A Concurrent Model Approach to Scaleable Distributed Interactive Simulation

Hubert Bahr
Hq STRICOM
AMSTI-ET
12350 Research Blvd.
Orlando, Florida
hab@ece.engr.ucf.edu
(407) 679-0815 fax (407) 679-9329

Ronald F. DeMara
Dept. of Electrical & Computer Engineering
University of Central Florida
Box 162450
Orlando, Florida 32826
rfd@ece.engr.ucf.edu
(407) 823-5916

KEYWORDS
live players, networking, communications

ABSTRACT

One of the primary difficulties in providing the interaction between Real and Virtual forces is conveying the large amount of data between the respective players for real-time interaction. In this paper the principles underlying dead reckoning are extended to the limit, by enhancing the prediction capabilities of the interaction between players. The approach uses pairs of full platform models, rather than just sub-element models. This minimizes the data transfer required between local and remote parties.

The proposed approach differs from dead reckoning in that the required correction data could approach zero, this is accomplished by placing a clone of the opponent platform at each location. Thus a high fidelity model is tuned in a closed loop environment to customize the simulation behavior to the unique capabilities of each platform. Concurrently a clone of this model is tuned open-loop at the remote platform. The focus of this paper is to introduce the concept and the components required for implementation. The components include Difference Analysis Engine, Adaptive Constructive Model, Simulator or Manned Weapons Platform Instrumentation, and the Situation Database.

INTRODUCTION

The Army has realized significant benefits by providing training through the interaction of linked simulations. Yet there remains a strong desire to extend these benefits to all trainers by allowing simulated Virtual Forces to interact with troops in the field. One of the primary difficulties in providing the interaction between Live and Virtual forces is conveying the large amount of data between the respective players for real-time interaction. The communications bottleneck is evident in several current systems, such as those at the Army Combat Training Centers (CTCs). At CTCs, the Army instruments a large number of troops to collect data, such as position location, and combat activities. Each center has a Range Data Measurement Subsystem (RDMS) installed to transfer this data to and from the field. The largest and most versatile RDMS provides an average of only 48 bits per second per player and a peak of 2400 bits per second, yet this is an expensive state of the art system, based on the latest commercial products [Bahr 94].

One approach used to reduce the communications traffic required during Distributed Interactive Simulation (DIS) is the use of *dead-reckoning* algorithms. Dead-reckoning takes advantage of knowledge of the physical behavior of entities which dictates that moving bodies can only change speed or direction in certain predictable ways. As long as the moving body does not deviate from this predicted route, there is no need to send additional information from the source monitoring the movement to the receiver using the motion information to determine the current location of the entity.

We introduce an enhanced approach which further reduces the communications bandwidth required for DIS. This paper explains the concept, identifies general requirements, and discusses characteristics of those requirements.

PROPOSED APPROACH

In the proposed *Concurrent Model* approach, the principles underlying dead-reckoning are extended to the limit. In dead-reckoning, at both the source and receiver an algorithm is executed based on the positioning information provided by the source to the receiver. The source continues to compute the current position and compare that to the measured position. As long as the calculated position is within certain error bounds no updates are provided to the receiver. Meanwhile, the receiver calculates the predicted position and uses it as the current position of the moving body as it has confidence that, in the absence of correcting information, this position is accurate within error bounds. In this case, the dead reckoning algorithm represents a model of moving-body positioning. Thus, positioning information is the parameter that is exchanged between two copies of the model.

This approach is extended to predict the interaction between players. The approach uses pairs of full-platform models, rather than only sub-element models. This minimizes the data transfer between the remote interactive parties, and yet maximizes responsiveness, while allowing detailed manipulation or articulated components at the local level. An interactive situation requires pairs of models for each participant. The difference between this and the dead-reckoning approach is this employs two high-fidelity models and ideally the required correction data could approach zero. Essentially, the respective platform is cloned on the target platform. An exact clone would respond identically as the simulated platform and crew, since it is collocated with the target there would be no measurable delays, thus the highest fidelity simulation. In actuality, cloning the crew and platform is impossible, but cloning a model is routine. Thus, the proposal is to place a high fidelity model of the simulated crew and platform on that platform, and in a closed-loop environment tune that model to match the capabilities of the platform. Concurrently, place a clone of the model on the target platform and in an open loop environment apply the corrections made to the reference model to its clone, thereby keeping it a clone of the reference model.

