Proceedings of the Symposium

AI and Games Symposium

A symposium at the AISB 2009 Convention (6-9 April 2009)
Heriot-Watt University, Edinburgh, Scotland

Symposium Chairs
Daniela Romano
David Moffat

Published by SSAISB:
The Society for the Study of Artificial Intelligence
and the Simulation of Behaviour
http://www.aisb.org.uk/

ISBN - 1902956818

AI and Games Symposium

A two-day symposium at AISB 2009 (6-9 April 2009).
http://www.dcs.shef.ac.uk/~daniela/AISB09/

PROGRAMME CHAIRS

Dr Daniela Romano, University of Sheffield, UK
Dr David Moffat, Glasgow Caledonian University, UK

INTRODUCTION

Computer games now form an important sector of the computing and entertainment industries, and they are very sophisticated in many ways. The need for better AI in games is deeply felt, however, and recognised by the industry. Conversely, games offer new challenges and excellent application domains for AI technology and research. They are increasingly used for education, serious games or game-based learning, where story and AI techniques create a believable, engaging experience for learners.

This symposium focuses on the application of artificial intelligence or intelligent techniques, frameworks and theories to create interactive, engaging, intelligent games.

It includes papers that review the areas of games and AI, and taxonomies of the ways they might relate. Games are seen by some authors as sandbox environments for the development of multi-agent systems. A wide range of subjects are encountered in the papers that follow, from automatic generation of football commentary, through models of emotions in artificial game characters, to the application of neural networks to their decision making.

Not only artificial characters, but the games too may be intelligent, including new kinds of interfaces to make games more realistic, by using large touch-sensitive screens, for example. There is also cutting-edge work presented on using direct BCI (brain-computer interfaces) for games.

Neither are games solely for entertainment purposes. Serious games are also presented, whether for teaching ethics or sustainable development

As well as the papers and posters in these proceedings, there are demonstrations of some of the papers, and of other work in AI & Games. There is also a tutorial session kindly offered by Robin Baumgarten, who will show how to use an AI interface for the real-time strategy game DEFCON (from Introversion Software).

We are also pleased to open the Symposium with an invited talk in the plenary session by Simon Lucas, from the University of Essex. Simon has been very active in the field of AI & Games, and is a co-founder of the "Artificial Intelligence and Games Research Network," which supports this Symposium.

We thank the organisers of the AISB convention for all their hard work with the venue and proceedings, the AI and Games Network for their support and we would also like to thank the members of the Programme committee.

TOPICS

Topics of interest include but are not limited to:

Intelligent Games

- How intelligent games are designed and engineered
- What artificial intelligence brings to games and vice versa
- Application of AI planning, machine learning etc. to games
- Path-planning, crowd simulations, flocking, steering behaviours
- Real-time performance of AI techniques in games
- Intelligent, (semi-)automatic content-generation
- Crucial elements of a believable intelligent games
- How to make AI characters more believable, intelligent and adaptable
- Group behaviours of AI characters
- Crucial elements of believable intelligence in games
- Psychological aspects of intelligent games
- Cognitive, social, and emotional impact of intelligent games

Technology for Intelligent games

- Combining games and technology to produce engagingand immersive
- Emergingtechnologies(e.g. virtual environments and content sharing, intelligent agents, human, robotics, wearable computing, artificial vision, mixed reality, and advances in understandiof human emotion and gesture) and their application to intelligent games

Users and Intelligent Games

- The interaction between players and the game
- Understanding the player's state, in playful and serious games
- Adapting to players, to improve gameplay

PROGRAMME COMMITTEE

Abdennour El Rhalibi, Liverpool John Moores University.
Alex Narayek, National University of Singapore, Singapore
Ana Pavia, INESC-ID/Instituto Superior Tιcnico, Portugal.
Daniel Ballin, BT Group, UK
Daniel Livingstone, School of Computing, University of Paisley.
Doron Friedman, Interdisciplinary Center (IDC), Herzliya , Israel
Gianna Cassidy, Glasgow Caledonian University, UK

