Automatic Detection of Discrepancies in After Action Review

Joakim Ekblad, Avelino Gonzalez
University of Central Florida
Orlando, Florida
jox@du.se, gonzalez@ucf.edu

Hans Fernlund
Dalarna University
Borlange, Sweden
hfe@du.se

Paul Barath
Saab Training Systems AB
Huskvarna, Sweden
Paul.Barath@sts.saab.se

ABSTRACT

After-Action Review (AAR) is an effective tool to evaluate and improve the performance of trainees in tactical training exercises. However, when the exercises grow in size, and might reside in several locations, providing feedback to the majority of the participants can be complicated. It requires extensive time and resources, and the review might be limited to the few most important tactical decisions made. This paper presents a model of how to automate the After-Action Review and make it easily accessible to all the participants to increase the efficiency and improve the performance of After-Action Reviews. A system built on expert models where the action of the trainees could be compared with these models can provide additional support for the trainees. However, such a system needs to automatically detect and classify discrepancies. Discrepancies between a trainee and an expert modeled agent can emerge in many situations. By minimizing the discrepancies shown in the AAR to only include the ones believed to be significant enough to decrease the performance of the trainee, the AAR will become more effective by reaching out to the majority of the participants of the exercise giving them individual performance feedback. Preliminary results of our experiments are promising and indicate that the model presented in this paper can be used to address the issues discussed above.

ABOUT THE AUTHORS

Joakim Ekblad received his Bachelor's degree in Information Technology from the Dalarna University, Sweden in 1999. He is currently working on his Master of Science in Computer Engineering at University of Central Florida while he is employed by the Intelligent Systems Lab**.**

Hans Fernlund received his Bachelor's degree in Computer engineering from the Dalarna University, Sweden in 1993. In 1995 he obtained his Master of Science in Computer Engineering from the University of Central Florida. Fernlund obtained his Ph.D. from the University of Central Florida in the spring of 2004. He is an assistant professor at the Dalarna University, Sweden.

Avelino J. Gonzalez received his Bachelor's and Master's degrees in Electrical Engineering from the University of Miami, in 1973 and 1974 respectively. He obtained his Ph.D. from the University of Pittsburgh in 1979 also in Electrical Engineering. He is a Professor in the department of Electrical and Computer Engineering at the University of Central Florida, specializing in Artificial Intelligence and simulation

Paul Barath received his bachelor degree in Forestry from the University of British Columbia, Canada in 1985. In 1997 he received his Master of Business Administration at Uppsala University in Sweden. He has been, for many years, working as a product and business developer for several companies in the sector of defense, space technology, health care and IT technology. He is now the Director of New Product Developments at SAAB Training Systems, Huskvarna, Sweden.

Automatic Detection of Discrepancies in After Action Review

Joakim Ekblad, Avelino Gonzalez
University of Central Florida
Orlando, Florida
jox@du.se, gonzalez@ucf.edu

Hans Fernlund
Dalarna University
Borlange, Sweden
hfe@du.se

Paul Barath
Saab Training Systems AB
Huskvarna, Sweden
Paul.Barath@sts.saab.se

INTRODUCTION

In military training, it is important that the trainee be provided with timely and individual-specific feedback in order to improve his performance in future missions. After-Action Review (AAR) is the process through which this feedback is traditionally provided. AAR is an important tool to evaluate the individual as well as collective task performances for trainees after the training session is completed. The instructor/observer (I/O) who normally provides the feedback must be aware of the actions executed by the trainee, and be able to determine their correctness. It is unrealistic to expect the I/O to continuously monitor every single individual participant in the exercise. This is especially true for large training exercises with many participants. There is an increasing interest in virtual simulations where the participants can be either real or virtual and in different training locations. Conducting constructive AAR in these exercises becomes even more difficult.

To get the most out of AAR, it should be complemented with automated systems that help the I/O generate the appropriate feedback for each individual trainee. To improve the ability of the I/O to provide this feedback, this research seeks to develop intelligent tools to compose a Smart After-Action Review (SmartAAR) technology suite. This approach is based upon the concept of AAR-by-comparison. That is, we seek to build agents that represent appropriate human performance and then use them as benchmarks during execution of the tactical exercise. The trainee's performance is compared continuously and possibly in real time to this benchmark. By pairing each trainee with his own 'personal' expert agent counterpart, individual feedback can be managed to the benefit of the trainee.

