A Novel Convergence Algorithm for the Hybrid Strategy Model Packet Reduction Technique

Declan Delaney°, Seamus McLoone*, Tomas Ward*

**Dept. of Electronic Engineering,*
NUI Maynooth,
Maynooth, Co. Kildare, Ireland
E-mail: seamus.mcloone@eeng.nuim.ie

°Dept. of Computer Science,
NUI Maynooth,
Maynooth, Co. Kildare, Ireland
E-mail: decland@cs.may.ie

Abstract – **Several approaches exist for maintaining consistency in Distributed Interactive Applications. Among these are techniques such as dead reckoning which use prediction algorithms to approximate actual user behaviour and thus reduce the number of update packets required to maintain spatial consistency. The Hybrid Strategy Model operates in a similar way, exploiting long-term patterns in user behaviour whenever possible. Otherwise it simply adopts a short-term model. A major problem with these techniques is the reconstruction of the local behaviour at a remote node. Using the modelled dynamics directly can result in unnatural and sudden jumps in position where updates occur. Convergence algorithms are thus required to smoothly reconstruct remote behaviour from discontinuous samples of the actual local behaviour. This paper makes two important contributions. Primarily, it proposes a novel convergence approach for the Hybrid Strategy Model. Secondly, and more fundamentally, it exposes a lack of suitable and quantifiable measures of different convergence techniques. In this paper the standard smoothing algorithm employed by DIS is used as a benchmark for comparison purposes.**

Keywords – **Hybrid Strategy Model, Convergence, Distributed Interactive Applications**

I INTRODUCTION

Techniques that generate and transmit approximate models of local user behaviour have been developed to reduce the number of update packets that must be transmitted between participants of Distributed Interactive Applications (DIAs). One such technique is dead reckoning, which reduces network traffic by preventing update packets from being sent if the local participant's position has not varied from a model of that position by a pre-established threshold (IEEE 1993; IEEE 1995). However, dead reckoning ignores information pertaining to the structure of the environment, the historical behaviour of users within the environment and the objectives of the users within the environment. Consequently a technique known as the Hybrid Strategy Model or HSM has recently been developed which exploits this information by identifying long-term patterns in user behaviour. By combining these longer-term models or strategies with a short term model, such as dead reckoning, a reduction in network traffic has been shown (Delaney et al. 2003b; Delaney et al. 2003a; Marshall et al. 2004a; McCoy et al. 2005).

An important issue arising from this approximation of local user behaviour is the reconstruction of a faithful replication at a remote location given this often divergent model of the local behaviour. If the received updates are rendered directly, the resulting behaviour can be disjoint and unnatural. To overcome this problem, techniques are employed to smoothen the behaviour. These are known as convergence techniques. In the case of dead reckoning first order and cubic spline algorithms are used (IEEE 1993; Lin and Schab 1994).

This paper addresses the issue of a suitable convergence algorithm for the HSM approach. It proposes a convergence technique based on the weighting of user behavioural models using a suitable blending function. This technique is visually compared with the standard DIS smoothing algorithm (IEEE 1993). In carrying out this work, it was discovered that there is a dearth of objective measures in effectively evaluating convergence algorithms (or at least the authors are unaware of any existing objective criteria for evaluating such algorithms). The paper will focus throughout on user entity motion dynamics, as such movement is responsible for the

majority of update packets generated in Distributed Interactive Applications.

The structure of the paper is as follows. Section II summarizes the Hybrid Strategy Model. Section III describes the novel convergence algorithm, based on the blending of user behavioural models. The standard convergence algorithm used in DIS is also introduced here. Section IV presents a series of graphical simulation results, comparing outputs from the HSM algorithm using (i) no convergence, (ii) the standard DIS convergnce and (iii) the proposed convergence technique. Some concluding remarks in addition to future work are presented in section V.