Ian Millington, author of "Artificial Intelligence for Games", Morgan Kaufmann Publishers, 2006, and founder of Mindlathe, an AI middleware developer for games
Ian Wright, AiLive.net, UK
Irene Mazzotta, University of Bari, Italy
Isabel Machado Alexandre, DCTI - ISCTE and INEC-ID, Portugal
Jeff Orkin, Media Lab, MIT, US
Jon Sykes, Glasgow Caledonian University, UK
Judith Good, Director, University of Sussex
Judy Robertson, Heriot-Watt University, Edinburgh, UK
Lisa Gjedde, Aarhus University, Tuborgvej 164, Copenhagen NV, Denmark
Marc Cavazza, Teesside University, UK
Marco Gillies, Goldsmiths, University of London, UK
Mark Overmars, Utrecht University, Netherlands
Mervyn Levin, Asia representative Serious Games Institute (SGI) UK
Patrvcia Restelli Tedesco, Universitade Federal de Pernambuco (UFPE), Brasil
Paul Richmond, University of Sheffield, UK
Peter Cowling, University of Bradford, UK
Pieter Spronck, University of Maastricht, Netherlands
Ruth Aylett, Professor, Heriot-Watt University, Edinburgh, UK
Sandy Louchart, Heriot-Watt University, Edinburgh, UK
Simon Colton, IC, London, UK
Simon Lucas, University of Essex, UK
Stephen McGlinchey, University of the West of Scotland, UK
Tom Welsh, Researcher, Glasgow Caledonian University, UK
Vinoba Vinayagamoorthy, British Broadcasting Corporation (BBC), UK

Table of Contents

A Taxonomy of Video Games and AI

E. A. A. Gunn and **B. G. W. Craenen** and **E. Hart**[1]

Abstract. Game developers have always striven to create better and more complex games. The use of artificial intelligence (AI) in games is increasing in game development to this end. However there appears to be a mismatch of implications and expectations between game developers and AI researchers of what AI can bring to games, mostly due to differences of interpretation of the problems involved. To overcome this we present a taxonomy to aid knowledge transfer between the two groups, including elements from environments, game theory, and information theory. We also identify other concepts of games and AI that could be important to both sides.

1 INTRODUCTION

Game developers have always striven to create better and more complex games. Despite advances in technology for rendering engines, physics engines, dynamic lighting and so on to this end, one area that is currently lacking is that of believable interactions with non-player-characters (NPCs) within games. Currently these NPC interactions can be described as static and deterministic, without the player getting a sense of character about a particular NPC, or indeed the game itself. Static and deterministic game-play means that game-play can degenerate into reacting to the behaviour of the NPCs rather than navigating through the game or its story line. This often leads to a "shallow and unfulfilling" game experience [10].

Increasingly, game developers are looking toward artificial intelligence (AI) to improve NPC behaviour. However, Fairclough et al [10] states there is little implementation of AI in computer games and identified a number of reasons why this might be the case: a lack of computer resources; suspicion by game developers as to the non-determinism of AI methods; a lack of development time; and a lack of understanding of the scope of AI.

On the other hand, academic research into AI within games is increasing to allow more dynamic, and more realistic games. However existing work is primarily focussed on the specific implementation of AI methodologies in specific problem areas. For example, the use of neuro-evolution to train behaviour in *NERO* [7]. With greater analysis of the problems faced in implementing AI methods in computer games, more accurate and efficient methodologies can be developed to create more realistic behaviour of artificial characters within games.

Overall, there seems to be a miscommunication between game-developers on the one hand and AI researchers on the other. The game-developers perhaps lack precise knowledge about AI methodologies and are consequently wary of implementing them into their games. AI researchers for their part perhaps lack a broader awareness of the requirements of games and where various AI methodologies could provide a solution.

[1] Napier University, Edinburgh, email: e.gunn@napier.ac.uk

We therefore suggest this taxonomy to facilitate communication between game developers on the one hand and AI researchers on the other, thus bridging the knowledge gap between the two. As far as we are aware, there is currently no such taxonomy, however we have found previous general classifications that provided useful background information for the taxonomy, and confirmed some of our observations. Konzack [16] proposes a critical analysis methodology for games, and goes through a specific example for his methodology. However, the areas classified are high-level and the discussion too general for direct use within our taxonomy. Aarseth [1] specifically points out the disparity between the classifications proposed by Konzack, and looks at the classification problem from an aesthetic point of view, more suitable to a social science methodology. Continuing in the theme of general classifications of games, and unlike our focus on using AI in games, Lindley [18] proposes a taxonomy of video games in general.

Our taxonomy represents a more technical view on the involvement of AI in computer games. We believe that further research to create an overall ontology to allow free discussion using a common language will help cement any interactions between game developers and AI researchers. Although we go some way in doing this by using existing concepts with complete definitions, a broader language would facilitate communication even further.

