Dynamic Player Modelling: A Framework for Player-centred Digital Games.

Darryl Charles[1] and Michaela Black[2]
School of Computing and Information Engineering,
University of Ulster, Northern Ireland.
[1]dk.charles@ulster.ac.uk
[2]mm.black@ulster.ac.uk

KEYWORDS
Adaptive Games, Dynamic Player Modelling, Neural Networks.

ABSTRACT
In this paper we outline a framework for creating player-centred digital games. At the core of our proposal is the requirement for games to be more responsive to different player types and their individual needs; for games to have the capability to adapt so as to provide an appropriate level of challenge for each player, to smooth the learning curve, and enhance the gameplay experience for each player individually. This is not an easy objective to achieve and adaptive game technology, although a popular ideal in some quarters, is still fraught with difficulties and some controversy. We address some of the most well known issues and outline a proposal for dealing with two of the more recent issues: that of monitoring the effectiveness of game adaptation on the basis of player intention and/or frustration, and dealing with dynamic player profiles – because players learn in different ways and at a different speed.

INTRODUCTION
All game players are different; each has a different preference for the pace and style of gameplay within a game, and the range of game playing capabilities between players can vary widely. Even players with a similar level of game playing ability will often find separate aspects of a game to be more difficult to them individually and the techniques that each player focuses on to complete separate challenges can also be very different. For these reasons and others it can be very difficult to design a game that caters for a wide range of player capability and preference. Game developers have traditionally dealt with the range of player abilities in a very straightforward manner, for example, by allowing the player to select a difficulty level at the beginning of the game, as with the classic first person shooter “Doom”. Once a player selects their level of difficulty for a game designed in this way, then there is usually no attempt within the game to monitor how a player is performing in order to adjust the level of challenge or gameplay experience. Recent games are better at allowing a player to set up preferences for their gameplay experience, e.g. as with “DeusEx” where a player can tailor their own avatar’s characteristics, but this relates more to setting up the gameplay experience before starting the game than intelligently recognizing and adapting to the needs of the player in-game. While the concept of an adaptive game is a controversial topic among some gamers and developers, there are clear benefits to tailoring the game experience to particular player types – especially for educational games (Beal et al 2002). Catering for the individual more effectively could help attract a wider participation, if for no other reason that it will be easier for players to get started, progress and complete a game. In a recent edition of the Edge (Edge magazine 2004) Poole provides an insightful discussion on the problem of “beginnings”, teaching the player, and lack of game completions by most players, while in the same issue Redeye highlights the niche quality to current games and their lack of accessibility to a wider group of people.

Adaptivity within games may primarily be implemented by auto dynamic difficulty technologies (Miller, 2004) but there are a number of other ways in which adaptivity can be advantageous. For example, in helping players avoid getting stuck, adapting the gameplay more to the player’s preference/taste, or perhaps detecting deviant player behaviour and modifying the game in response. What we mean by deviant player behaviour is, for example, when a player uses or abuses an oversight in the game design to their advantage. Often this means that the player finds it easier to succeed in the game but their enjoyment of the game is lessened because the challenge that they face is reduced and they are not encouraged to explore the full features of the game – i.e. players will often repeat a successful strategy over and over again because it leads to a predictable win, even if it is boring and somewhat ruins the game. This happens frequently in real-time strategy games such as “Warcraft” or “Command and Conquer”. Bungie, the creators of “Halo 2” – a game much praised for its AI – acknowledged the importance of this when they designed the AI deliberately to prevent the player using “boring” tactics but positively reinforced the player when they used imaginative or adventurous tactics (Griesemer & Butcher, 2002).

In this paper we propose and discuss a novel framework incorporating advanced ideas about player-centred game design. This comprises of four key aspects: player modelling, adaptive game environments in response to player needs, monitoring the effectiveness or appropriateness of any adaptation, and dynamic player remodelling or classification.

