Formalizing a Framework for Dynamic Hybrid Strategy Models in Distributed Interactive Applications

Aaron McCoy, Tomás Ward, Seámus McLoone and Declan Delaney

Department of Electronic Engineering,
National University of Ireland Maynooth,
Maynooth, Co. Kildare,
Republic of IRELAND
E-mail: amccoy@eeng.nuim.ie

Predictive contract mechanisms such as dead reckoning are widely employed to support scalable remote entity modelling in Distributed Interactive Applications (DIAs). By employing a form of controlled inconsistency, a reduction in network traffic is achieved. Previously, we have proposed the Dynamic Hybrid Strategy Model (DHSM) as an extension to the concept of dead reckoning that adaptively selects extrapolation models based on the use of local performance criteria. In this paper, we formalize the notion of the DHSM as a generalized framework for network traffic reduction in DIAs, alongside a set of consistency metrics for use as local performance criteria.

Keywords – distributed interactive applications, predictive contract mechanisms, dead reckoning, consistency, scalability.

I INTRODUCTION

A Distributed Interactive Application (DIA) is a distributed virtual reality system through which individuals can share information via individual and collaborative interaction with each other and their environment [1]. Distributed Interactive Applications offer the realization of simulated virtual worlds that embody a modern extension of communication, encompassing the concepts of shared time, shared space and shared presence [2]. The definition of a DIA encompasses a diverse range of applications that have seen rapid advances in technology and global popularity due to the widespread availability and ease-of-use of the Internet [3].

The two primary factors limiting the large-scale deployment of a DIA are network latency and network bandwidth. Network latency refers to the delay in communication between two end-points, while network bandwidth refers to the rate at which data can be communicated per unit time between two end-points. High network latency and low network bandwidth capacity represent the largest contributors to the difficulties faced by DIAs in maintaining and supporting: (a) shared state consistency, (b) potential scalability, and (c) real-time interactivity.

In this paper, we are concerned with the use of predictive contract mechanisms for the reduction of network traffic in DIAs, including the well-known dead reckoning algorithms formally defined in the IEEE Standard for Distributed Interactive Simulation (DIS) [4]. Traditional dead reckoning mechanisms often ignore available contextual information that may be influential to the state of an entity, sacrificing remote predictive accuracy in favour of low resource and computational overhead [2]. Previously, we have proposed an extension of dead reckoning, known as the Dynamic Hybrid Strategy Model (DHSM), that builds upon the foundation of the Hybrid Strategy Model (HSM) technique [5, 6]. Both the HSM and the DHSM are hybrid predictive contract techniques that dynamically select remote extrapolation models based on local evaluation of current entity dynamics.

In this paper, we formalize the concept of the DHSM as a generalized framework for the reduction of network traffic in DIAs, providing a specification and associated algorithm for implementation in real-world scenarios. We formalize a set of corresponding consistency metrics that can be used as performance criteria for the dynamic (and adaptive) selection of extrapolation models, including the use of time-space consistency metrics as a combination of spatial and temporal threshold dimensions.

The rest of this paper is organized as follows. In Section 2, we review work relating to both predictive contract mechanisms and measures of consistency. In Section 3, we formally define consistency metrics in relation to dead reckoning mechanisms. In Section 4, we present our proposed DHSM framework. Section 5 offers conclusions and suggestions for future work.
II RELATED WORK

a) Predictive Contract Mechanisms

Dead reckoning is the most widely employed form of predictive contract mechanism in current use. It was introduced during the development of SIMNET [7] and subsequently formally defined within the IEEE Standard for Distributed Interactive Simulation (DIS) [4]. Controlling hosts interacting within a DIA are required to maintain two parallel models for each entity under their control – a high-fidelity dynamical model that represents true entity state as determined through autonomous local control, and a low-fidelity dynamics model representing remotely approximated entity state. By periodically comparing the deviation between the two models against an error threshold or violating a transmission timeout criterion, controlling hosts can locally determine when state information is required to be replicated to remote hosts. Hence, by employing a form of controlled inconsistency, dead reckoning achieves a reduction in network traffic.