For the purposes of this paper certain assumptions are made. High-fidelity models/simulations are available and technology will provide the necessary, cost effective, computational resources. The focus of this paper is then on: (1) How the differences can be detected. (2) How the models can be adjusted to minimize the errors of prediction. (3) What technological capabilities are required to make it work, in terms of data storage, computational capability, and communications bandwidth. The goal is to tune the high fidelity model to the point it can accurately predict the manned platform's response to all shared information. The range of applicability of this approach is highly dependent on the degree that this goal is met.

Assuming that the goal is totally met, we obtain a model of my opponent within inches of me if necessary. Gone are the restrictions on bandwidth, and latency imposed by shared communications media. Now all the constraints on the interaction are those imposed for realism, i.e., simulation of real world delays, such as, propagation of sound, appearance of damage, speed of projectiles, responsiveness to controls, and reaction times. Certainly, this does not imply that the communication requirement is zero, but that it can be limited to those items that are more commonly sent by tactical communications such as, orders and intelligence information. This information must also include changes in status caused by other interactions. It is still important that each model have access to his normal view of the battlefield, however, that is normally shared information, which will be addressed later in this paper. The modeled opponent I am playing is a specific opponent not some generalization. If I defeat this opponent, his replacement could react totally different. A benefit of this approach is that the set of parameters used to tune the model become a description of the manned platform specific to that platform and crew. This description could be used for comparison purposes to other crews for the same platform with the eventual result that performance goals could be established and training results could be measured.

Detection of differences between simulator and model need to occur in a manner that leads to the correct interpretation of the cause of the difference. This is a field very akin to the Artificial Intelligence area of Model Based Reasoning. The only real distinction is that Model Based Reasoning is normally applied to a diagnostic setting where the model is assumed correct and the observed system is assumed to be at fault. However, there should be no problem in shifting the frame of reference to assume that the observed system is correct and that the model has the defect that needs correction. Yet, there remain several challenging areas in making this practical. The diagnostic routines must be fast as we are still working in real time, and the model must be adaptable to correction. One of the beneficial consequences of making the models adaptable, is that it implies versatility with characteristics defined by a set of parameters. As such, there is high probability that the models used could be generic with the characteristics rapidly tailored by parameters, and potentially adapted using Machine Learning techniques.

By using concurrent models clones at both ends of the communications link we can also take out the delays in the feedback loop. The error detection and correction loop would exist only at the source end and the receiving end would primarily operate-open ended by just utilizing the correction parameters sent from the source end. There always exists the possibility that the correction parameters sent to the receiving end could be lost or have errors so some method of state verification will have to be employed. The goal is to tune the model to where it accurately emulates the crew and platform being modeled. This is much different from just changing the output to agree with reference system. It means analyzing the error to determine what model manipulations are required so the error would not have occurred. In view of the situation that we have humans in the loop, we will never have a perfect model so in this case the goal would be to minimize the average error.

As introduced earlier in this paper, even if we had perfect models and the error correction data was zero, there would remain a requirement for communications of other information between the field platforms and the rest of the participants. What this concept has done is compressed the requirement for data to the minimum essential, and removed transmission delays from apparent reaction times. The data that is required is primarily *shared data*. This is initially information that could be pre-stored at each entity, such as, terrain data base, expected weather conditions, the set of generic models, and force composition. During the exercise this information would have to supplemented by new orders, intelligence information, and changes of status of any player in the field of view. It would also have to add and delete targets from each player's field of view.

We talk about interactive simulation and at times we tend to think of this as one-on-one, whereas, for each platform it is reality team on team. To cope with the team interaction the requirement for models executing on each platform must be extended to include one for each player in the field of view. However, this does not increase the number of required reference models as all clones of player would be subject to the same correction data.