PLAYER-CENTRED GAME DESIGN

Most game design is, of course, already centred on the player but it tends to focus on large groups of players rather than catering for individual players – in this paper we hope to persuade the reader that games that are adaptive in catering for the individual will be one of the key innovations in future games. One of the novel aspects of the framework that we propose in this paper for player-centric games is the ability of a game to dynamically model, remodel, or reclassify a player as they play the game. Players differ not only in their characteristics and ability as they begin to play the game, but every player will learn at a different rate and each player will excel in (or just simply enjoy) different aspects of the game.

The most common game model for differentiating between player – and even this is quite rare – is shown in Fig. 1. A player may set their difficulty preference (and perhaps make a few other choices relating to their ability or preference) before beginning the game. Within the gameplay itself there may be a simple hinder/help mechanism, as in the racing game "Mario Kart" where a player who is doing well will not receive good power-ups or weapon bonuses while a player who is struggling will gain a lot of help through a discreet speed up or by receiving more powerful item drops. Most of the simpler methods used – and often most effective – are straightforward help mechanisms, for example in the "Crash Bandicoot" series if a player repeatedly fails at the same point in the game then a mask is provided to the player character which acts as a shield. This essentially allows the player to make one mistake and still be able to progress, e.g. the character may hit a land mine once without losing a life.

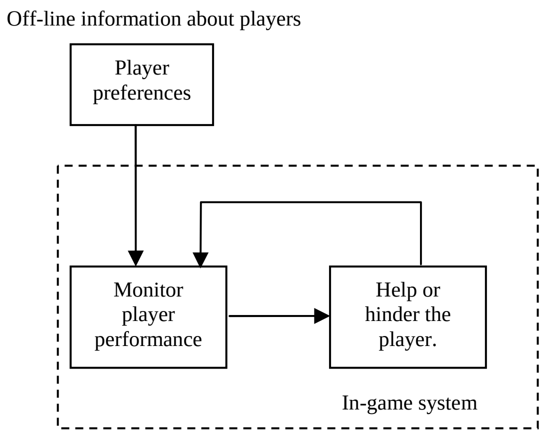

Fig. 1 A typical current game system that changes in response to the player.

Often such systems are "life" based as with "Maximo" where a player is provided with a coin by the angel of death character "Grim" in order to buy another go when they die – when a character fails at a challenge they may go back to a save point within the game level if, and only if, they have a life/death coin. In this way a weaker player still has an opportunity to progress while a stronger player is encouraged to play sensibly – because they have a limited number coins. However, as well designed as this mechanism is, the game can still be prohibitively difficult at times for the novice player. When a player runs out of coins then he/she has to reload a save and restart the level, and this inevitably is one of the reasons that many people will never finish this game. "Prince of Persia: The Sands of Time" provides another mechanism which operates in a similar manner by allowing the player to press a button that "rewinds" a sequence back in time by up to approximately 10 seconds. This is particularly useful for dealing with mistakes, accidents, or misjudgements by a player, for example, let's say a player makes his/her character jump across a gap and the jump is miss-judged so that the character falls to his death. Usually in a game – particularly with game consoles – this would mean that the level would have to be restarted or the player would have to go back to a previous save point, but with "Prince of Persia" a player simply rewinds that mistake and tries it again (up to a limited number of times obviously). This mechanism proves to be excellent in reducing frustration simply by adding a quality game design feature.

A different type of help mechanism used in 3rd person view games is to have a game character look at areas that are interesting as with "Eternal Darkness" where the player character will turn his or her head to look at pictures etc. that perhaps should be examined. "Ico" is even more impressive in this regard in that the non player character "Yorda" who accompanies the main player character will often wander around independently, looking and pointing at things that the player should examine after he/she has been stuck in an area for a while.