II THE HYBRID STRATEGY MODEL

Distributed Interactive Applications involve potentially thousands of simultaneous participants. In an effort to reduce the number of update packets transmitted between users, models of user dynamics are employed so that only samples of the dynamics need to be communicated. User dynamics can then be reconstructed remotely using the samples as input to the model. The most commonly used model is dead reckoning. This operates by maintaining a short-term model of the behaviour of each entity at each node. When the modelled and the actual behaviour for an entity differ by a threshold value, an update packet indicating its current state is sent to all other nodes. Between updates the remote nodes employ the same short-term dead reckoning model.

Our work to date has shown that long-term behaviour may also be modelled to a greater or lesser extent and that the use of such a model would complement the short-term model (Delaney 2005). This motivated the development of the Hybrid Strategy Model (HSM) to further reduce the number of update packets that need to be transmitted. The hybrid model, M, for entity dynamics is of the following form:

$$M = p\chi + (1-p)\Gamma \qquad (1)$$

where χ is any conventional dead reckoning model, Γ is one of several long-term entity strategy models and p is a binary weighting factor governed by:

$$p = 1 \text{ for } \|M - \Gamma\| \geq \theta$$
$$= 0 \text{ otherwise} \qquad (2)$$

where θ represents a distance measure threshold between the modeled behavior and the long term model Γ. In this way a long-term model, Γ, based on a priori data can be employed when the entity movement is 'close' to such trajectories. The particular model used is dynamically chosen so that improved fits can be obtained over single model approaches. At any instant in time the nearest model from the set available is used in equation (1).

Long-term models can be generated from historical data and knowledge of the environment. Here, long-term models were constructed by recording actual user dynamics in two separate environments. To do so, users navigated between a start position and a target position with the objective of doing so in the shortest time possible; each path traced out by the user entity is referred to as a *trajectory*. The optimum behaviour to realise an objective is referred to as a *strategy*. A trajectory that attempts to mirror the strategy is called a *steady-state trajectory*. The short-term model employed in the HSM is dead reckoning. In this paper the models all relate to user spatial dynamics.

III THE NOVEL CONVERGENCE ALGORITHM

Each update packet generated by the HSM technique indicates that a transition has occurred from one model to another. If this transition is implemented instantly, the resulting user movement may appear unnatural and disjoint. The convergence algorithm serves to smoothen the transition, hence increasing the playability of the application and improving the realism of the reconstructed user dynamics. This is illustrated in Figure 1(a)-(b).

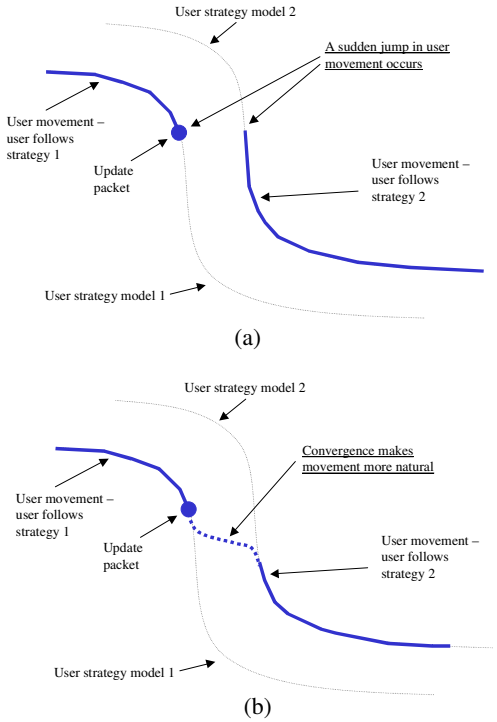

Figure 1(a)-(b): (a) A sudden jump in position occurs when there is a transition between models; (b) Convergence smoothens the transition, improving movement fidelity.