THE CONCURRENT REMOTE MODEL (CRM). Initially, interactive simulations were directly connected as depicted in Figure 1. All information representing the opponent was passed between the players as the information became available. As long as the respective players had equivalent time bases, resolution, and physical distances did not introduce excessive delay, or bandwidth limitations, this approach was valid.

With DIS the goal was to broaden the scope of interplay to the point that location of the simulators, types of simulators, and quantity should not be constraining. In this situation, bandwidth and delays force us to look for methods to restrict the required update frequency and dependency on timeliness. One approach as depicted in Figure 2 is to use dead-

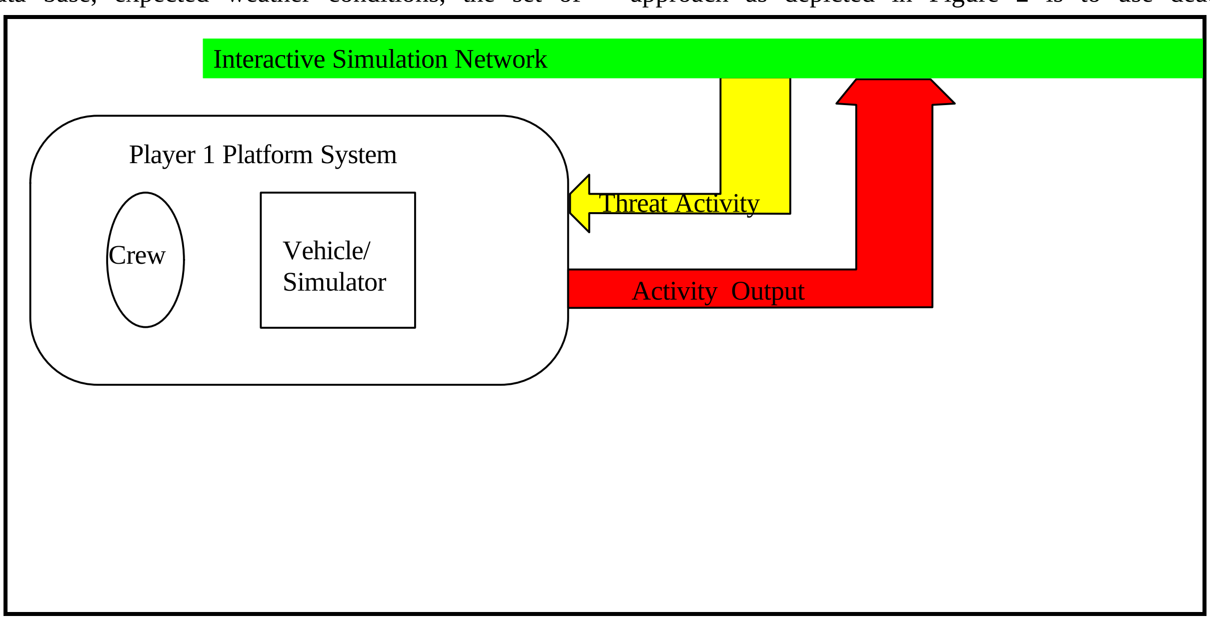

Figure 1. Direct Interconnection

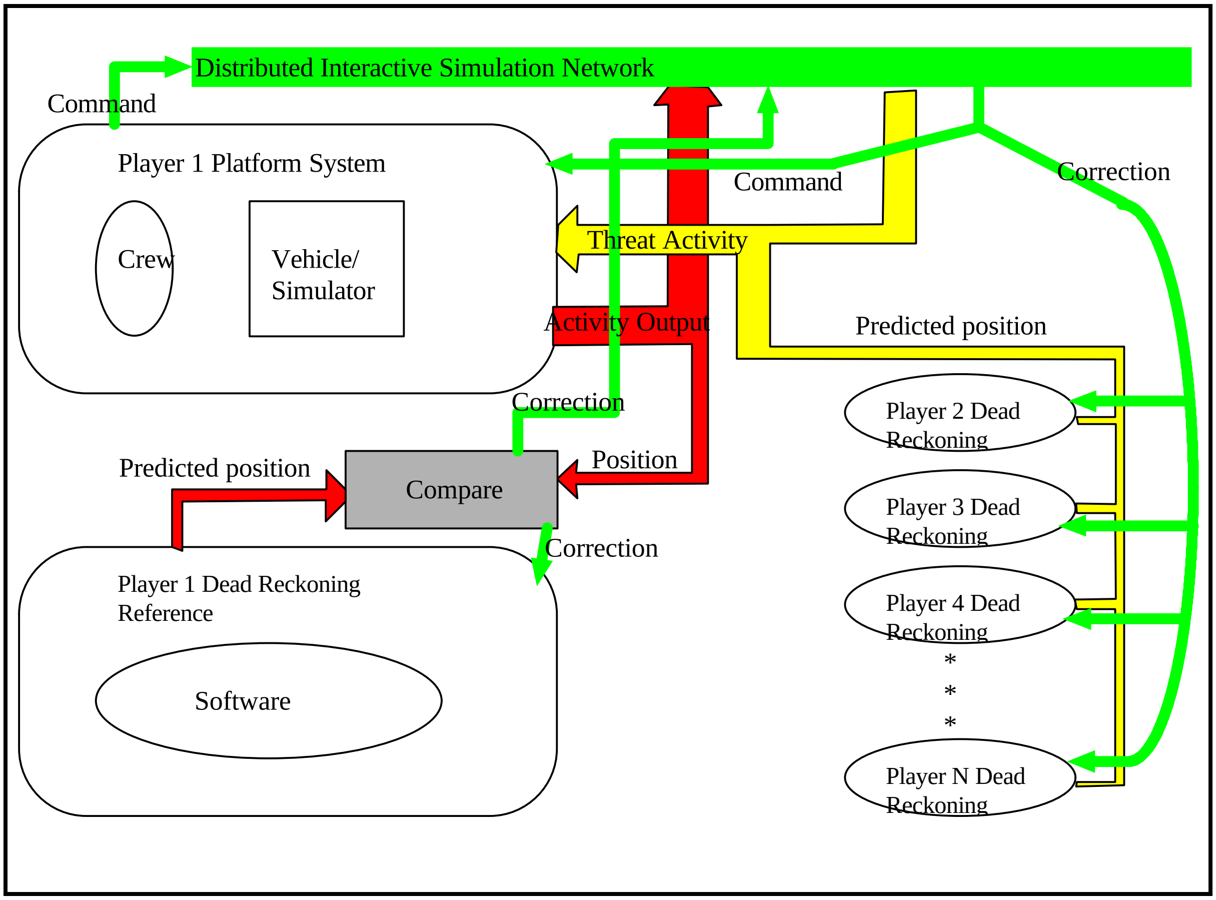

Figure 2. DIS

reckoning models to estimate parameters in between updates. Along with the ever broadening use of the dead-reckoning approach to include over variables The *Concurrent Remote Model* (CRM) shifts the transfer of data away from the interaction parameters to be primarily the typical Combat Situation Information, with model tuning parameters as required. The interaction information as depicted in Figure 3 is still there, available in greater quantity, higher precision, and less delay than by either of the previous methods. The difference is all this information is generated locally at each platform. The CRM platform consists of the major elements identified in Figure 3. They are the crew served simulator or weapons system and its requisite instrumentation. The difference analysis engine that replaces the comparator of the dead-reckoning approach. The adaptive constructive model that serves as either the reference model or a clone of the reference model and the situation data base. Each of these blocks is described in more detail in the following sections. As illustrated in Figure 3, the play of the simulator or the manned platform is only noted locally, the play of the reference model is the "official" view of besides position, we are starting to communicate additional situation data besides just the interaction parameters.
the engagement. This allows a consistent view across the exercise while still allowing individual evaluation to take place at the platform level. Since the reference model and all of clones are changed synchronously, they play the same for a given situation regardless of location. The maintenance of the situation database thereby becomes the primary purpose of the DIS. Since this database should be the same on any platform participating in the same conflict location, the information on this link can be broadcast to all platforms.