Today, there are many support systems for AAR in military exercises. Some of them record the actions and reactions of all actors during an exercise that could be re-played and viewed by the instructors and actors in an AAR session. Extending such a support system for AAR with expert agents can then serve as the basis for AAR by comparison. If the expert agent receives the same inputs as a trainee, the action taken by the agent could be played in the simulated environment of the AAR support system and the discrepancies between the trainee and the expert agent could be identified, marked and logged.

In this paper we present the concept of AAR by comparison, including techniques on how to detect discrepancies, synchronize the agent with the trainee and logging important deviations. The AAR by comparison is designed on a contextual approach, supporting human behavior modeling and situational awareness.

AAR BY COMPARISON

Teaching guidelines and doctrines to military trainees has its drawback in that it is unrealistic to expose all possible scenarios or actions to the trainee. There is often no specific correct action to take for a given situation. More realistic would be to have models of the expertise at hand against which to compare the trainee's action.

The objective of this research is to establish a method whereby simulated expert agents experience the same situations in a simulated environment as the human trainee does in the real world exercise. A comparison between them could then serve as a basis for an individual AAR system. Such a system could be regarded as an evaluation support system. It is not necessary that the system is to be fully automated or provide feedback directly to the trainee. If the system juxtaposes the performance of the expert agent with the environmental data apparent to the agent, it will give the trainee an excellent platform for self-evaluation and learning.

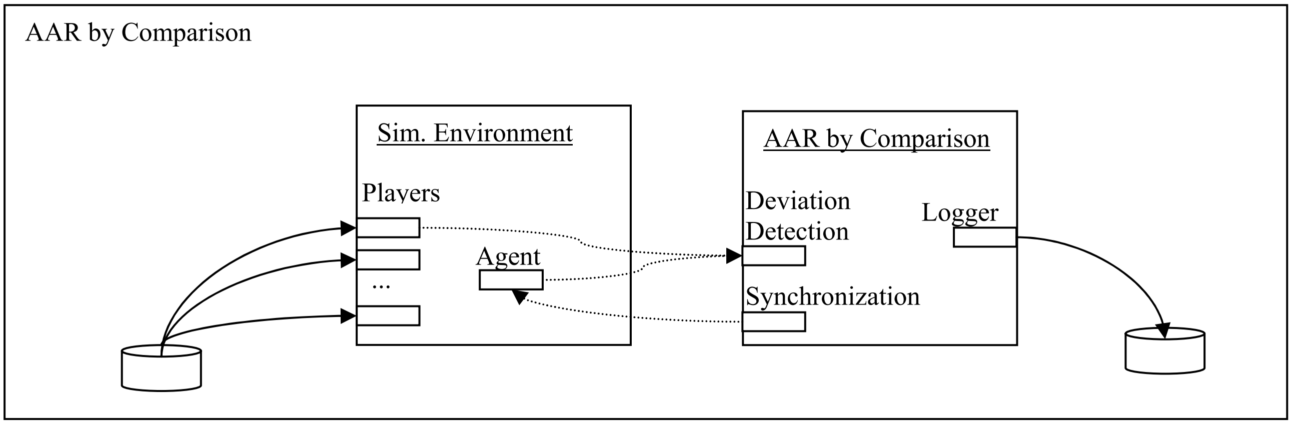

Figure 1: AAR by Comparison

In training exercises, whether live or virtual, there are different types of deviations between the trainee and the expert agent, and with different severity. If the trainee and the agent for some reason chose different paths at a decision point, the deviation might become large. If neither encounters problems on the way, the deviation may be unimportant. Conversely, very small deviations in performance might have severe implications. The two could behave almost the same but one of them might expose itself to the opponents (e.g., be in line-of-sight of an enemy combatant within firing range). Such a small deviation can be the result of two completely different tactics applied to the current situation. It could be the difference between seek cover and attack. This constant comparison between human and agent will permit the continuous evaluation of the trainee's performance in the exercise.

We envision each trainee's performance being continuously compared to the expert agent. As long as the actions of the trainee agree with those of the agent, the trainee is considered to be performing correctly; however, upon observation of a discrepancy from the benchmark expert agent, the discrepancy is noted and logged for evaluation later.