These approaches supplement clear, directional level design but are not particularly dynamic. So the basis that we propose for a player-centred framework should build on recent research within the AI community with methods such as intelligent interfaces (Rogers & Iba, 2002, and Livingstone & Charles, 2004). Mainstream AI research is relatively unused within the game development world yet progress in this area for games has the potential to revolutionise gameplay (Charles, 2003) as much as 3D game technology has in the past. AI can provide a perceptual and functional interface between the player and game (Charles & Livingstone, 2004) to enhance the experience for an individual player.

PLAYER MODELLING AND ADAPTIVE GAMES

A few game developers and researchers are now considering player modelling (Houlette, 2004) and adaptive games (Charles, 2003, and Charles & Livingstone, 2004), though work in this area is still relatively rare. Fig 2 illustrates our view of how a basic adaptive game system could be set up. Two sources of information can be used to identify the player-type for a game: firstly the information that a player provides when they begin the game by setting basic preferences and inputting information about themselves. The second source of information should be taken from the player's gameplay habits and performance in-game. Together this information can be used to match the player to pre-defined models and the game can then be adapted to cater specifically to their needs and abilities.

Players may not necessary need to be modelled by one single object but several object models may be used to model them that cover different aspects of the gameplay and their relationship with the player. A more refined object model is obviously better because player modelling can be complex, for example one player may be excellent at combat but terrible at problem solving, while for another the opposite may be true. In this case it is clearly better to model both aspects separately, rather than try to fit them into a coarser single model.

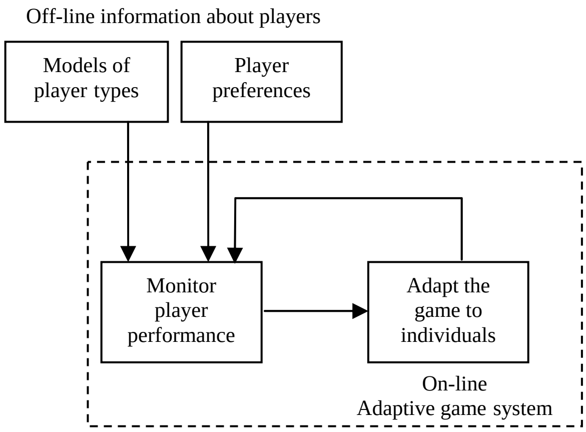

Fig 2. A basic adaptive game system.

Player Modelling

It could be said that there are two main reasons for player modelling in digital games. Firstly, modelling a player in order to instil human-like qualities into a non-player character, as was demonstrated by an example in a recent paper (McGlinchey, 2003) where it was shown that the characteristics of an individual player could be captured while playing a game of "Pong" by a Self Organising Map (SOM) neural network. The SOM could then be used as the "AI" for an artificial computer opponent in subsequent games. The second reason for player modelling and the approach that we are interested in within this paper, is modelling players – or perhaps classifying typical player types or behaviour – so that we may recognise predefined player types or behaviour within the game. The reason that we want to recognise the type of player currently playing is because we wish the game to adapt the needs of the player.

To enable the creation of initial user profiles, some monitoring of game players is required to attain information. Additionally, information about the player – provided by the player themselves – such as whether they are a novice or advanced, male or female, young or old, and other basic general factors may be used as part of the player modelling or clustering process. This information can be used as part of the initial modelling or classification process and it can serve as a starting point for the dynamic modelling process in-game or to help label player groups. For example, we know that there are certain differences, in general, between some of these groups in terms of reaction time and in game play deliberation. Of course, caution must be taken when adopting this approach, because this initial classification process will be quite coarse, e.g. girls may generally like games like "The Sims" and "Everquest" due to the pace of the game and other factors but many prefer action/adventure or sports games. Identification of which type of information produces the most informative profiles is a very important initial task. Key fields of data can be identified as attributes of information, for example gender attribute with values: male, female, and once the necessary attributes have been identified and the information collected, some pre-analysis can done. If predetermined profiles are not obvious we can use unsupervised machine learning techniques such as clustering to partition groups of players. We demonstrate how both may be achieved with neural networks in the next chapter.

