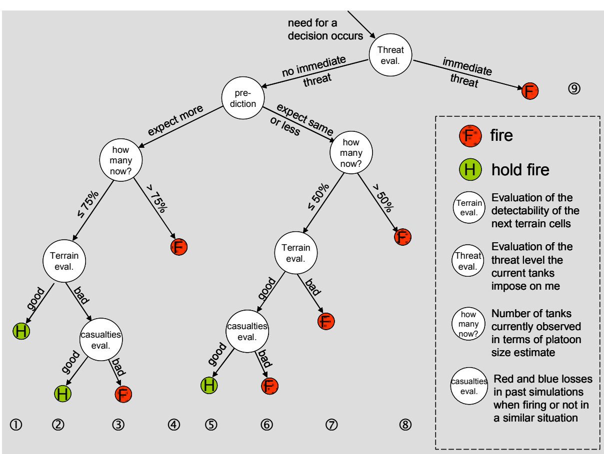

Figure 7. Decision Tree

In our basic initial example of an agent, a tank platoon commander, the tank decides to fire according to the decision tree in Figure 7. Once a tank is in view, this decision tree gets activated because there is a need for a decision. The decision component proceeds downwards through the tree until it hits a node that says "fire" or "hold fire." At each node a condition is checked and based on the outcome of this condition the respective path is chosen.

The top node evaluates the threat level of the tanks observed to the blue (friendly) tank platoon where leader's decision process is being modeled. Determination of threat level is based on range in the current implementation. Other potential factors could include the heading of the tank or whether the enemy gun points towards the blue position. Our handling of threat assumes that the blue tanks are in a turret-down or hull-down position, in which the probability of detection is relatively small. The threat level might also be influenced by the mission, not only by the risk of being shot at. A good example would be a mission of suppression of enemy reconnaissance. Even if the enemy tank does not detect the blue position it can still be a severe threat, because of the capability of reporting reconnaissance results that might endanger blue's own operation. The heading of the enemy tank's gun cannot currently be retrieved in Combat XXI. Therefore, it is not modeled. However, clearly determination of a threat can be based on various parameters. If an immediate threat is assessed, that is the enemy comes within a certain threshold distance, then the tank immediately starts the engagement process to prevent being shot themselves (path 9 in Figure 7). Otherwise a hierarchical approach in the decision tree is further pursued.

In the next two layers of the tree, the prediction of how many tanks will most likely be seen in the next observations is used. If the prediction will be at most the same number of tanks as currently seen and this value exceeds 50% of the estimated platoon size then the engagement process is also initiated. This rule captures the case where currently three or four out of four tanks possible are observed and it is unlikely to see more in the next observation. Therefore, firing at the ones observed would be a reason-able thing to do. For the case that currently only two tanks out of four are observed, the terrain is additionally assessed, and in case of a good detectability of the future terrain cell the casualty evaluation is conducted, otherwise the engagement starts. If the casualty evaluation is promising then fire is on hold, otherwise also the engagement process starts immediately (cases ⑤ - ⑧ in Figure 7).

If the prediction indicates a higher number of tanks than currently observed, then the expected percentage of the estimated current platoon size the tank commander will see is determined. This captures the situation when, for example, a platoon has five tanks and four tanks are seen and the system actually starts firing at them (case ④in Figure 7).

If the number of enemy tanks currently observed is less than the maximum number possible, for example two in our example, then the terrain evaluation triggers the engagement process. If good detectability is anticipated, the model holds fire (case ① in Figure 7). If poor detectability is anticipated, the model assesses the casualties evaluation from preliminary runs. If the casualty evaluation indicates fewer losses when waiting to fire then the model holds fire, otherwise it fires (cases ② and ③ in Figure 7).

These factors enable the simulated platoon commander to make better decisions. The decision tree and the conditions were discussed with officers from the armor branch of several countries represented at the Naval Postgraduate School. In existing models, inappropriate immediate firing remains unpunished because the attacker also behaves inappropriately, ignoring the first shot or even a resulting kill and continuing to follow the scripted path.

The decision component also creates the explanatory component of the system. This means it provides a text string from which the user can see why decisions made by the model turned out the way they did; making the rationale transparent to the user. There are no anonymous numbers that lead to a decision. All numbers used have a meaning in terms of losses, time or probabilities. Therefore, the decisions can be explained in a natural human way.