CONTEXTUAL DISCREPANCIES

It is our opinion that people in tactical situations also behave in a context-based fashion. Several researchers in cognitive psychology promote models that are based on context-like structures, most notably Endsley (1995) in her study of situational awareness and Klein (1989) in his recognition-primed decision making approach.

It is our assertion that the most important discrepancies between the expert agent and the trainee occur when they are in different contexts. While discrepancies in time and location may be common throughout an exercise, they may not represent serious problems. A discrepancy in the contexts of the expert agent and a trainee will nearly always be the result of inappropriate actions by the trainee. Hence, in order to facilitate this comparison, the modeling paradigm for the expert agent is context based.

When comparing the agent and the trainee, the expert agent executes in a simulated environment and acts upon the situation that the trainee encounters in the real world. The context model structure needs to be tailored for human behavior representation in simulated agents. For this we employ Context-Based Reasoning (CxBR).

Context-Based Reasoning (CxBR)

Gonzalez and Ahlers (1998) presented CxBR as a modeling paradigm that can efficiently represent the tactical behavior of humans in intelligent simulated agents. Results have shown that it is especially well-suited to modeling such behavior. CxBR is based on the idea that:

- A situation calls for a set of actions and procedures that properly address the current situation.
- As an exercise plays out, a transition to another set of actions and procedures may be periodically required to address a new situation.
- Things likely to happen under the current situation are limited by the current situation itself.

CxBR encapsulates knowledge about appropriate actions and/or procedures for specific situations, as well as compatible new situations, into hierarchically-organized contexts. All the behavioral knowledge is stored in the Context Base (i.e. the collection of all contexts). The top layer of contexts in the hierarchy contains the Mission Context. At the next layer are

Major Contexts and below them, a number of Sub-Contexts layers can exist. Figure 2 shows an example of a context structure from a simple context base that models contextual components of tank platoon behavior. Mission Contexts define the mission to be undertaken by the agent. While it does not control the agent per se, the Mission Context defines the scope of the mission, its goals, the plan, and the constraints imposed (time constraints, weather, etc.).

The Major Context is the primary control element for the agent. It contains functions, rules and a list of compatible Major Contexts that can follow the current one. Identification of a new situation can now be simplified because only a limited number of all situations are possible under the currently active context. Sub-Contexts are abstractions of functions performed by Major Contexts which may be too complex for one function, or that may be employed by other Major Contexts. This encourages re-usability. Sub-Contexts will de-activate themselves upon completion of their actions.

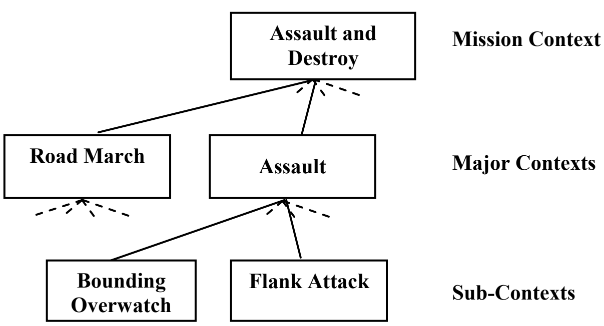

Figure 2: Context-base organization

One and only one Major Context is always active for each agent, making it the sole controller of the agent. When the situation changes, a transition to another Major Context may be required to properly address the emerging situation. For example, a tank platoon may make contact with an inferior force that requires a transition from a Road March to an Assault Major Context. Transitions between contexts are typically triggered by events in the environment – some planned, others unplanned. Events internal to the agent (i.e., mechanical breakdown) can also trigger transitions. Expert performers are able to recognize and identify the transition points quickly and effectively.

CxBR is a very intuitive, efficient and effective representation technique for human behavior. A full description of CxBR can be found in Gonzalez and Ahlers (1998).

DEVIATIONS AND SYNCHRONIZATION

A discrepancy can be of two types (not mutually exclusive): 1) The position, movement, or firing action of the trainee is significantly different from the agent's. 2) The context of the human trainee is different from that of the agent. The first is rather easy to determine by merely overlaying the locations and actions of the trainee onto that of the agent. Given the many possible moves and micro decisions, this type of discrepancy is likely to be a very coarse filter that will result in many logged discrepancies. Many of these discrepancies will turn out to be of little tactical consequence (i.e. unimportant).