The methodology behind dead reckoning (that is to say, the distribution of low-fidelity models to remote hosts and the subsequent (non-periodic) update of those models based on local computation performed by the controlling host) is quite often referred to as the ‘players and ghosts’ paradigm [8]. In this case, it is assumed that each entity is owned by a single controlling host (the player), who is required to update the remote simulations of that entity (the ghosts) when the deviation between local and remote state exceeds the given threshold(s).

Dead reckoning mechanisms consist of two main components, namely prediction and convergence [2]. Prediction algorithms (often known as extrapolation models) are defined in Section IV. They specify how entity state is remotely extrapolated between entity state updates (ESUs). Convergence algorithms define how remote entity state is smoothly corrected due to errors arising from inaccurate extrapolation. Detailed discussion of convergence algorithms is beyond the scope of this paper.

As previously described, dead reckoning schemes ignore available contextual information that may be influential to the state of an entity. To compensate, a number of adaptive extrapolation schemes have been proposed throughout the literature, aimed at reducing the error associated with remote extrapolation. These schemes can be distinguished according to the nature of computation associated with the adaptive process:

1. Local Computation Schemes: Local computation schemes require controlling hosts to select an appropriate extrapolation model for each entity under their control, distributing application-wide computational overhead linearly with the number of hosts [5, 6, 9, 10, 11];

2. Remote Computation Schemes: Remote hosts are required to adaptively select extrapolation models for each remotely modelled entity. Computational overhead is distributed throughout the application quadratically with the number of hosts [12, 13].

In order to determine when to send a new entity state update (ESU), dead reckoning mechanisms require an explicit means of measuring the current degree of dynamic shared state consistency between the high-fidelity and low-fidelity entity models (in a practical sense, the level of inconsistency accrued between the local and remote representations of an entity will be higher than that estimated locally due to the presence of network latency).

Traditionally, dead reckoning mechanisms have relied on spatial-based and temporal-based metrics as orthogonal quantities, merely issuing an entity state update (ESU) when either has been exceeded (these metrics are used under the guise of ‘error threshold’ and ‘transmission timeout’, respectively). The error threshold is designed to bound the remote predictive error associated with the extrapolation of entity state, while the transmission timeout seeks to guarantee a minimum frequency of ESU transmission at the cost of potentially redundant bandwidth usage. Used in isolation, temporal-metrics are quite often referred to as frequent state regeneration with heartbeat packets [2, 14]. In order to optimise the selection of these parameters, various adaptive error threshold schemes have been proposed throughout the literature [8, 10, 15, 16, 17, 18, 19], along with a series of techniques for reducing the reliance on the transmission timeout criterion [14].

b) Measures of Consistency

In the literature, drift distance has been defined as a metric for evaluating the consistency in a DIA that represents absolute spatial difference between local and remote entity models [20]. Phase difference has been defined as a consistency metric that represents the temporal difference between the rendering times for local and remote entity models [21]. Export error has been defined as the deviation in physical time of the remote entity model from the local entity model [22]. Deviation and uncertainty have been defined as absolute spatial difference and the size of the area in which an entity could reside in, respectively [17].

More recently, the use of a formalized time-space consistency metric has been proposed to evaluate the effect of inconsistency on user perception in a DIA, and it is given by the following equation [23]:

$$\Omega = \begin{cases} 0, & \text{if } |\Delta(t)| < \epsilon, \\ \int_0^{t+\tau} |\Delta(t)| dt, & \text{if } |\Delta(t)| \geq \epsilon, \end{cases} \tag{1}$$

where Δ(t) is the spatial difference between local and remote entity models, ε is the minimum perceivable error (from a user perspective), t₀ is the time at which...
the difference starts, and \( \tau \) is the duration for which the difference persists. When \( \Omega \) equals zero, absolute consistency occurs. Time-space consistency metrics have been proposed for use as local error thresholds within dead reckoning mechanisms [24].

In the next section, we formalize the concepts of spatial, temporal, and time-space consistency metrics within the context of dead reckoning mechanisms. It should be noted that our proposed set of definitions are specific to our particular application domain, and are not intended as general metrics. This means, for instance, that our definition for temporal consistency is independent of other definitions proposed in such papers as [25], which typically deal with intervals of validity for sampled variables.