Weighted model convergence blends two models, the model that was used up to the most recent update packet and the new model that best describes the remote user's behaviour. This is illustrated in Figure 2 where a transition has just occurred from strategy 1 to strategy 2. (x_1,y_1) is the point on strategy 1 at which the entity was located when an update packet was received. Starting from this point the weighted model convergence uses a section of the strategy 1 model up to point (x_4,y_4). Likewise, a section of strategy 2 from point (x_3,y_3) up to point (x_2,y_2) is chosen. Here, (x_3,y_3) is found by an orthogonal projection of (x_1,y_1) onto strategy 2. Both sections are equal in length and contain the same number of points, N. These two coordinate data sets are then blended according to a weighting scheme to obtain a smoothened path.

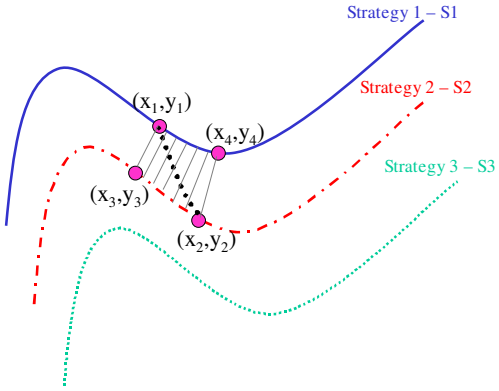

Figure 2: Convergence means that there is a smooth transition from one strategy to another by blending multiple points on each strategy.

This process of blending the two long-term models by an appropriate weighting scheme is described by the following equation:

$$W = p\Gamma_1 + (1-p)\Gamma_2 \qquad (3)$$

where

W is the model resulting from the blending of Γ_1 and Γ_2;
Γ_1 and Γ_2 are long-term models;
p is a weighting factor with $0 \le p \le 1$.

The weighting factor, p, may be generated by a suitable normalized weighting function. A natural blending of models suggests giving more weighting when close to a strategy model and covering the intermediate distance quickly. Intuitively we want a gradual smooth change from the first model, quickly head for the second model and then gradually converge to the second model – see Figure 1(b). This desired behaviour suggested the use of a sigmoid blending function:

$$p(x,a,b) = \frac{1}{1+e^{-a(x-b)}} \quad \text{(sigmoid)} \qquad (4)$$

where

$x \in [0,1]$;
$a, b \in \Re$.

A *rate of convergence* is also specified. This determines the speed with which the weighting function begins using the new model and forgets the old model. It is expressed as a real number between 0 and 1 and is an essential factor of the convergence protocol. In the simulation test bed the x input values to the convergence functions depends on the rate of the convergence parameter value. Beginning from 0, x is incremented by the rate of convergence parameter value on each clock cycle of the test bed (every 32ms). This continues until x reaches 1.

In this paper, our convergence technique is compared with the standard DIS convergence approach. In the latter case an interpolation algorithm usually based on cubic splines is employed to interpolate between the last modelled position of the entity and new position communicated in the update packet. The interpolation normally speeds up the entity so that it converges to a realistic position rather than jumping directly to the position indicated in the update packet.

IV SIMULATIONS AND RESULTS

In this section the operation of two convergence algorithms is examined: the standard DIS smoothing algorithm and the novel weighted convergence algorithm using a sigmoid blending function. Data was recorded from a number of users for two distinct environments as described in (Delaney et al. 2003b; Marshall et al. 2004a; Marshall et al. 2004b). Each user navigated from a fixed start location to a fixed end location and the trajectories they traced out were sampled using both dead reckoning and the HSM. These samples were then used to reconstruct the trajectories using both convergence algorithms.

To facilitate a visual comparison between the various plots produced, two user trajectories (one from each of two different environments) are used for illustration throughout: trajectory T1 from test environment 1 and trajectory T2 from test environment 2. These were chosen because they are steady state trajectories with interesting entity dynamics. They thus provide several state update packets and hence require the use of convergence throughout the trajectory length. T1 and T2 are indicated as dashed lines in all Figures.

As a prelude to applying the convergence algorithm, HSM update packets are generated for the two trajectories.