The second type of discrepancy is the more significant but more complicated to discover. To make a useful comparison to a context-based model, the AAR system must infer the context in which the trainee is currently operating. Inferring a trainee's intentions and the set of skills being used at the time of the comparison can provide a very useful means of reviewing his performance. The problem, of course, is how to infer the context in which the human is operating. One approach is to use a pattern matching technique that compares the trainee's action with that of the expert agent under various contexts simultaneously. The comparison that results in the closest match will indicate the context in which the trainee is most likely to be operating. This matching of patterns can be said to infer the context and/or sub-context in which the trainee is operating.

After a discrepancy has been detected and logged, the expert agent needs to be synchronized with the trainee. If they are not synchronized, the agent and the trainee might continue their missions on completely divergent paths and further comparison will not be possible. The agent therefore, needs to be forced to regain the same state as the trainee, both when it comes to location, time and status, but also forced to operate in the same context as the trainee. During synchronization, the agent also needs to update its temporal memory. Now all the pieces for the AAR by comparison can be completed, as shown in Figure 1. The recorded data from the trainees (i.e. players) are played in the simulated environment together with the expert agent. The deviation detection unit detects and records (with help of the logger) discrepancies between the agent and the trainee under evaluation. After the discrepancy has been logged, the agent is synchronized with the player again.

DETECTING PHYSICAL DISCREPANCIES

A physical discrepancy (d) stems from the assertion that a non-trivial difference in position, heading or velocity between the agent and the trainee that has been measured and observed. The trainee in the exercise is coupled with an agent in the SmartAAR simulation. The discrepancies between the trainee's and the agent's position, heading, and velocity are retrieved at every simulation cycle. Realistically, there will always be a small deviation in the agent's position, heading and velocity compared to the trainee. Such discrepancies can be discarded as irrelevant and not logged. In our presented model we use a threshold in order to filter this kind of discrepancies.

Agent Synchronization

The basic concept in synchronizing the expert agent with the trainee is to put the expert agent into the same situation as the trainee in order to determine whether the trainee is responding to the situation being currently exposed to in an acceptable manner.

During a mission, there are typically several ways to properly execute it. On the other hand, there are also several incorrect ways to do it. In order for the agent to determine if the trainee is conducting the mission in an acceptable manner, the agent needs to frequently be superimposed on the trainee for a short period of time in order to compare itself with the action of the trainee. If the discrepancies are minor, the performance of the trainee is determined to be acceptable and the agent is again synchronized with the trainee to be able to detect if the trainee's next behavioral pattern is acceptable. If we allow the agent to be totally autonomous and freely act in the environment, the accumulated deviations will, after a period of time, be large enough to trigger a discrepancy of the trainee. This discrepancy, by definition reflects a potentially serious mistake by the trainee; therefore, we believe that the agent must continuously be synchronized with the trainee in a pre-determined time interval. This type of synchronization is referred to as pulse synchronization.

Notably, other trainees in the exercise will not react to any of the expert agent's actions. Nor will the agent's actions change the environment in any way.

Physical Discrepancy Detection Model

The following model is used in the SmartAAR simulation software to detect, classify and report physical discrepancies. The first step in our model is to determine whether a discrepancy truly exists. The discrepancies (d_i) of a time period of length l_D are kept in vector D.

The sum of vector D (D_t) is basically an accumulated value over a time period. A possible discrepancy is detected if D_t is larger than the accumulated threshold τ for the same time period. To determine and classify the level of the discrepancy, a trend at a specific point in time (k_t) is calculated (Eq. 1).

The value of k_t reveals whether the discrepancy is increasing ($k_t > 0$), decreasing ($k_t < 0$) or simply flat lining ($k_t = 0$).

$$k_t = \frac{d_0 - d_l}{l} \tag{1}$$

The variable k_t is then compared to the asymptotic function shown in equation 2. The asymptotic function is designed to emphasize large and probably more meaningful discrepancies rather than smaller ones that are believed to be of lesser importance.

$$\varphi = \frac{5}{e^{0,05 \times D_{avg}}} \tag{2}$$

$$\text{Where } D_{avg} \text{ is the average of vector } D$$

By using a variable k-value (φ) and by comparing it with k_t (Eq. 3) it is possible to classify the trend t_t for the time period. The trend classification t_t for the discrepancy is either decreasing ($t_t = -1$), increasing ($t_t = 1$) or not significantly changing ($t_t = 0$).

$$\begin{aligned} &If\ k_t < 0 \rightarrow t_t = -1 \\ &If\ 0 \le k_t < \varphi \rightarrow t_t = 0 \\ &If\ k_t \ge \varphi \rightarrow t_t = 1 \end{aligned} \tag{3}$$

The asymptotic function in (2) and its implication on the trend classification is presented as a graph in Figure 3. The darker areas in Figure 3 depict that the trend t_t will be classified as an increase ($t_t = 1$), while the brighter areas will be classified as not significantly changing ($t_t = 0$). Noticeably, the lower discrepancies return very high k-values, but as the discrepancies increase, the function will return lower k-values.