Difference Analysis Engine (DAE):

The Difference Analysis Engine (DAE) is the element that compares the performance of the simulator or manned weapons platform with the reference model and develops the parameters that are passed to the reference model and its remote entity clones. It develops the parameters that are used to adapt the ACM. This is the primary place where the states as defined by the training platform (simulator or

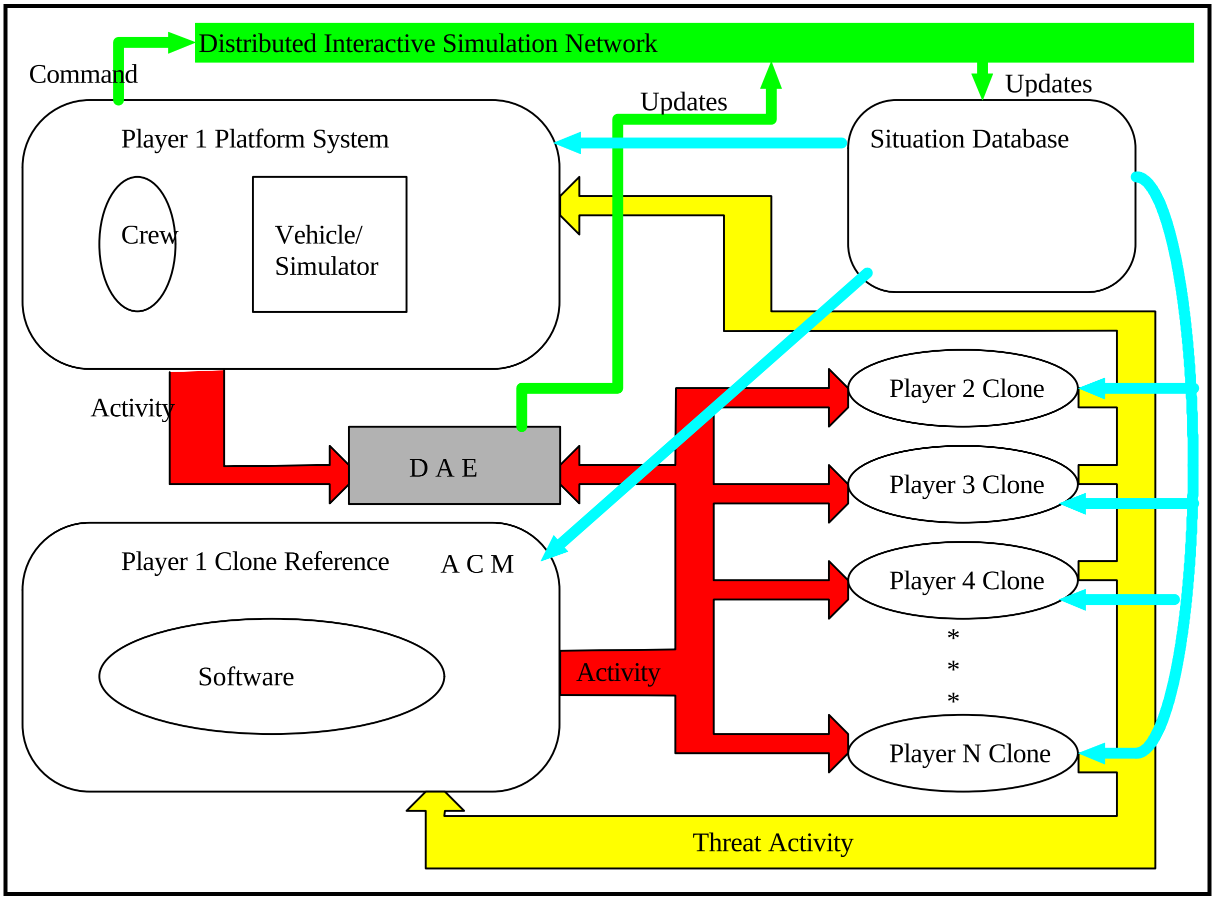

Figure 3. CRM

manned weapons system) are used. The status reported to the rest of the exercise is the output of the reference model. However, in this subsystem the results from the training platform are treated as absolutely correct, the results of the reference model are considered as flawed if differences occur. The parameters generated from its analysis will be used at some time in the future. This delta between current time and future time is design dependent but will be large enough to ensure correct transfer to all clones. It is assumed that changes will be made to the clones synchronously with the reference model. The synchronous time base will probably be based on GPS time. Next to a simulator, this is probably the most highly customized portion of concept. This subsystem has built in to it knowledge of how the simulator generates results, how the reference model generates results, and what the school solution is. It also takes advantage of the crew's history to improve its predictions. This is the subsystem that uses Artificial Intelligence techniques to determine why the parameters need to be changed and what changes to make. This subsystem learns how a specific manned platform performs, converts that knowledge into a set of parameters that it transfers to models of the platform, and expects those models to perform as if they were clones of the manned platform.

Adaptive Constructive Model (ACM):
The *Adaptive Constructive Model* (ACM) is the element that will be cloned to serve as the reference model and the remote entity models. It is anticipated that this model is constructed from a set of generic modules, such as tracked vehicle with turret mounted gun. For example, Gun type is 105MM, power unit is turbine and it is a heavy armor platform. Along with this would be parameters that would differentiate this particular weapons system from the others in its class. In addition to