Our taxonomy (illustrated in figure 1 for reference throughout the paper), provides a bridge between game developers and AI. It can, however, also serve another purpose: Laird and van Lent [17] discuss the development of game AI as an aid to one of the original purposes of AI, that of creating human-level intelligence in an artificial system. Through use of this taxonomy, methods developed as an aid to game intelligence can be used in other spheres of research within AI. With better understanding of *why* certain AIs work in certain game worlds and types, we can thus specify *how* specific concepts within games can be built upon and generalised outside of games. Kleiner [15] reinforces this idea and applies a more methodological view of the situation, concluding that the vast sums of money involved in the game industry can benefit and accelerate general AI research. A recent article in *IEEE Spectrum* [25] discusses the possibilities.

Furthermore, AI researchers can analyse their AI methodologies, match them to the theoretic concepts in our taxonomy, and thus identify game types that would describe how they could create their own game, or use an existing game, as a core laboratory for their research.

This paper begins with a discussion on game genres in section 1.1 as initial background; analyses the environmental and player elements of game types in section 2; applies game theoretic concepts to the game type elements extracted in section 3; and finally maps AI methodologies to the game theoretic concepts in section 4. We follow with a number of useage scenarios to illustrate how our taxonomy can be used in section 5 before providing a conclusion in section 6.