Once the most appropriate attributes have been identified we then may produce our separate player profiles where each cluster group represents a different profile of player. If we wish to be able to interpret the properties of these individual groups, they can be labelled and the individual examples of each group provided to a supervised machine learning technique such as a tree induction classifier (Quinlan, 1986). This common inference task consists of making discrete predictions about a concept, in this case each profile, and this prediction problem is referred to as the classification problem. The task of a classification algorithm is to accept a set of training examples which will depict the current state of knowledge for that concept/profile. These training examples are a set of descriptive attributes with an associated class, and this class represents a value for the concept. The algorithm will induce a knowledge structure to distinguish between the values of the concept. A tree induction algorithm will produce a classifier in the form of a tree from which rules can be interpreted as one for each path from the root of the tree to each leaf. These rules depict knowledge which represents the concept. As will be demonstrated in the next chapter we can also use other supervised learning algorithms such as neural networks.

Adaptive games

Adaptation can have two related meanings: one meaning that relates simply to change, and another related to learning and transformation. In the first case the adaptation from one form to another has been predetermined and the adaptive states are known in advance, and in the second case the adaptation occurs after some learning from experience and the transformed stated may be previously unknown. Both forms of adaptation are relevant for games, but adaptation from learning is the most interesting and also the more controversial. The reason for this controversy is that mechanisms within games that have online learning are unpredictable and therefore are very difficult to test thoroughly. Scepticism (or even anger!) is also often expressed by gamers and developers with regard to games that change according to player performance. “Mario Kart” provides one of the most well-known examples; in this game if a player is winning then he/she does not get any of the powerful power-ups and the computer controlled cars often speed up – and the opposite is the case if a player is losing. While this can be annoying if a player is dominating a race, it does even out player ability disparity in a multiplayer competition and thus the race may be more evenly matched and thus exciting. However, if players are aware of “cheating” AI they may alter their gameplay accordingly; i.e. a player may decide to remain in second or third place until near the end of the race so that they may receive a significant power-up or weapon to unleash on the leader on the last corner of the race – thus a new (perhaps unpredicted) gameplay mechanic is introduced. There is some evidence that adaptive game technology is more effective when the player is unaware that it is happening, for example, the primary author of this paper played and completed “Max Payne” without realizing that it incorporated “auto-dynamic difficulty” technology (Miller, 2004).

There are two opposing desires in players that we need to take into account: the desire of a player to learn the rules so as to master the game, and the requirement to avoid “sameness” or lack of variety of gameplay. Thus, while we believe that there is a clear need for player-centric adaptive technology within games to cater for individual players needs, to help them learn and play the game, to enhance their playing experience, to recognise when the player is stuck or frustrated and help out. There is also a requirement that the rules of the game do not change significantly, which would frustrate many players, and ideally either the player should not aware of the adaptive nature of the game or they should have the option to switch it off.

NEURAL NETWORKS FOR THE MODELLING PROCESS

The use of neural networks for the player modelling process is quite an obvious approach but the authors are not aware of them having been used much for this purpose in games yet and so we provide an overview to a few possible supervised and unsupervised approaches below. Neural networks are good at detecting patterns and clustering data (depending on the method) and so we can use a variety of neural network techniques in different ways to identify or understand different players. Additionally, as neural networks are essentially learning machines they hold a number of possibilities with regard to our ideas about adapting to individual players and the dynamic re-modelling of players.