the weapons system capabilities model you need to add the crew model. This gets into modeling things such as reaction time, gunnery accuracy, target recognition, driving tendencies, impulsiveness and etc. This would be an unbounded task except that characteristics of the weapons platform and training narrow the range. Other modeling required is of those characteristics that tend to vary over the course of the battle, or due to battle damage. A key characteristic of these models is that the performance of the model can be adjusted in real-time

during use. The model must continuously generate as an output state, all outputs that determine the location and status of the weapons system and its crew. This includes all weapon firings, and hit and damage information. All parameter changes to this model are applied synchronously to the reference model and all clones. That is, parameter changes are received with the time that they are to be applied. Then when the prescribed time is reached the changes are made. The reference model directly interacts with clones of the target systems, while the clones interact with the reference models of the target systems. The state as generated by the reference model is taken as the state of the weapons platform.

Simulator or Manned Weapons Platform Instrumentation (SMWPI):
Simulator or Manned Weapons Platform Instrumentation (SMWPI) is the set of sensors that are used to determine the state of the Weapons Platform and its crew. It must provide the 4 dimensional location and time measurements, crew status, weapons status, stores status, and articulated components status of the platform. The stores would include for example ammunition, fuel, and water. While this information is readily available on simulators, instrumentation will probably have to be added to most ground weapons platforms. Some research may have to be performed is this area to determine the required accuracy and resolution of these sensors.

Situation Data-base:
The Situation Data-base is data that is stored on each platform required for concurrent simulation to work. Assuming that a model can always adapt more precise information to the level of detail that it requires, the level of detail required for each element is that required by the most discerning threat platform or the reference model. Component data-bases would include such items as depicted in Table 1.