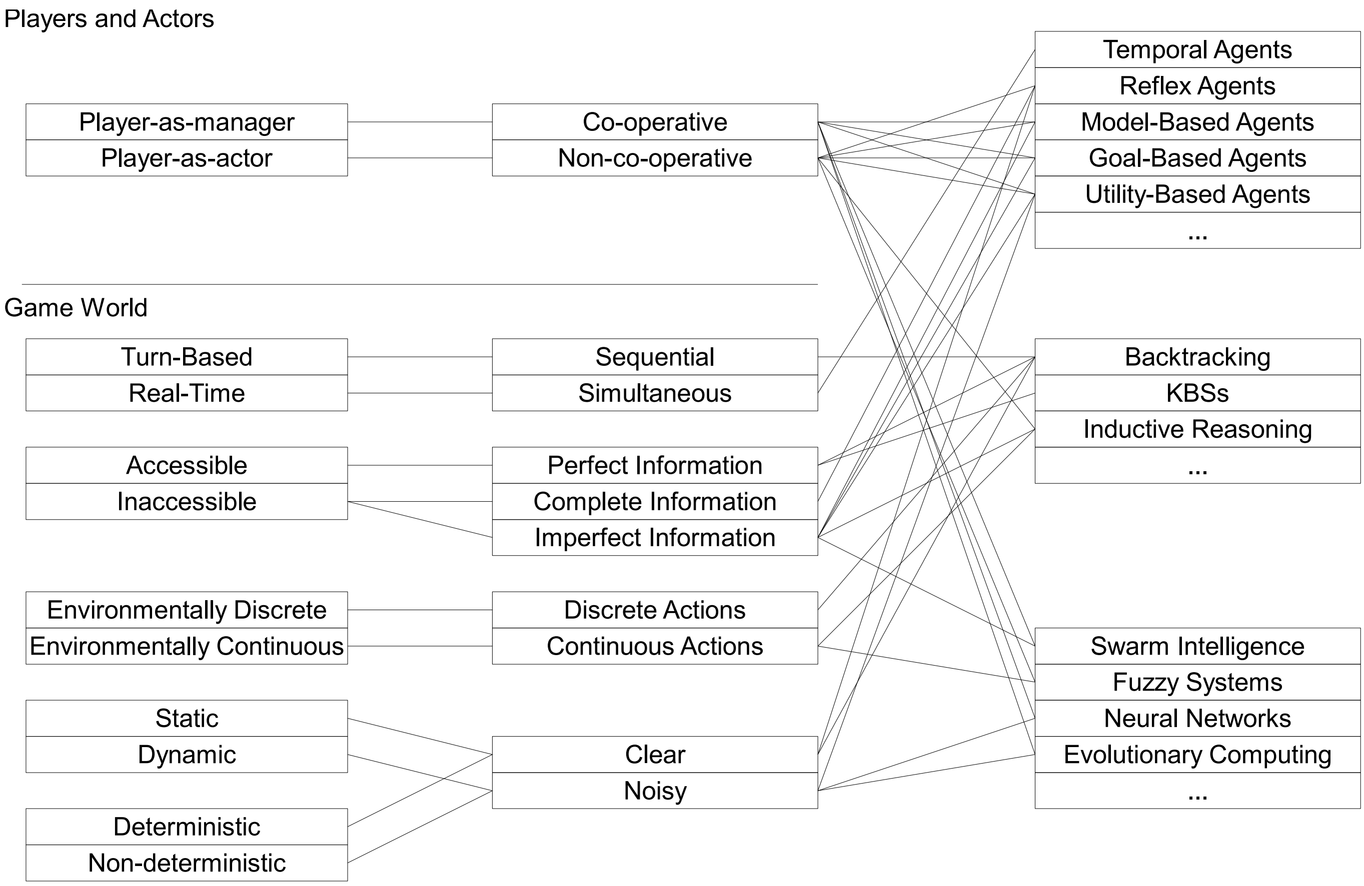


Figure 1. The complete taxonomy

1.1 Game Genres

The most logical place to start with a taxonomy of a subject is to use already existing classification schemes: within games, that would be the genres that are currently used. A comprehensive study and classification of genres has already been conducted [27]. However, Apperley [2] makes the case that current genres are merely representational: existing genres are broken down in part due to their visual aesthetics, and in part by the subject and content of the games in consideration. What is missing is any form of delineation based on interactivity.

He goes on to say that the primary problem with current genres of computer games is what is described by Bolter and Grusin [3] as a "logic of remediation", the "formal logic by which new media refashion prior media forms": placing games into categories that already exist for media types such as novels and films. Frasca [11] creates this delineation by referring to two aspects of categorisation of games: *narratological* categorisation, based on the subject, content and representation of the game (current genre analysis); and *ludological* categorisation, which is concerned with the rules, mechanisms, rewards and structure of games.

This taxonomy is designed for game developers and AI academics, both of which are more interested in the ludological categorisation of games. We therefore rejected the use of the narratological categorisation within the taxonomy. Instead we focus on the components of the games and delineate them using the ludological categorisation. We provide a short excerpt of the narratological categorisation here purely as background for the taxonomy.

1.1.1 Action

Action games are fast-paced, requiring quick judgement and snap decisions. These primarily involve interaction with computer-controlled game actors who help or hinder the player through the course of the game. The player usually controls just one character. Generally, the goals within these games are simple, however the challenge is in accomplishing these goals by navigating through a level consisting of hostile enemy NPCs.

1.1.2 Adventure

Games where there is a rigid structure to the game - for example sequence based movement throughout the game, where the player is presented with only a picture of the surroundings - fall under the category of adventure games. These games generally have interesting, long and complex story-lines, and the ultimate goal of the game is discovered through the course of the story.

1.1.3 Role-playing

Modern computer role-playing games (CRPGs) allow the player to take on the "role" of a character, directly controlling that character in the game world. One of the key elements of RPGs is the idea of character advancement, where throughout the course of the game the activities of the character are rewarded by allowing the player to advance their character to become better at certain activities. The goals of CRPGs vary wildly depending on the setting, history and type of character the player plays. Most modern RPGs also provide an environment for enthusiastic players to create their own settings. This module system is being adopted by most game genres, and can provide an ideal opportunity for testing game intelligences.

1.1.4 Vehicle Simulation

This genre is subtly different to the preceding three - instead of controlling a character, the player controls a vehicle. This vehicle exist within a game world that includes abstractions of real-world physics to ensure that the player experiences as close to a “real” experience as possible. Vehicle simulations can be purely simulation based, or may have goals that need to be achieved depending on the type of vehicle available.

1.1.5 Strategy

Strategy games take the player from a character-based perspective to a more high-level perspective. In almost all cases, escalation of force and the creation of units is tied directly into resources which are collected by specialist units or structures. These resources are then used in the creation of other units and structures. The goal in most of these games is to complete an escalating campaign of several levels, the conclusion of which is the completion of the game. Individual maps can have differing objectives but the key component is the strategic distribution and use of units with varying abilities under the player’s control.

1.1.6 Management

Like strategy games, management games (also called “god” games) involve a higher-level view of the game world, and involve resources with which the player can alter the game world, or change it in some way for the population of actors within the game world. Some of the games within this genre have no specific goals in mind - they are primarily simulation games - but others offer specific objectives.