6. Experiments

We conducted four experiments. They all used the same general Combat XXI scenario. The first experiment addressed the question of whether there would be a difference in prediction accuracy as a function of the number of state machines. This is a more technical experiment and was used to make design decisions. We describe in the following those experiments conducted with military officers from various services at the Naval Postgraduate School. The number of human participants varied between 6 and 11. The next experiment (No. 2) compared the prediction accuracy of the model to that of humans. Human participants observed the simulation screen and were asked to predict how many tanks will be detected in the next observation cycle. They were provided as tools graphical representations of the information used by the program: the Markov Chain, terrain attributes, and estimated outcomes (losses) from potential firing. This and the next experiment address the model's validity by comparing its performance to human performance. The third experiment examined how the tools, provided to the participants and mandatory for the model to work, impact the human predictions. The fourth experiment compared the firing behavior from humans and the model based on experiment 3. In this paper we focus on the prediction and firing behavior.

7. Results

7.1 Prediction Behavior

The prediction behavior was examined in experiment 2 and 3. Figure 8 displays the results from experiment 2. Figure 9 displays the results from experiment 3. The task was similar to experiment 2 with the twist that in the first four replications, no tools were provided. In the second four replications all tools were provided. The participants from experiment 3 are disjoint with those from experiment 2.

		first prediction	all predictions
Model	mean	0.80	0.67
	sdev	0.45	0.23
Human	mean	0.85	0.63
	sdev	0.17	0.09

Figure 8. Results from experiment 2

The analysis of the data collected was done in terms of how often the prediction was correct. This was assessed again only for the first prediction of each replication and for all common number of predictions. The mental simulator of course got the tools in both cases.

	Scenario	only first prediction Human			only first prediction Mental Simulator		
		predicted correctly	predicted wrongly	ratio	predicted correctly	predicted wrongly	ratio
no tools	final4.xml_REP6	2	4	0.33	2	0	1.00
	final4.xml_REP7	3	3	0.50	2	0	1.00
	final4.xml_REP8	4	2	0.67	1	1	0.50
	final4.xml_REP9	5	1	0.83	1	1	0.50
			mean	0.58		mean	0.75
with tools	final4.xml_REP6MOD	4	2	0.67	2	0	1.00
	final4.xml_REP7MOD	4	2	0.67	1	1	0.50
	final4.xml_REP8MOD	5	0	1.00	2	0	1.00
	final4.xml_REP9MOD	5	1	0.83	1	1	0.50
			mean	0.79		mean	0.75
			Sdev above	0.22		Sdev above	0.29
			Sdev below	0.16		Sdev below	0.29

	Scenario	all predictions Human			all first prediction Mental Simulator		
		predicted correctly	predicted wrongly	ratio	predicted correctly	predicted wrongly	ratio
no tools	final4.xml_REP6	9	16	0.36	6	4	0.60
	final4.xml_REP7	3	3	0.50	2	0	1.00
	final4.xml_REP8	4	2	0.67	1	1	0.50
	final4.xml_REP9	5	1	0.83	1	1	0.50
			mean	0.59		mean	0.65
with tools	final4.xml_REP6MOD	10	7	0.59	5	1	0.83
	final4.xml_REP7MOD	7	4	0.64	1	3	0.25
	final4.xml_REP8MOD	6	0	1.00	2	0	1.00
	final4.xml_REP9MOD	5	1	0.83	1	1	0.50
			mean	0.76		mean	0.65
			Sdev above	0.21		Sdev above	0.24
			Sdev below	0.19		Sdev below	0.34

Figure 9. Results from the experiments for comparing prediction accuracy.

7.2 Firing Behavior

This experiment uses the data collected in experiments 2 and 3. There, the participants predicted the next observation and decided to fire when an observation sequence met their individual criteria for a firing decision. In experiments 2 and 3 the model did not make any firing decisions. The participants were not influenced by the mental simulation model's behavior. In experiment 4, the model decided to fire according to a particular path through the decision tree. Figure 10 displays the results from the firing comparison quantitatively for the scenario final4. The x-axis denotes the various replications of the scenario "final4." The left four replications denote the runs without tools for the participants and the right four replications with the tools provided. The y-axis indicates at what observation the human participants fired on average and in addition when the model fired. In the right four replications one can argue that the humans with the tools basically mimic the model's algorithm. However, then the left data points, REP_7 to REP_9, are harder to explain since the tools were not available to the human participants at that time. The first data point, REP_6 is explainable similarly to the prediction experiment. Having no information about transition probabilities and terrain cell attributes makes it hard to decide when to fire. Furthermore, especially Army participants applied their knowledge of a map this scale to their decision making process without considering that this knowledge is not incorporated in the combat simulation system. Except for the first data point, all decisions of the model to fire are within one standard deviation of the human participants' mean displayed as a yellow hyphened line.