<table>
<tr><th colspan="3">Table 1. Situation Data-base Components.</th></tr>
<tr><td>Terrain-database:</td><td colspan="2">A detailed terrain and features data-base that allows models to exercise the tactics appropriate to the combat environment. Ground vehicles for instance are blocked by impassable areas, may be masked by terrain features or dust. Can sink in dry lakes, etc. This data-base replaces the human observation of the terrain so in some ways it serves as part of the model's eyes.</td></tr>
<tr><td>Threat-platform models, sub-models:</td><td colspan="2">The set of adaptive constructive models that are clones of the reference models on the respective simulators/weapons platforms. Parameters for these models can vary from identifiers of functions, numerical values, to identifiers of pre-tabulated characteristics.</td></tr>
<tr><td>Pre-defined orders:</td><td colspan="2">Standing orders, or orders issued prior to the start of the exercise.</td></tr>
<tr><td>Vulnerability data:</td><td colspan="2">Susceptibility of the local platform to the various threats, as well as the susceptibility of the threat to the local platform.</td></tr>
<tr><td>Weather-data:</td><td colspan="2">Any weather related information that can impact the results of the exercise.</td></tr>
<tr><td>Mine-field, obstacle data:</td><td colspan="2">Contains locations and types of mines and other obstacles. Includes visibility data.</td></tr>
<tr><td>Current state data:</td><td>Status of combat team members:</td><td>Location of team members and their combat status.</td></tr>
<tr><td></td><td>Current intelligence information:</td><td>Contains information about foe above and beyond what sensors can provide.</td></tr>
<tr><td></td><td>Status of each threat platform:</td><td>Current status of each of the targets within field of view.</td></tr>
<tr><td></td><td>Current orders:</td><td>The set of orders that govern platform's objectives and techniques for achieving those objectives.</td></tr>
<tr><td>Predetermined parameters of all potential threats:</td><td colspan="2">A data-base of all players identified as potential participants in the exercise. This allows the initialization of the clones based on an identifier rather than by detailed transferred parameters.</td></tr>
<tr><td>Learned performance of local crew and platform against each threat:</td><td colspan="2">A historical data-base used by the DAE subsystem to initialize the reference model.</td></tr>
</table>

COMMUNICATION DEFINITION

Ideally, all communications would be the standard tactical communications. However, a platform is normally equipped only for its direct mission nets, so although the information could be available in the RF spectrum, it is not available in a form compatible with simulation. Therefore, a minimum DIS style communications requirement will remain even for the ideal case. This minimum requirement would be the transmission of command, control, intelligence, and coordination information used on the battlefield by any involved player whether real or virtual. Only then can a clone and its reference model have any hope of matching the platform's responses. As the communications capability of the tactical system increase so must the information transmitted by the DIS net.

One of the problems with this requirement is that much of this information is primarily transferred by voice, thereby requiring conversion before used for simulation. AI systems are restricted to dealing with bounded discrete sets of inputs. The biggest technical challenge in this area would probably occur in the conversion of coordinating information transferred between platforms of a combat team. The reason this is challenging is that it primarily occurs between platforms in close proximity over communication channels with limited range. While this is a difficult requirement, its solution also has some valuable side benefits, as this is also a precise area where training feedback can have a large payoff. For example, a team leader issues a command for one action, while the team member interprets it as a command for a different action, if this information has been converted by an independent entity, it allows objective assessment of the error. With the correct assessment of the problem, corrective actions for example training, change of procedures, or change of equipment, could be instigated. If a voice to text conversion capability was provided at the source radio, and a feedback display was provided on the commanders display it could provide a redundant capability to assure that the desired command was indeed transmitted. The text could also be transmitted.

The second class of communications data is in the area that this concept attempts to reduce. This is the data required to allow the interaction of the entities. With the CRM concept it is limited to the transmission of parameter changes to tune the clones. In the ideal conceptual case this is zero. The clone approach to the remote approximating model reduces these required updates still further by not only knowing where it is, but also what the objective is, the available paths between the clone and its objective, and all obstacles. Furthermore, the clone is making the same decisions as the crew it is simulating. As a result, it can be directly connected to the platform and play the same as if it was a simulator in the same room. Now if the crew of the simulator starts acting as if they were inebriated, a corrective message needs to be passed to the clone so it starts acting the same way.

PROCESSING OF DISCREPANT RESULTS

The primary results used for battle assessment is the states generated by the clone reference model. These results should be identical at all sites as they are synchronously updated for all copies and they fight the same opponents. They only engage other reference clones. The only place other results are observed are at the individual players, and their local DAE. These results may be used for evaluation of the model but will primarily be used early in the development cycle. This would lead to a situation much like a refereed sports activity. The call stands, you can argue about it and maybe get preferred treatment on the next call, but all you can really do is send a copy of the action and a complaint to the league office. The DAE uses the results to adjust the reference model and its clones, so in reality the disagreement is noted and will influence future results. For this reason it would be beneficial if the model parameters could be transferred with the crew from training exercise to exercise. The comparison of the crews to clones responses can be made available for individual assessment, independent of the exercise results. No matter how perfect the clone, there exists the possibility that results achieved by the simulator or weapons platform differs from the reference model. Some key items that were kept in mind while deciding how to handle this occurrence are:

- This is a simulation whether live or virtual troops are involved.
- The vagrancy’s of actual combat are as amazing as those that are simulated.
- Is it necessary for both views of the interaction agree?
- Are we scoring the actual interaction or are scoring a response to training?