1.1.7 4X

4X is a term used to represent games that require *eXploring, eXpanding, eXploiting, and eXtermination*, and the key difference between this and strategy games (of which 4X is sometimes classed as a subgenre) is the level of abstraction of the player: in strategy games the player is usually cast in the light of a theatre commander, sent there by their superiors, whereas in 4X games, the player is playing the role of a world leader. Some 4X games allow the player to also participate as a theatre commander in individual combat scenarios.

1.1.8 Life Simulation

Life simulation games are a relatively small genre of games, but perhaps of most interest to artificial intelligence researchers. Most AI researchers are familiar with *The Game of Life*, and life simulation games are much in the same vein. There is little in the way of goals or objectives, the challenge is to create or adapt a game world or configuration of actor to allow the successful survival of actors within the game world. Broadly speaking the simulations can be broken down into biologically inspired simulations and socially inspired simulations. Specific mention should be made of the large market for console games within Japan where a large component of the game is the normal day-to-day running of a life, and the popular “dating game” market, both of which involve a high level of social interaction with NPCs.

1.1.9 Puzzle

Puzzle games are generally small games where there are no other actors in the game - the game is completed by using logic and deduction to complete the goals.

1.2 Definitions

Throughout the paper the following definitions are used to differentiate between a number of key concepts that do occur together in the taxonomy.

- Game World
  A *game world* is a definition of the environment that the game takes place in, consisting of the rules and rewards of the game.
- Actor
  We shall define an actor as an entity that exists within a game world and that can interact with the game world, and/or other actors, and are under the control of one or more players as the main method of playing the game. These players can be human or artificial. However, this does not preclude the use of intelligence, as in many games the actors are semi-autonomous.
- Player
  We shall define a player as a controlling intelligence, either human or artificial, that plays the game. Where specific differentiation is needed between human player and artificial players, this has been explicitly mentioned.
- Participants
  We shall group actors and players together where necessary and call them *participants* within a game. This allows us to generalise concepts over multiple tiers of intelligence that may present themselves within games.

2 GAME TYPES

Genres mainly group the content of the games and the style of play, and have been mentioned as a background and to provide some observations about various types of games. Key factors in a game that affect our taxonomy have been extracted, and this provides the first layer of our taxonomy.

It is clear that there are two distinct areas that need to be considered in any game - that of the players and how they play the game; and the game itself - its rules and rewards (what we call the game world). This is a segregation that we will use throughout this and the next layer.

2.1 Players and Actors

Analysing how the players interact with a game is vital, as any artificial player designed to compete with a human player must be able to do so within the same rule-set as the human players themselves.

There is an interaction with and control of actors in the game world by a player, which can range from no actors to any number of actors (a one-to-zero-or-many link, see figure 2). There are two special cases that need to be considered: that of no actors, and that of one actor. The first case describes games where the player is competing against the world itself - manipulating it directly “from on high”, such as in puzzle games and some social simulation games. We shall refer to these as *player-versus-environment* games. Player-versus-environment games generally have no intelligence as there is no artificial player to compete against (for example in puzzle games).

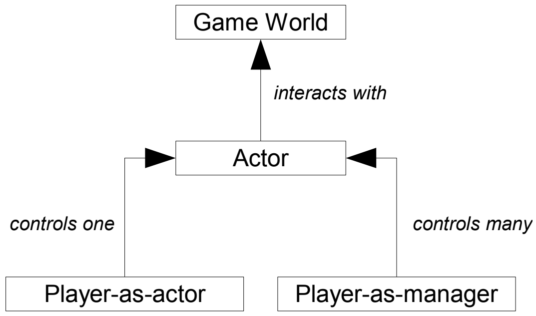


Figure 2. The player-actor-game-world interactions. Note that the player does not directly interact with the game world.

As such it is mentioned here for completeness. This does not affect the taxonomy.

The second special case is where one actor is considered the proxy for the player in the game world, and only through controlling and manipulating that character directly can the player achieve the objectives of the game. We shall call this *player-as-actor* interaction.

For numbers of actors greater than one, the game usually becomes more of a management game, where the player uses all actors under their control to complete the objectives of the game, and as such we shall refer to them as *player-as-manager* games.

2.2 Game World

We can use concepts from agent-based systems to help us categorise various types of game world by using the metaphor *a game world is an environment*. Russell and Norvig [24] provide us with a summary of environment types, which we can adapt for our purposes in the taxonomy. We need to view the environments, and thus the game worlds, from the actor level, no-matter how that actor is controlled. This is because of the way the player interacts with the game world: via an actor or actors. This is enables us to create an accurate taxonomy, as it is the actors that navigate and exist within the game world. We present these concepts in the following subsections.