Supervised Approaches

In-game data is very valuable in the process of tailoring a game to the individual player and building accurate player models. For example, we can use reaction times, choices made, styles of play, accuracy of shots/hits, how often a stage needs to be repeated before completing, average health, number of deaths per level, kills per level per possible kills as with “Max Payne” Auto-dynamic Difficulty technology (Miller, 2004). This data may be used directly to decide how to change the parameters of the game environment, attributes of the player character, or non-player character behaviour dynamically through the training of a neural network such as the Backpropagation network. With this approach player entered game data may also be used alongside the in-game player data to moderate the response of the network. This aspect could be important because it may provide a clue to how rapidly or how much the game should be adapted to the player. For example, if an advanced game player is currently playing then they may be less frustrated by not completing a challenge after a few attempts than a novice and therefore the game adaptation may be by a smaller amount or not at all. There are problems with using user-entered profile data (or perhaps any type of profiling), for example, profiling may become frustrating or even redundant if more than one player plays the game at the same time (taking turns) and thus sharing the same profile, in this case it would be impossible for the profiling and adaptation to be accurate. Also, every type of game would require a different approach and the technology may not be appropriate for many types multi-player games because players would be playing against each other on an uneven playing field. For example, in “Soul Calibur II” it is possible for a weaker player to increase their “life bar” relative to their opponent but it actually unusual for this to occur in practice because players like to feel that they are competing on a level playing field.

We can also take another neural network approach to player modelling by using a clustering algorithm. In this way we use the neural networks to cluster player types according to out-of-game and in-game data, grouping player with a similar profile into the same group type. There is a wide range of ways in which this may be done, for example we could use a radial basis network with fixed cluster centres to classify the players, with the centres fixed on different areas of the data space that we believe to provide a good “centre” for our player classification. By monitoring and adapting the player profile throughout the game then the player may achieve a new classification, and thus the game would respond differently. Radial basis networks may also have moving “centres” and so the centres can be moved automatically during training to fit the data more appropriately. It is also possible to retrain the full network during gameplay on the basis of new data, although this is not necessarily an easy thing to do. For example, a single player, depending on the method, may only provide one new data point and so re-training may be futile. This is generally an issue with online learning in games; it is not only slow but often there is not enough new data to significantly impact the training of the network, and needs to be taken into account when choosing which method to use and how to implement it.

Unsupervised Approaches

There is very little digital game research going on that involves unsupervised learning, perhaps because of a lack of expertise in this area. However, we would like to demonstrate here that there are a few very positive and promising uses for unsupervised neural networks for forming a statistical understanding of player data. Unsupervised neural networks are generally used to explore or investigate structure or patterns in data on the basis of statistics or information theory (or similar). It is not known, a priori (though we may have an idea), what the relationship is between the data variables and we would like to investigate this. This is similar to data mining the player data and we can use this approach to help us understand the difference between player styles or capabilities then use this knowledge in our player modelling process. Once the neural network has been trained to our satisfaction then it may be used directly in-game to identify player types or behaviours. Many of these algorithms are also quick to train and so may be more suitable than other approaches for on-line re-training. The techniques that we focus on in the examples below are known as projection methods. With projection methods we typically want to explore the relationship between the input variables but with clustering approaches we treat each data example as a data point (e.g. a player description) and attempt to group data points together based on some similarity measure.

Let us say that we wish to explore the relationship between the variables that we have chosen to uniquely describe a player in a game, e.g. average health, times shot, enemies shot, enemies killed, etc. Then using statistical neural network approaches such as Principal Component Analysis or Factor Analysis we may explore the data so as to identify the correlational (or high order statistical) relationship between the variables. Factor analysis is particularly interesting in this regard because it is frequently used by statisticians in an exploratory mode. A well known example of the use of this method is where the statistical relationship for different forms crime in different cities are explored, e.g. murder, theft, robbery etc. Factor analysis can decipher which input variables have the strongest correlation and the statistician can interpret what this means. It may be found that there is a strong link between robbery and murder and so the output of the network that identifies this relationship may be said to have identified a correlational link which can be explained because these are violent crimes. Similarly, a non-violent crime correlational may be discovered. Using this method to explore player data we may have an advantage in our interpretation of the data because we can also collect information additional about the player that can help us interpret the statistical relationships, e.g. how old are they, sex, what type of games they like to play, how often do they play etc. These values could also be used in the statistical analysis but we would suggest that they may be better served in helping us interpret the correlations discovered by the outputs of the network. Whereas a clustering method would group players together so that we can label these groups as novice, normal or advanced, on the basis of the complete data point. Factor Analysis can identify relationships between sub-sets of the data variables that may be used to identify more refined aspects of player behaviour, e.g. output one could identify the overall capability of the player and output two may identify whether the player is cautious or just dashes in etc. Being able to identify more subtle or complex aspects of player behaviour could be very valuable in tailoring the game experience to the player, and it also potentially opens up new possibilities for dynamic gameplay. For example, if we are able to discover patterns that relate more to player emotion or motivation then this may be used with other sensory devices to discern the needs or desires of the player and the game can be adapted to account for this.