To get a longer perspective of the observed discrepancy its trend classification t_t is added to a vector T containing the trends over a time period of length l_T. The sum of the vector T (T_t) is used to classify the discrepancies.

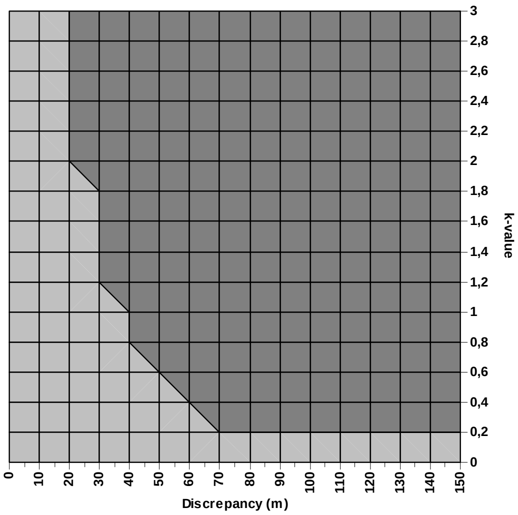

Figure 3. The asymptotic function (3).

Whenever a physical discrepancy is detected by observing D_t being greater than τ, it can be classified. We believe that a discrepancy can be classified into three different levels. These three levels represent the significance of the discrepancy. A second threshold (δ) is introduced to separate less significant from more important discrepancies. This threshold is needed to be able to control the levels depending on the context.

The discrepancy is classified as a first level discrepancy when $0 < T_t \leq \delta$. This means that the discrepancy is of such size that it should be taken into account in the analysis. However, it is still of such small magnitude that it is not displayed to the trainee at this point. The discrepancies at this level are classified as being insignificant, but still have potential to grow larger and eventually become significant. Because of the discrepancies are believed to be insignificant at the time, the pulse synchronization remains enabled.

Second level discrepancies are such that $\delta < T_t < l_T$ since the discrepancy is above the threshold level but is not constantly increasing over the time period. Typical for this type of discrepancy are those that are above the first threshold, but also above the second threshold which separates it from the first level. The magnitude of the discrepancy is believed to provide useful information to the trainee. We believe that these discrepancies are such that the trainee can learn from. Therefore, whenever a discrepancy has been classified as second level, the pulse synchronization is turned off in order to further analyze the detected discrepancy.

When $T_t = l_T$ is observed, the discrepancy is constantly increasing and having such a magnitude so that it is classified as a third level discrepancy. We believe that whenever a discrepancy belongs to this level, the discrepancy is significant. Therefore, it is logged and reported by the SmartAAR application and the pulse synchronization is again enabled.

Displaying a Physical Discrepancy

To provide only significant feedback, the agent will only be displayed to the trainee whenever a significant physical discrepancy is observed. Whenever a second level discrepancy is detected, the agent will become visible to the trainee. If the discrepancy is believed to not be severe and goes back to be a first level discrepancy, or if it is below the determined threshold (τ) the agent will become invisible.

Note that at this time we only use the magnitude of the discrepancy to suggest severity. Of course, "important" discrepancies will turn out to be the result of doing something in a different but equally acceptable way. This can be likened to syntactical vs. semantic differences in text. We have not yet addressed the latter type of discrepancy characterization and will look to contextual discrepancies to assist in this task.