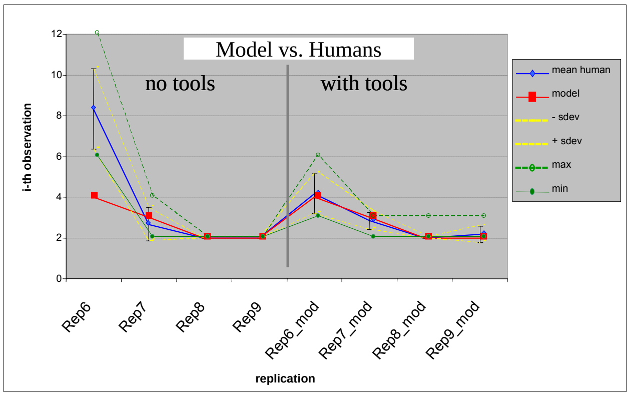

Figure 10. The firing behavior is similar between human participants and the model. Without tools the human participants are off in the beginning but rather quickly adjust their behavior.

The results show that not only the predictions but also the firing decisions perform in the human range. It is obvious that the model never immediately fired the moment a target popped up. Neither did the human participants. For those replications where the humans fired later or early the model decided similarly. Note that the results from the experiment were not used to calibrate the model, and the decision tree was developed independent of the results from the human participants. However, human tank experts were considered prior to the development of the decision tree. We consider this a favorable result for our model.

8. Conclusions

This first approach to the computational modeling of mental simulation is far from being perfect or comprehensive. However, it contributes with a reusable architecture and the implementation we chose shows that mental simulation can be successfully implemented in a combat simulation environment. We hope we have helped pave the way for adding expectations and imagination to better imitate human behavior in a combat simulation.

9. References

ACQUIRE Range Performance Model for Target Acquisition Systems. (1995). Version 1 User's Guide, U.S. Army CECOM Night Vision and Electronic Sensors Director-ate Report, Ft. Belvoir, VA.

Klein, G. (1999). Sources of Power: How People Make Decisions. Cambridge MA, MIT Press.

Klein, G., Ross, K., Moon, B., Klein, D., Hoffman, R., and Hollnagel, E. (2003) „Macrocognition", *IEEE Intelligent Systems*, 18 (3), 81—85.

Kunde, D . (2005). Event prediction for modeling mental simulation in Naturalistic decision making, Dissertation MOVES Institute, Monterey, Naval-Postgraduate School.

Kunde, D. and Darken C. (2005). Event prediction for modeling mental simulation In Naturalistic Decision Making. Proceedings of the 2005 Conference on Behavior Representation in Modeling and Simulation (BRIMS), Universal City CA, May 16-19 2005.

Miller, N.L. and Shattuck, L.G. (2005) Applying a Dynamic Model of Situated Cognition to the Investigation of Mishaps . *Proceedings of the Human Factors and Ergonomics Society's 49 th Annual Meeting.* Orlando , Florida , September 2005.

NDM. (2005). Proceedings of the Seventh International NDM Conference (Ed. J.M.C Schraagen), Amsterdam, The Netherlands, June 2005.

Sokolowski, J.A. (2002). "Can a Composite Agent be Used to Implement a Recognition-Primed Decision Model?," In Proceedings of the Eleventh Conference on Computer Generated Forces and Behavioral Representation, Orlando, FL, May 7-9 2002. pp. 473-478.

Sutton, R. and Barto, A. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. Psychological Review, (88):135-170.

Warwick, W., S. McIlwaine, R. Hutton, and P. McDermott. (2001). Developing Computational Models of Recognition-Primed Decision Making. In Proceedings of the Tenth Conference on Computer Generated Forces, May 15-17, Norfolk, VA, pp 323-331.