AN ADVANCED FRAMEWORK FOR PLAYER-CENTRED GAMES

Two particular novel technology aspects that we discuss in this paper are monitoring adaptation through sensory equipment and dynamic player modelling and we explore these in more detail within this section. Detection of the need for the game to adapt based, for example, on measuring player frustration (Gilleade & Dix, 2004) is one approach for game adaptation but we propose a slightly different model, one in which the game is adapted on the basis of detecting player type coupled with game performance. The effectiveness of adaptation can then be measured by a reduction in the level of frustration and other measures. If adaptation does not improve player performance or their frustration levels then perhaps this is because the player has been classified incorrectly, or more likely as they have progressed through the game the model that fitted the player initially is no longer applicable. Therefore in this scenario it may make sense to reclassify or dynamically remodel the player – Fig 3 illustrates how this advanced framework may be executed.

Measuring the Effectiveness of Adaptation

We need to know when to adapt the game to a player (Gilleade & Dix, 2004) but also we should monitor if our adaptation has been effective or appropriate. If we make a change based on the game data coupled with the player profile and this frustrates, or hinders the player more (or vice versa) then we may make one of two conclusions: our adaptation is inappropriate or our model of the player is inaccurate. In either case this is a good reason to have the feedback loop in our model illustrated in Fig. 3.

Assuming that there are discrete changes to the adaptation of the game and that these have tested these thoroughly before game release, we then can focus on making sure that we classify the player correctly so that the state of the game is appropriate to them. This is especially important because players learn at different rates and so we need to take account of concept drift (Black & Hickey, 1999) in the classification process (see next section).

The manner in which we measure the requirement or appropriateness of adaptation may be most effectively achieved using affective computing techniques by monitoring a player's emotional state through input devices, coupled with in-game data. It seems clear by initial research that attempts to detect a player's emotion through input devices that it is not very straightforward. For example, the

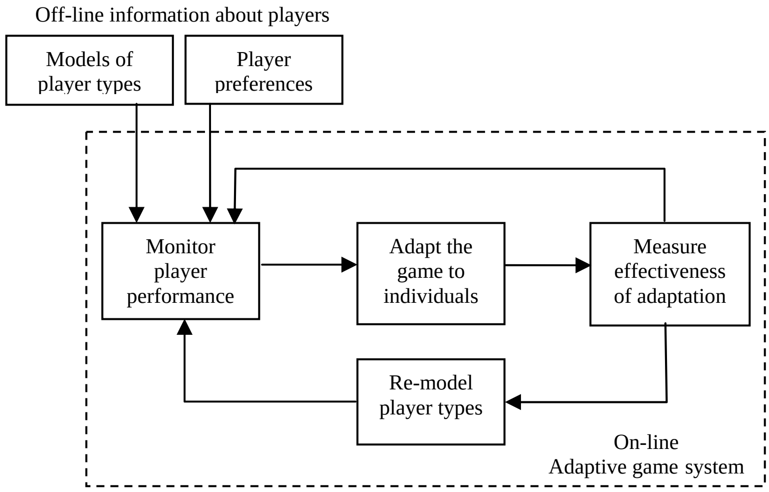

Fig. 3 Adaptive game system diagram illustrating the three phases that that takes account of errors in adaptation.