III CONSISTENCY METRICS

a) Concepts of Time

We view a DIA as a discrete domain where time is modelled as a monotonically increasing sequence of time instances bounded by a minimal element \( t_0 = 0 \):

\[
T = \{ t_i \}, \, \forall \, i \in \{ 0 \} \cup \mathbb{Z}^+ \tag{2}
\]

Given the above model of time, we define the closed interval between any pair of time instances \([ t_m, t_n ]\) as:

\[
T_{[t_m, t_n]} = \{ t_i | t_i \in T, \, t_m \leq t_i \leq t_n \} \tag{3}
\]

We define a temporal-metric \( \tau \) for referencing wall-clock time as the constant duration (as measured in seconds) between successive time instances:

\[
\tau = t_i - t_{i-1}, \, \forall \, i \in \mathbb{Z}^+ \land \, t_i \in T \tag{4}
\]

Hence, a given time instance \( t_i \) can be de-referenced to the corresponding wall-clock (physical) time by:

\[
t_i = i \tau, \, \forall \, t_i \in T \tag{5}
\]

Equations (2-5) provide a time-base for the definition of dead reckoning consistency metrics, as proposed throughout the following series of subsections.

b) Spatial Consistency

We define a measure of spatial consistency for entity \( E \) interacting in a DIA at time \( t_i \) as the instantaneous, absolute positional difference between the local low-fidelity and high-fidelity models (denoted as \( x^{low} \) and \( x^{high} \) respectively) for that entity:

\[
\Delta^E_{t_i} = \left| x^{high}_{t_i} - x^{low}_{t_i} \right| \tag{6}
\]

Hence, for the purposes of dead reckoning, we say an entity \( E \) is spatially consistent (SC) at time \( t_i \) if and only if the measured spatial consistency for the entity at that time is less than or equal to a bounded spatial error threshold \( \delta_\phi \):

\[
SC^E_{t_i} \Leftrightarrow \Delta^E_{t_i} \leq \delta_\phi \tag{7}
\]

c) Temporal Consistency

We define a measure of temporal consistency for an entity \( E \) interacting within a DIA as the total elapsed wall-clock duration (measured in seconds) during the closed interval \([ t_r, t_l ]\), where \( t_r \) is an arbitrarily chosen past time instance such that \( r < i \):

\[
\Phi^E_{[t_r, t_l]} = t_l - t_r = ( i - r ) \tau \tag{8}
\]

Hence, for the purposes of dead reckoning, we say an entity \( E \) is temporally consistent (TC) at time \( t_i \) if and only if the measured temporal consistency for the entity during the closed interval \([ t_r, t_l ]\) is less than or equal to a bounded temporal error threshold \( \delta_T \), such that \( t_r^E \) represents the transmission time of the most recent entity state update (ESU) for that entity:

\[
TC^E_{t_i} \Leftrightarrow \Phi^E_{[t_r^E, t_i]} \leq \delta_T \tag{9}
\]

d) Time-Space Consistency

We define a measure of time-space consistency for an entity \( E \) interacting in a DIA as the area under the graph of spatial consistency over time throughout the closed interval \([ t_r, t_l ]\), where \( t_r \) is an arbitrarily chosen past time instance such that \( r < i \):

\[
\Omega^E_{[t_r, t_l]} = \frac{\tau}{2} \sum_{j=r+1}^i \left( \Delta^E_{t_j} + \Delta^E_{t_{j-1}} \right) \tag{10}
\]

Hence, for the purposes of dead reckoning, we say an entity \( E \) is temporally-spatially consistent (TSC) at time \( t_i \) if and only if the measured time-space consistency for the entity during the closed interval \([ t_r^E, t_l^E ]\) is less than or equal to a bounded temporal-spatial error threshold \( \delta_\Omega \), such that \( t_r^E \) represents the time of transmission of the most recent entity state update (ESU) for that entity:

\[
TSC^E_{t_i} \Leftrightarrow \Omega^E_{[t_r^E, t_l^E]} \leq \delta_\Omega \tag{11}
\]

IV DHSM FRAMEWORK

a) Specification

A Dynamic Hybrid Strategy Model (DHSM) \( D \) can be specified as an ordered quadruple of the following form:

\[
D = < M, \, m_0, \, \rho, \, H > \tag{12}
\]
wherein:

- \( M \) is a finite, non-empty set of \( n \) candidate extrapolation models, only one of which is ever active at any one time:
  \[
  M = \{ m_1, m_2, \ldots, m_n \}
  \]  
  \( i = 1, 2, \ldots, n \)