These update packets are then used to reconstruct the trajectories a remote user would reconstruct and render if there was no convergence. Figure 3(a)-(b) shows the update packets generated for T1 and T2 by the HSM with a threshold value of 10 world units and no convergence algorithm. Stars indicate the update packets and the dashed line represents the original entity trajectory.

The reconstructed trajectories are shown as solid lines and it can be seen that although they maintain the overall shape of the original user trajectories, they are quite jagged and involve sudden jumps in position each time an update arrives. The sections produced are simply segments of the strategy models that match the original trajectory. The update packets lie between these segments and indicate a transition from one long-term model to another.

Clearly, the resulting trajectory rendered to the remote user is highly unrealistic, with poor fidelity to the original user trajectory, particularly in terms of smoothness. Hence the need for a convergence algorithm.

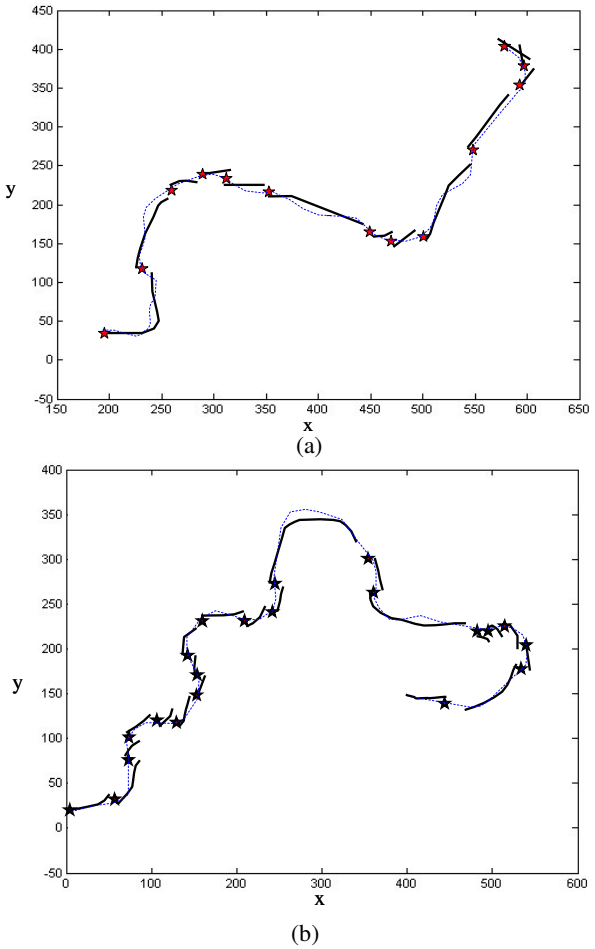

Figure 3(a)-(b): Plots of the remote reconstruction of (a) T1 and (b) T2 using update packets generated by the HSM, with no convergence algorithm. The dashed line represents the original trajectory.

The first attempt at smoothing the entity's remote trajectory to make it more realistic was based on point-to-point first order linear convergence, the standard smoothing algorithm specified in the IEEE DIS standard (IEEE 1993; IEEE 1995). Application of the convergence algorithm to the update packets generated by the HSM is shown in Figure 4(a)-(b). The rate of convergence was set to 0.015 and the convergence distance was 10 world units.

It can be seen that the general character of the original local trajectory indicated by the dashed line is maintained in the reconstructions. The gaps in Figure 3(a)-(b) have been smoothened over. However, the reconstruction shown by the solid line is still somewhat jagged with sudden, unnatural changes of direction in several places.

In Figure 4(b) the boxed area indicates an inaccurately reconstructed section. This is a consequence of a constant convergence rate being employed that is slower than the velocity of the entity. This can be fixed by varying the convergence rate throughout the simulation. This was not implemented here.