BUILDING THE EXPERT AGENTS

Unfortunately, expert agents are not trivial to build. Such agents are complex because expert behavior emerges from years of experience and the knowledge can be intuitive. Experts are often unaware of exactly why they act as they do. It is widely accepted that battle-tested soldiers perform better than highly skilled ones without live experience. The rate of success increases as more experience is gained. This infers that the best doctrines cannot cover all possible situations and their correlated actions. The best performance refinement method is live action. In creating the expert agents, the advantage would be to use the knowledge from highly skilled and experienced experts; however, there is evidence in the literature suggesting that agents developed through an interaction between the model engineer and the subject matter expert tend to represent generic, doctrine-like behavior (Calder et al., 1993; Guha, 1991; Ourston et al., 1995; Smith and Petty, 1992). Furthermore, it has been shown that what the experts teach is not necessarily what the experts themselves practice (Deutsch, 1993). Hence, the preferred learning method used is to model the experts by observing their behavior rather than through interviewing them. The method adapted here uses a machine learning algorithm to build the knowledge within the contexts, by observing the experts in action (Fernlund, 2004). An empty context frame was initially developed from expert knowledge and doctrines. This is the very basic structure (i.e. hierarchy) of human behavior in the battlefield. Experts can easy identify the different contexts (e.g. hasty attack, road march, defense, bounding overwatch, etc.) in which a soldier or a group of soldiers can find themselves during different missions. When this empty context structure is defined, the machine learning algorithm is engaged to model the behavioral knowledge within each context and the knowledge to activate correct context (i.e. situational awareness) by observing experts in action.

In order to investigate the feasibility of the SmartAAR by comparison, a test bed of data from live exercises has been established. The data was collected from exercises where two opponent tank platoons made an unanticipated contact. The data collection was made available because the exercise was equipped with an AAR support system that recorded all the soldiers and vehicles movements and actions during the exercise. Data collected from the tanks, and used by the learning algorithm include position, speed, heading, turret-heading, player status, use of Laser Range Finder, Fire and Hit results from fire simulation. Further data used comes from a terrain classification (e.g. forest, open field, water, etc.) of the environment.

As both tank platoons are trainees, there is unfortunately no expert data at hand. The theories of AAR, by comparison, can still be evaluated by using data from one of the platoon to develop the agents and then evaluate the agents by doing AAR by comparison with the other platoon. Earlier research with this machine learning strategy has showed that it is able to create high fidelity individual behavior for the agents created (Fernlund, 2004). In other words, if the behavior of the two platoons differs, it will also be recognized by the comparison of the agent and the opponent platoon.

It is not our intent to further present the machine learning strategy to build context in this paper. Details regarding the machine learning algorithm can be found in Fernlund (2004). Rather, the attention is to present the concepts behind an automatic AAR by comparison.

PHYSICAL DISCREPANCY DETECTION EXPERIMENT

The experiment presented here is based on recorded observations from a real exercise. The simulation that is run on the recorded data is 11 minutes long. An expert agent has been added on top of the recorded data and coupled to one of the trainees to determine the performance of the trainee. The idea of this experiment is to show that the earlier proposed model of detecting physical discrepancies is operational as far as presenting only relevant discrepancies to the trainee. The following control parameters were used for the model. Length of vector D, $l_D = 8$; Length of vector T, $l_T = 16$; Threshold, $\tau = 20$; Threshold, $\delta = 3$. In this experiment, the focus has been placed on detecting a discrepancy in position. Nevertheless, the same model can be used to detect discrepancies in heading or velocity.

A graph of the discrepancies and their classification for the experiment is presented in Figure 4 while Table 1 presents statistics for the experiment. Measuring the number of discrepancies (significant or insignificant) detected in the experiment is done by looking at the