- \( m_0 \in M \) is the default active extrapolation model;

- \( \rho \) is a performance function used to rate the accuracy of state estimation across the set of extrapolation models \( M \), such that:
  \[
  \rho(M) = \{ \rho(m_1), \rho(m_2), \ldots, \rho(m_n) \}
  \]  
  \( i = 1, 2, \ldots, n \)

- \( H \) is a model transition table (function) that maps from \( \mathbb{M} \times \mathbb{N} \rightarrow \mathbb{M} \), and can be represented as a set of ordered triples of the following form:
  \[
  H = \langle m_i, \rho(M), m_j \rangle
  \]  
  \( i = 1, 2, \ldots, n \)

wherein:

- \( m_i \in M \) is the current active extrapolation model at time \( t_{ls} \);

- \( m_j \in M \) is the next active (selected) extrapolation model at time \( t_{ls+1} \);

b) Extrapolation Models

Given the discrete time-base defined previously by Equations (2-5), we can specify the general form for an extrapolation model \( m_i \in M \) as a direct functional mapping from both the current and \( k \) previous states, actions, and environmental (external) influences on the future (estimated) state \( q \) steps ahead for an entity \( E \) interacting in a DIA at time \( t_i \):

\[
\mathbf{s}^{E}_{t_i-q} = f_{m_i}(\mathbf{s}^{E}_{t_i-t_{ls}}, a^{E}_{t_i-t_{ls}}, e^{E}_{t_i-t_{ls}})
\]  
  \( q = 1, 2, \ldots, k \)

wherein:

- \( \mathbf{s}^{E}_{t_i-t_{ls}} \) is the estimated entity state \( q \) steps ahead;

- \( \mathbf{s}^{E}_{t_i-t_{ls}} \) are the current and \( k \) past entity states;

- \( a^{E}_{t_i-t_{ls}} \) are the current and \( k \) past entity actions;

- \( e^{E}_{t_i-t_{ls}} \) are the current and \( k \) past influences.

The functional form of \( f \) defines how entity state is extrapolated remotely, while its inputs define what information must be communicated over the network in the form of entity state updates (ESUs) (due to the fact that remote hosts must have the information to perform the extrapolation). Dead reckoning schemes typically extrapolate over single steps \( q = 1 \), ignore action and environmental variables, and traditionally define state vectors in terms of multi-dimensional location, velocity, and acceleration components. Equation (16) implicitly defines both one-step \( (k = 0) \) and multi-step \( (k > 0) \) extrapolation models [10].

A detailed discussion on how to define the form of the extrapolation function \( f \) is beyond the scope of this paper. The problem may be formulated as one of inference in a temporal probability model, and in this respect has been well studied in the field of artificial intelligence using statistical models such as Dynamic Bayesian Networks (DBNs) and Kalman Filters [26].

c) Performance Function

The role of the performance function \( \rho \) is to assign a rating (or score) for each extrapolation model based on overall predictive performance (i.e. the accuracy of state estimation) between consecutive entity state update (ESU) transmissions. Thus, for an entity \( E \) interacting in a DIA at time \( t_i \), \( \rho \) is defined over the interval \( [t_i^E, t_j] \), where \( t_i^E \) is the time of transmission of the most recent ESU for that entity. Naturally, the obvious choice would be to simply utilize the current consistency metric to rate model performance. In this respect, a temporal consistency metric is independent of entity dynamics, and has no performance function analogue. For spatial consistency metrics, scores can be assigned using the average temporal consistency for an entity \( E \) over the interval \( [t_i^E, t_j] \), given by:

\[
\rho(M) = \frac{\bar{A}^{E}_{t_j-t_i}}{(j-i)} = \frac{1}{(j-i)} \sum_{j=i+1}^{j} A^{E}_{t_j}
\]

For time-space consistency metrics, scores can be assigned using Equation (10). Performance functions and consistency metrics are not mutually exclusive – hence the distinction we have made between the two. For instance, it is entirely reasonable to utilize spatial consistency metrics for ESU generation, alongside a time-space performance function for assigning scores to models.