emotional affect detected in the player through a gamepad analog button (Sykes & Brown, 2003) will vary with each player, game type and even perhaps when it's played. The information may be corrupted by interaction stress in, for example, playing an action game by altering physiological factors that would normally infer emotion, such as skin conductance (when using an appropriate sensor). Facial expressions or body movement may be used to infer the emotional state of the player – whether they are happy, content or frustrated – and game cameras such as the PS2 EyeToy is becoming more popular and widespreadly incorporated into games. The difficulty with using a camera for facial expression though is that, to some degree, there is an expectation that the player will roughly maintain their position relative to the camera (Gilleade & Dix, 2004) – this is particularly an issue with game consoles.

With simple modification of existing input devices temperature or pulse (i.e. heart rate) sensors may be added like those on a typical exercise bicycle. These would not be expensive to implement but could potentially revolutionize game design with respect to a games' responsiveness to an individual players' needs. Even in casual way this information may introduce interesting new directions in gameplay – if you imagine a game from the horror genre such as "Silent Hill" or "Resident Evil". In this example the game could wait until a player seems at their more relaxed before landing that shocking surprise on him or her. Normally, games of this type must craft the levels and script events very cleverly to achieve the same effect, and it is very difficult to perfect. It will probably prove to be the case that one method alone will not be enough to accurately gauge a player's mood. That a mix information from standard sources such as the mouse or joy pad, along with more advanced sources of information provided by cameras or other sensory devices along with the player's profile, will be necessary to make decisions that tailor the game to individual players on the basis of their emotional state. Statistical methods such as neural networks will then be necessary to decipher the structural relationships in the data.

Dynamic Player Modelling and Reclassification

The idea that a player's model needs to be adapted has been recognised recently (Houlette, 2004) but this is still a very new area for digital game research. On a basic level a player model may be thought of as a statistical representation of the player based on the frequency of repeated actions or average values of the parameters of their player character etc. It should be obvious then that an individual player's profile is likely to change throughout the progress of a game. This can be for all sorts of reasons, for example they are learning the action aspects of game more quickly than adventure aspects or perhaps they have reached a new gameplay dynamic in the game that they can't quite get to grips with – all players will be different so these things are very difficult to predict.

Because of the nature of game playing there will be new examples available about the player's profile as they play the game, hence the requirement for on-line learning. These on-line learning systems will receive examples on a continual basis and are required to induce and maintain a basis for classification and thus may have to deal with concept drift (Black & Hickey, 1999). Game players will adapt their strategy to survive or win as the game adapts to suit their profile. This change in the player behaviour, as discussed previously, may be part of their learning process: i.e. they get better at the game over time, or a may be forced into a strategic change of tactics. This adaptation, known as concept drift, can therefore be an immediate change in tactic or a slow progression to another. By concept drift we mean that some, or all, of the basis for defining a profile is changing as a function of time.

Typically there are a number of sub-tasks involved in the handling of drift within incremental classification learning. In increasing order of difficulty these are:

1. Identifying that drift is occurring;
2. Updating classification in the light of drift;
3. Tracking and modeling/analysing the pattern of drift over a period of time.

Machine learning techniques have been used with this form of user profiling/modelling in other domains such as cellular fraud in telecommunications (Fawcett & Provost, 1999). The aim was to analyse calling behaviour and detect anomalies. It also highlighted that patterns of fraud are dynamic; bandits constantly change their strategies to avoid detection. This links very well into game players having profiles which change/evolve through the life of a game. Game players may be thought of as behaving like the fraudsters; they adapt and change their strategies as a mechanism to win/survive and so profiles can be monitored and adapted using existing machine learning techniques. For example, recent work (Black & Hickey, 1999) has demonstrated that profiles may be induced from telecommunication customers in relation to using a product, and that changes may be detected in the customers who are currently using the product.