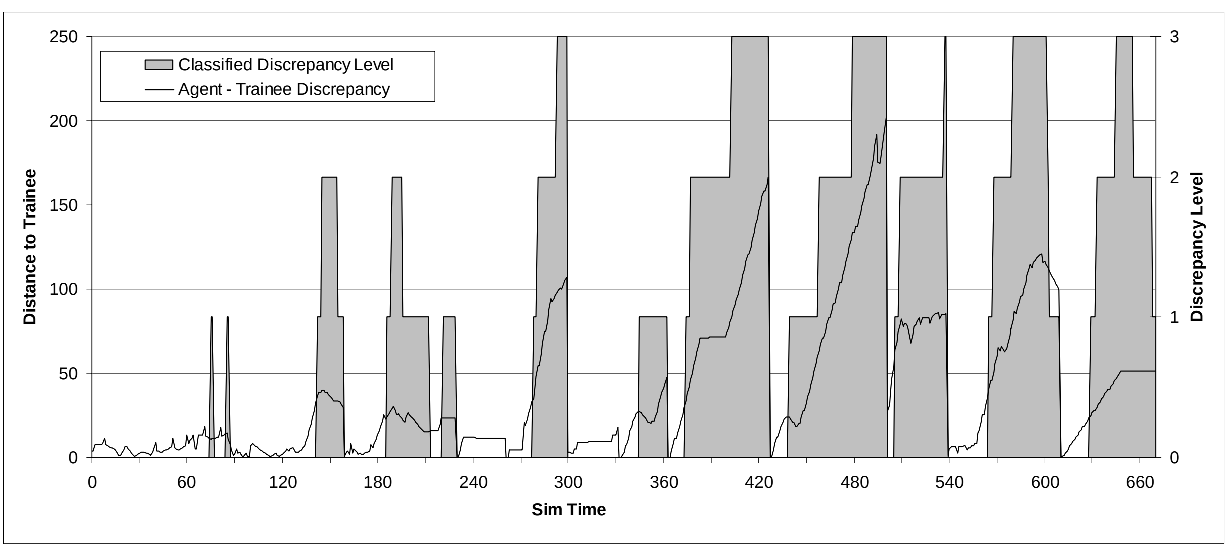

Figure 4. Graph of the discrepancies detected in the experiment.

number of first level discrepancies detected. The graph in Figure 4 shows that a total of 12 discrepancies of Discrepancy Level 1 were found. Furthermore, by looking at the same graph, it shows that further analysis has been carried out 8 times (Discrepancy Level 2) out of the total of 12 discrepancies detected. The number of significant discrepancies is measured by counting the number of third level discrepancies (Discrepancy Level 3), that in our experiment came to a total of 6. The number of detected discrepancies is shown as *Classified* in Table 1 below.

In the graph in Figure 4 some interesting observations can be made. After about 70 seconds in the simulation, the first discrepancy is detected. It is classified as a first level discrepancy but turned out to not become a higher level discrepancy but merely dropped below the threshold level. The second discrepancy (85 seconds) detected is also a first level discrepancy. In this case, a pulse synchronization event took place, replacing the agent back at the exact same situation as the trainee. It might seem awkward to synchronize the agent with the trainee at this point, but our theory is that the trend of the discrepancy is not increasing to the point where any greater and significant discrepancy will occur.

The third discrepancy in the graph of Figure 4 (140 seconds), is first classified as a first level discrepancy. When looking at the trend and the size of the discrepancy after a period of time it is classified as a second level discrepancy. At this point, the synchronization event is turned off and the agent can be observed by the trainee. For this particular discrepancy, one can see that it turned out to decrease after awhile and is yet again classified as a first level discrepancy, thus turning on the pulse synchronization again. We believe that a discrepancy pattern like this provides very little and insignificant information to the trainee.

At about 370 seconds into the simulation, the first significant (third level) discrepancy is detected. It grows fast early and is thus classified as a first level discrepancy for just a short period of time before being classified as a second level and having the simulation turn off the pulse synchronization. In this case the growth rate of the discrepancy tends to slow down for about 30 seconds. As the synchronization is turned off, we observe it to see whether it will continue to grow into a significant discrepancy or if the discrepancy is temporary and thereby decrease. In this case, it starts growing rapidly again, becoming classified as third level. This is communicated to the trainee running the SmartAAR application and the pulse synchronization triggered at about 430 seconds into the simulation.

Table 1 presents some statistics derived from the experiment. It shows the size of the average discrepancies along with the standard deviation for each of the discrepancy levels. The table also presents a time measurement in percent per discrepancy level that the trainee did deviate from the agent. It is quite interesting that the trainee is not believed to deviate at all from the agent for 50% of the time. After analyzing the discrepancies, it can be said that he didn't deviate from the agent for 87% of the time.

Also, Table 1 shows how many times a discrepancy has been classified as a certain level (*Classified*). A total of 12 classified discrepancies of various levels were detected in the experiment. Whenever a discrepancy is detected, it always starts as being classified as a first level discrepancy, if it grows larger and is classified as a second level discrepancy it would still at some point have been classified as a first level discrepancy. Nevertheless, six of the detected discrepancies were classified as being significant and reported back to the trainee. Out of the 12 discrepancies detected, eight of them turned out to be interesting enough to be further analyzed. This means that our method is forgiving to a reasonable number of discrepancies that are not believed to be interesting to the trainee. Once these discrepancies were analyzed, the majority of them turned out to be of such degree that they could be classified as significant.