d) Model Transition Table

The model transition table \( H \) utilizes the performance function \( \rho \) to dynamically select active extrapolation models, as shown in Figure 1. It accepts the current active extrapolation model (denoted as \( m_i \in M \)) and the performance function evaluated over the set of all extrapolation models (denoted as \( \rho(M) \)), and outputs the next active extrapolation model (denoted as \( m_j \in M \)). Technically, it can be called at any time \( t_i \) and for any arbitrary performance function interval \( [t_{ls}, t_{ls+1}] \) to select a new active extrapolation model. In general, however, we only call it whenever a new ESU needs to be transmitted from the local host to remote hosts for an entity \( E \) (i.e. an interval \( [t_i^E, t_j] \), where \( t_i \) is the present time and \( t_j^E \) the time of most recent ESU).
Figure 1: Selecting a new active extrapolation model.

The model transition table $H$ can be viewed as a black-box, whose role is to define the set of rules that govern the dynamic selection of active extrapolation models. In a basic form, it simply chooses the model that exhibits the best score, as determined using the performance function. Issues may arise, however, in situations where multiple models exhibit an identical performance rating. In such cases, the responsibility for resolving the contention and specifying the model precedence also falls to the model transition table $H$, and this must be dealt with by defining any ancillary model selection conditions, parameters, functions or criteria that must be met.

e) Algorithmic Description

The operation of the DHSM, from the perspective of the local (controlling) host for an entity $E$ interacting in a DIA, can be described as follows:

1. $M$ is distributed to remote hosts, alongside a state update to initialise $E$ and inform them of $m_0$, $m_i$ is set as the default active extrapolation model;

2. At each time instance $t_i$ during the simulation, the local host:
   a) Computes the current high-fidelity entity state based on user-input;
   b) Estimates the current low-fidelity entity state for each model $m_i \in M$ in parallel;
   c) Quantifies the prediction error for each model $m_i \in M$ using the given consistency metric(s);
   d) Checks for error threshold violation (i.e. ESU transmission) using the active model only;
   e) Promptly proceeds to Step 3 (see below) if an error threshold violation is detected;

3. If an error threshold violation is detected, an ESU is required to be transmitted to remote hosts, and the local host:
   a) Assigns a score to each model $m_i \in M$ using the performance function, as described above;
   b) Uses the model transition table to select a new active model, as described previously;
   c) Sends the identifier and required state data for the new active model to remote hosts ASAP;
   d) Resets the score and state estimators for each model $m_i \in M$, and proceeds back to Step 2.

For simplicity of description, the algorithm given above assumes a single entity $E$. Handling multiple entities is a trivial task, however – the only real issue is whether to define the components of the DHSM in an entity-centric or host-centric domain. In an entity-centric domain, components are defined on a one-to-one basis for each individual entity, offering greater potential flexibility at the cost of enlarged processing and memory overhead. In a host-centric domain, the components are shared among subordinates who are controlled by the host – so for instance, performance functions and sets of extrapolation models are shared among entities. The choice is application-dependent.

V CONCLUSIONS

In this paper, we have defined a formal specification for the Dynamic Hybrid Strategy Model (DHSM) as a general framework for the reduction of network traffic in DIA. The DHSM expands on the concepts of traditional dead reckoning and incorporates theory relating to the field of Switching Kalman Filters [26], namely the use of multiple state estimators in parallel with one another. By scoring model performance as a measure of consistency and subsequently rating them relative to one another, the DHSM attempts to make an informed decision regarding the suitable selection of remote extrapolation models based on their recent accuracy in estimating future entity state locally. The dynamic model switching feature enables the DHSM to adapt to perceived changes in entity behaviour, all computed on a local-basis by each controlling host.

Ultimately, the utility of the DHSM for providing a further reduction in network traffic over traditional dead reckoning mechanisms depends on the capacity of the set of extrapolation models that comprise it to accurately estimate entity state. The DHSM assumes that recent entity behaviour provides an indication of future entity behaviour on a comparatively long-term basis. Given this limitation, future work will involve the investigation of alternative extrapolation models based on the techniques of user modelling [27].

VI ACKNOWLEDGEMENT

This material is based upon works supported by the Science Foundation of Ireland and Enterprise Ireland under grant no. IRCSET/SC/04/CS0289, and also by Enterprise Ireland under grant no. SC/2002/129.

VII REFERENCES

[1] E. F. Churchill, D. N. Snowdon and A. J. Munro,  
Collaborative Virtual Environments: Digital Places