The primary purpose of interactive training is to introduce the variety of responses available from an intelligent foe. Furthermore, it is important that the foe exhibit the reaction speed and responsiveness that could occur in actual combat. If corrective actions are desired, they should be inserted as realistically as possible within the normal response times of human interaction. Avoid negative training. It is believed that the proposed approach does the best job of resolving any noted differences.

SYSTEM INTEGRATION AND APPLICATION

If this concept was used as a goal for trainer development, it would provide for the outline for incremental development of the various elements that would be integrated for a large scale, combined live and virtual exercise. The initial adaptive constructive model of the weapons platform would be developed during the concept exploration phase of the tactical system. It would be used to study the fighting effectiveness of the proposed system, and allow development of proposed tactics. The comparison analysis engine would be developed during the training simulator development and would allow the refinement of both the simulator and constructive model. From the earliest phase, limited copies of these models/simulators could be used in exercises to help in the assessment of the weapon systems potential. During the development phase of the weapon system the embedded trainer would be fashioned to use other systems adaptive constructive models as the threat source. It would also use the Data-base for embedded training. The platoon trainer would add DIS component that would be used for coordinated operations against constructive models. If a training console was added it could be used to send parameter changes to the adaptive constructive models. For force- on-force training the comparison analysis engine would be added to the embedded trainer to allow the interactive training with the opposing force whether virtual or live. At this time any additional sensors would be added to provide the full status of the platform, and any required voice conversion.

It is assumed that the embedded trainer provides the “cockpit display” that shows the threat force and its actions. This display is part of the normal “cockpit display” used during actual platform operation. The data-base used for embedded training includes the terrain database, and model data-base that will also be used for interactive force-on-force training.

CONCLUSIONS

From a subjective viewpoint the proposed Concurrent Remote Model concept will minimize the data that needs to be exchanged among remote player platforms participating in joint exercises. The requirements for implementation of the concept identifies area’s that require further research to provide all the elements. This concept introduces an approach that can be used guide trainer development from embedded, through force-on-force devices that will minimize duplication while maximizing training effectiveness. This concept requires a subsystem to learn how manned weapons platform performs in the course of a battle, quantifies it in a set of parameters that are applied to a model which then fights as a cloned version of that platform. These parameters with their model completely describe the performance characteristics of the platform, and are available for purposes of evaluation and comparison.

REFERENCES

Bahr 94. Combined Event and Process Simulation Model of a Distributed Data Collection System. Hubert Bahr, STRICOM, Conference Record of the 1994 IEEE Southcon Conference. Pages 326-330.

ABOUT THE AUTHORS

Hubert A. Bahr is a Decorated Vietnam Veteran with 27 years of Federal Service. He received his B.S. degree in engineering from the University of Oklahoma in 1972 and his Masters Degree in computer engineering from the University of Central Florida in 1994. For the past 18 years he has been involved with instrumented Force on Force Ranges. He is currently an electronics engineer in the Engineering Directorate of STRICOM. His research interests are in the areas of parallel processing, artificial intelligence, and computer architecture. He is also pursuing his Ph.D. at the University of Central Florida.

Ronald F. Demara received the B.S. degree in electrical engineering from Lehigh University in 1987, the M.S degree in Electrical engineering from the University of Maryland, College Park, in 1989, and the Ph.D. Degree in computer engineering from the University of Southern California in 1992.

From 1987 through 1989, he was an Associate Engineer at IBM Corporation, Manassas, VA, where he designed embedded and complex systems. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, University of Central Florida, Orlando. His research interests are in the areas of parallel processing, artificial intelligence, and computer architecture