Table 1. Statistics from the experiment

	$D_t < \tau$	Lvl 1	Lvl 2	Lvl 3
Average Disc.	9.3	36.6	67.1	118.1
Std. Dev.	8.0	22.5	22.6	39.3
Classified	-	12	8	6
Time spent in level (%)	50.3	15.7	20.9	13.1

CURRENT RESEARCH – CONTEXTUAL DISCREPANCIES

The SmartAAR application utilizes the novel modeling paradigm Context-based Reasoning to simulate the human tactical behavior of the agent. We believe that the trainee also operates by using a contextual behavior, making it possible to detect contextual discrepancies between the agent and the trainee. Given that the trainee is operating in our simulation as recorded data, the first step in this process is to evolve a model whose purpose is to determine the trainee's context. Our current model for this is to use an agent for each possible context the trainee can take on and expose it to the same situation as the trainee. Each of these agents generates a behavior. In order to detect the similarity of two behaviors, the position, heading and velocity of the agent and the trainee is compared to each other individually for each context. By doing this comparison we can determine the trainee's current context. Once the trainee's context has been determined it can be easily compared to the agent's context and a discrepancy can eventually be detected. This work is currently on-going and it promises to add an important dimension to AAR by comparison.

CONCLUSIONS

In this paper we proposed an automatic AAR approach that could be applicable, not only in military training, but in wide range of applications. Applying automatic AAR by comparison would enhance the evaluation and possibly be advantageous to more of the participants during an exercise. By giving each participant individualized feedback that focuses on their behavior by comparing it with an expert, it would be the basis for an automatic and self-instructing AAR. For training evaluation, the process of creating take-home packages, or web portals, can now be automated. This would also ease conducting AAR in exercises with actors in different locations (live, virtual or mixed).

We also presented a model that which can detect significant physical discrepancies to be reported to the trainee when conducting AAR. The experiment showed that the model is capable of filtering out a reasonable amount of discrepancies that we believe are insignificant to the trainee. Contextual comparison will serve to filter out additional discrepancies.

The method of SmartAAR by comparison could be further developed to conduct evaluation and feedback in near real time. As the trainees are out in the field performing an exercise, they could be equipped with instrumentation that gives them feedback while they are still in the exercise. This would be possible if the expert agent is executed in the same environment as the trainee (i.e. the agent resides in a simulated environment of the real world) and experiences the same events at the same time as the trainee.

REFERENCES

Calder, R., Smith, J., Coutemanche, A., Mar, J., and Ceranowicz, A. (1993). ModSAF Behavior Simulation and Control, In Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation, Orlando, FL., March, 1993, pp. 347-356.

Deutsch, S. (1993) "Notes Taken on the Quest for Modeling Skilled Human Behavior" Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation, Orlando, FL, March 17 19, pp.359-365.

Endsley, M. (1995), "Towards a Theory of Situational Awareness in Dynamic Systems", Human Factors, (1995) 37(1), pp. 32-64.

Fernlund, H. (2004), "Evolving Models from Observed Human Performance", Doctoral Dissertation, University of Central Florida, Computer Engineering, May 2004, http://etd.fcla.edu/CF/CFE0000013/Fernlund_Hans_K_200405_PhD.pdf

Gonzalez, A. J. and Ahlers, R.(1998), "Context-Based representation of intelligent behavior in training simulations", Transactions of the Society of Computer Simulation, Vol. 15, No. 4, 1998, pp. 153-166.

Guha, R V, 1991. "Contexts: a formalization and some applications." MCC Technical Report ACT-CYC-423-91 December.

Klein, G. A. (1989), "Recognition Primed Decisions", in Advances in Man-Machine Research, W. Rouse (ed.), Greenwich, CT: JAI Press, pp 47-921989.

Ourston, D., Blanchard, D., Chandler, E., and Loh, E. (1995). From CIS to Software, In Proceedings of the Fifth Conference on Computer Generated Forces and Behavior Representation. Orlando, FL., May, 1995, pp. 275-285.

Smith, S.H., and Petty, M.D. (1992). Controlling Autonomous Behavior in Real-Time Simulation. In Proceedings of the Second Conference on Computer Generated Forces and Behavior Representation. Orlando, FL., March, 1992, pp. 27-40.