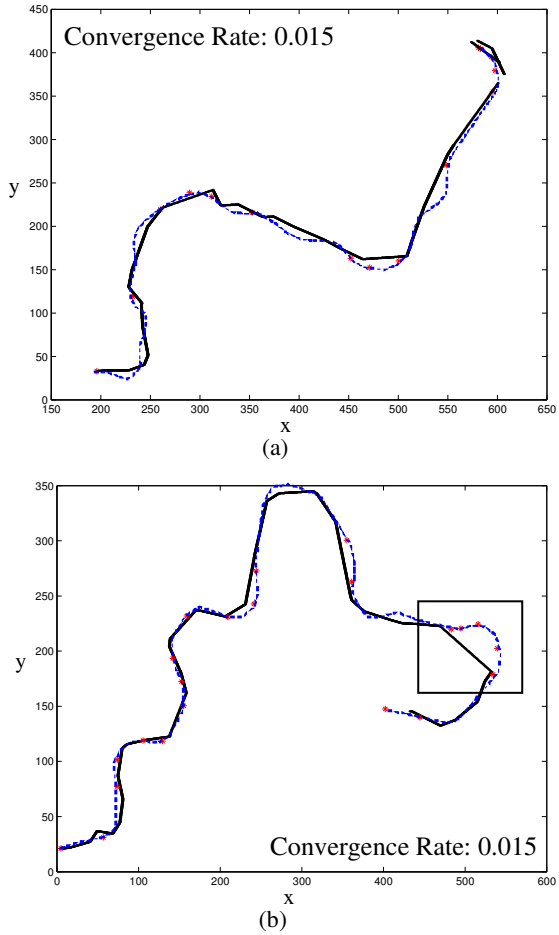

Figure 4(a)-(b): Plots of the remote reconstruction of (a) T1 and (b) T2 using update packets generated by the HSM, with the point-to-point convergence. The dashed line represents the original trajectory.

The novel weighted convergence approach is now considered. The HSM involves transitions between different models and each transition involves two models. The weighted sigmoid convergence algorithm blends the two strategy models according to a sigmoid blending function. Three parameters can be varied – the rate of convergence and the two parameters that determine the shape of the sigmoid curve. Figure 5(a)-(b) illustrates a reconstruction using a typical sigmoid blending function with a = 15, b = 0.5 and the rate of convergence = 0.015. The local trajectory is shown as a dashed line.

Once again the resulting curves show that the gaps in Figure 3(a)-(b) have been smoothened over. However, the reconstruction remains somewhat jagged with some unnatural movement in the form of oscillations as indicated in Figure 5(b). The performance of the convergence algorithm is dependant on the parameters of the sigmoid function. In an effort to understand the impact of the sigmoid parameters (i.e. a and b) on the reconstruction, additional smoothing tests were carried out.

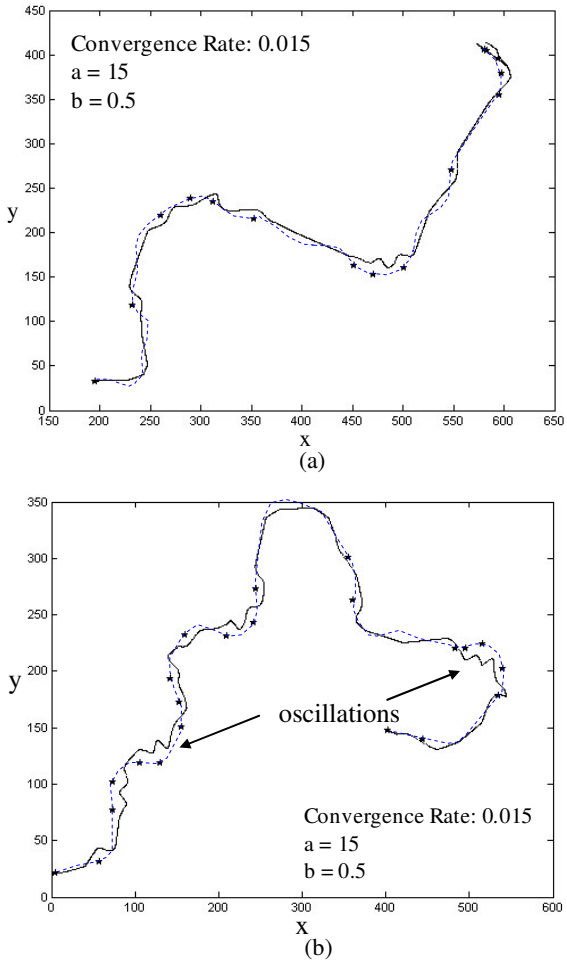

Figure 5(a)-(b): Plots of the remote reconstruction of (a) T1 and (b) T2 using update packets generated by the HSM. Sigmoid weighted convergence is used.