As already indicated, player's profiles may change in many ways. We can break these down into two aspects of change: a progressive move – referred to as evolutionary adaptation, or immediate change – referred to as revolutionary adaptation (Black & Hickey, 1999). This work also introduces a methodology called TSAR (Time Stamp Attribute Relevance) which has been used successively to adapt to concept drift in telecommunication customer data (Black & Hickey, 2002). This methodology can be applied to neural network approaches, as discussed earlier, so as to deal with concept drift in online learning within digital games.

CONCLUSIONS

Modern digital games are extraordinarily good at many things but even the best examples of these games are still not very capable at monitoring players, distinguishing between different player groups and altering the game state to meet individual players' needs. In this paper we described a framework for dealing with this issue and providing more adaptable games, and in particular approaches for dealing with two particularly current issues: that of monitoring the effectiveness of adaptation through affective and statistical computing approaches, and the dynamic remodelling of players based on ideas from concept drift. We proposed several neural network approaches as part of the realisation of this framework and in future work intend to test these ideas further. The improvements that may come from positive developments in this area could be as straightforward as helping the player in learning how to play the game, through to encouraging gameplay innovation in digital games.

References

Beal C, Beck J, Westbrook D, Atkin M & Cohen P, "Intelligent Modeling of the User in Interactive Entertainment", pp 8 – 12, AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, Standford, 2002.

Black, MM, Hickey RJ, "Maintaining the Performance of a Learned Classifier under Concept Drift", Intelligent Data Analysis 3 (1999) 453-474.

Black MM & Hickey RJ, "Classification of Customer Call Data in the Presence of Concept Drift and Noise" Soft-Ware 2002: Computing in an Imperfect World, Belfast, 8-10 April 2002, in "LNCS 2311", Springer-Verlag (Berlin), ISBN 3-540-43481-X, Pages 74-87.

Charles D, "Enhancing Gameplay: Challenges for Artificial Intelligence in Digital Games", 1st World Conference on Digital Games 2003, University of Utrecht, The Netherlands, 4-6th November 2003.

Charles D, Livingstone D, "AI: the Missing Link in Digital Game Interface Design?", 3rd International Conference on Entertainment Computing, September 1st-3rd, 2004, Eindhoven, The Netherlands.
Edge Magazine (UK) #139, Trigger Happy and Redeye articles, August 2004.

Fawcett T. & Provost F, "Activity Monitoring: Noticing Interesting Changes in Behavior", 5th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining, San Diego, pp 53 – 62, 1999.

Gilleade K. & Dix A, "Using Frustration in the Design of Adaptive Videogames", Proceedings of ACE 2004, Advances in Computer Entertainment Technology, ACM Press, 3-5 June, 2004.

Griesemer J. & Butcher C, "The Illusion of Intelligence", Game Developer Conference, San Jose, 2002, slides available from (last accessed 11/07/2004),
http://halo.bungie.org/misc/gdc.2002.haloai/talk.html

Houlette R, "Player Modelling for Adaptive Games", pp 557-566, AI Game Programming Wisdom II, Charles River Media, 2004.

Livingstone D. & Charles D, "Intelligent Interfaces for Digital Games", AAAI-04 Workshop on Challenges in Game AI, 25-26th July 2004.

McGlinchey S, "Learning of AI Players from Game Observation Data", GAME-ON 2003, 4th International Conference on Intelligent Games and Simulation, pp. 106-110, Nov. 2003.

Miller S, "Auto-dynamic Difficulty", website forum debate, last accessed 11/07/2004,
http://dukenukem.typepad.com/game_matters/2004/01/autoadjustin g_g.html

Quinlan, J.R., Induction of Decision Trees, Machine Learning. 1(1), 81-106, 1986.

Rogers S. & Iba W, "Adaptive User Interfaces", 2000 AAAI Spring Symposium, Technical Report SS-00-01.

Sykes J. & Brown S, "Affective Gaming: Measuring Emotion through the Gamepad", Human Factors in Computing, pp 732-733 CHI 2003.