The b value determines the point at which both models are given equal weighting in the transition. High values of b mean that this point is further from the start of the transition and so the old strategy curve will be dominant in the blending process. Lower values of b mean that the new strategy will dominate for more of the transition. A value of 0.5 results in both being weighted equally when the transition is half way complete. Simulations showed that for the environments examined, the most suitable b value is 0.5. In most applications it can be envisaged that the influence of the old long-term model will decrease and the new strategy will increase at a constant rate, with both models contributing equally when the transition is 50% complete.

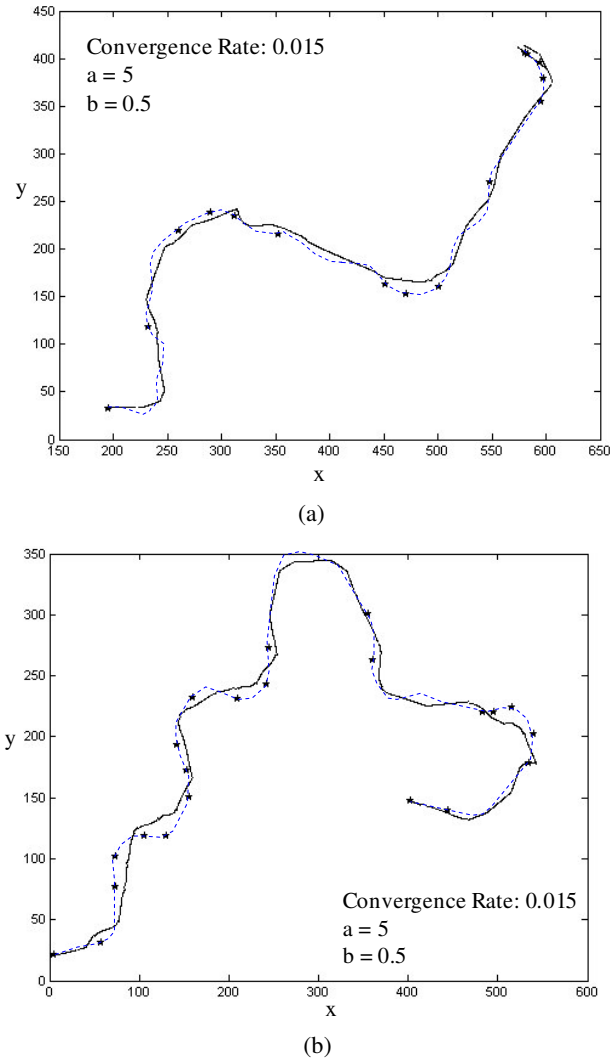

Figure 6(a)-(b): Plots of the remote reconstruction of (a) T1 and (b) T2 using update packets generated by the HSM. Sigmoid weighted convergence is used.

The a value determines the slope of the sigmoid – the greater the a value the greater the slope. Simulations show that this value has a direct effect on the oscillations noted in Figure 5(a)-(b). Low values of a serve to

attenuate the oscillations and as *a* increases, the severity of the oscillations increases. A high value indicates a steep slope and only serves to speed up the convergence process. Earlier, in figure 4 it was noted that the rate of convergence was too slow resulting in poor accuracy in parts of the output. Here, the convergence rate is the same but the *a* value can effectively increase this rate and hence, the oscillations obtained. Clearly, a compromise between model accuracy, realism of a player's movement and the game playability is required to choose suitable values for both parameters *a* and the convergence rate.

Simulations showed that values of a = 5 and b = 0.5 produced good results for a convergence rate of 0.015. The associated plots are shown in Figures 6(a)-(b).

Visually comparing these plots with those in Figures 4(a)-(b) would seem to indicate that the former obtains a smoother, more natural output at the expense of model accuracy. However, from the remote point of view this is an acceptable trade-off in a world where playability and the end user perceptual experience are more important than achieving absolute consistency.

V CONCLUSIONS

This paper has described a novel convergence algorithm based on the blending of long-term behavioural models using a sigmoid function. The technique was evaluated by visually comparing it to both the original user entity trajectory and to the smoothing algorithm used in the IEEE DIS standard. It was found that the smoothness of the reconstructed behaviour using the weighted sigmoid convergence algorithm depends on the rate of convergence, with a high rate of convergence resulting in a jagged reconstruction and a low rate of convergence resulting in the loss of behavioural detail. The results also depend on the parameters employed in the sigmoid function. Reducing the *a* value increases the naturalness of the reconstructed curve. A central *b* value of 0.5 was found to be most suitable.

The HSM technique and the convergence algorithms involve a number of input parameters that determine the efficacy of the algorithms. Our work on convergence algorithms has highlighted the lack of objective criteria for evaluating the effectiveness of such algorithms. There is therefore a need for further studies to investigate possible objective evaluation measures in terms of smoothness and fidelity to the original behaviour. In addition, the use of psycho-perceptual data will be examined to measure the impact of different convergence algorithms on the end users.

ACKNOWLEDGEMENT

This work is supported by Science Foundation Ireland and Enterprise Ireland under grants no. SC/2002/129/ and IRCSET/SC/04/CS0289.

REFERENCES

Delaney, D., T. Ward and S. Mc Loone (2003a). On Reducing Entity State Update Packets in Distributed Interactive Simulations using a Hybrid Model. 21st IASTED International Multi-conference on Applied Informatics, Innsbruck, Austria, February 10-13, 833-838.

Delaney, D., T. Ward and S. Mc Loone (2003b). Reducing Update Packets in Distributed Interactive Applications using a Hybrid Model. 16th International Conference on Parallel and Distributed Computing Systems, Reno, USA, August 13-15, 417-422.

Delaney, J. D. (2005). Latency Reduction in Distributed Interactive Applications using Hybrid Strategy-based Models. Electronic Engineering. National University of Ireland, Maynooth. PhD. pages: 271

IEEE (1993). IEEE Standard for Distributed Interactive Simulation - Application Protocols. IEEE Std 1278-1993. I. C. Society. New York, IEEE.

IEEE (1995). IEEE Standard for Distributed Interactive Simulation - Application Protocols. IEEE Std 1278.1-1995 (Revision of IEEE Std 1278-1993). I. C. Society. New York, IEEE.

Lin, K.-C. and D. E. Schab (1994). *The performance assessment of the dead reckoning algorithms in DIS*. Simulation **63**(5): 318-325.

Marshall, D., D. Delaney, S. McLoone and T. Ward (2004a). Exploring the Spatial Density of Strategy Models in a Realistic Distributed Interactive Application. International Symposium on Distributed SImulation and Real-Time Applications (DS-RT '04), Budapest, Hungary, October 21-23, IEEE, 210-213.

Marshall, D., A. McCoy, D. Delaney, S. McLoone and T. Ward (2004b). A Realistic Distributed Interactive Application Testbed for Static and Dynamic Entity State Data Acquisition. Irish Signals and Systems Conference (ISSC), Belfast, June 30 - July 2, IEE, 83-88.

McCoy, A., S. McLoone, T. Ward and D. Delaney (2005). Dynamic Hybrid Strategy Models for Networked Multiplayer Games. 19TH European Simulation Multiconference (SCS-ESM 2005), Riga , Latvia, June 1 - 4, The Society for Modeling and Simulation International, 727-732.