2.2.1 Accessible vs. Inaccessible

The accessibility of a game world relates to the information of the game world that is available to each actor within the game world. If an actor has knowledge of every aspect of the game world and knows of everything that is going on within that game world, then the game world is *accessible* to that actor. If, however, there are limits to what an actor may know about the game world (for example using the concept of fog-of-war), then the game world is *inaccessible*.

2.2.2 Environmentally Discrete vs. Environmentally Continuous

Actors within the game world may make a number of possible actions at any point, determined by the range of potential actions within a game world. If there is a finite set of actions that an actor can take (for example only being able to move one square in any one of the cardinal directions on a grid), then the game world is *environmentally discrete*. Where there is a continuum of possible actions, such as allowing an actor to turn to any direction, then the game world is *environmentally continuous*.

2.2.3 Static vs. Dynamic

Artificial participants need time to consider their moves just as human players do - albeit they make conclusions significantly faster than any human player could achieve. If the game world alters whilst an participant is “thinking”, then the game world is *dynamic*. If, however, the game world remains the same until an participant has made a move, then the game world is *static*. A third variety is *semi-dynamic*, where although the current state of the game might not alter whilst a participant is deliberating, the measure of performance of the participant reduces until a move is made. The example given to us by Russel and Norvig is that of time-measured chess games, and there are video games where a similar technique is used.

2.2.4 Deterministic vs. Non-deterministic

A game world is *deterministic* if the next state can be explicitly concluded from the present state of the game world and the actions carried out by the actors. If there is an element of uncertainty, or if the game world changes despite actions by the actors, then the game world is *non-deterministic*. There is a clear distinction in the case of an inaccessible game world, such that the deterministic nature of the game world should best be considered from the actor’s perspective. Actions that are carried out without the actors knowledge may influence what that actor sees in the next state. So, despite the fact that the overall game world state is *deterministic*, from the actor’s perspective, the game world is *non-deterministic*. Random elements within a game world also create a *non-deterministic* world.

A further definition of environment is provided by Russel and Norvig, that of episodic vs. non-episodic. If an actor can take an action, the results of which have no relation on future actions, the environment is episodic. If however the consequence of one action relates directly or indirectly to the available information or set of actions at a future point, the environment will be considered non-episodic. Such a definition is too strict to use in this taxonomy, as all actions within the course of a game have consequence.

2.2.5 Turn-Based vs. Real-time

Turn-based games place the players in a game-playing sequence. Whilst this type of game could be, theoretically, applied to any game, there are only a small number of genres where this mechanic is used, primarily 4X, strategy, some role-playing, and some life-simulation games. These games can require a great deal of strategic thinking, and as such having the time to analyse a situation and make decisions based on that is almost necessary.

A number of games also use *semi-turn-based* mechanics, where the player has the opportunity to pause the game to make decisions or queue up actions, and then return to normal *real-time* playing afterwards; or where certain sections of the game are turn-based, and the rest is real-time.

Non-turn-based games are called *real-time* games.

2.3 Other Game Type Considerations

The concepts identified provide clear definitions to allow us to categorise games for the taxonomy. However, there are other important concepts that relate to all games. We have identified the following.

2.3.1 Layered game-play

Not all games stick strictly to one method of game-play throughout the entire game. For example a turn-based mechanic during combat, and a fast-travel mechanic that gives an overview of the entire world, in which the player can move anywhere - despite the normal game-play being on a hexagonally divided game board in real-time. These parts of the game are separate, but connected. It is therefore necessary to identify each discrete game-play types within a layered game, and apply the taxonomy to each area to better apply research methods to each area as necessary.

2.3.2 Artificial Participants

Generally, games do not pitch the human player against a multitude of completely individual and separate actors. Usually there is at least one faction that the player is playing against. The impression is that all the artificial actors that the human player faces are given directives from the faction, and as such work together. This then creates the situation where we have *human-player-as-actor* against *artificial-player-as-manager*. Such cases need to be taken into consideration to allow the correct methodologies to be used in the correct places.

2.3.3 Hierarchical Intelligence

Player-as-manager games provide us with a potential hierarchy of intelligences that would be required: the artificial player and the artificial player's actors. Different AI methods would be required in this case, as the artificial player would require high-level strategic decision making. On the other hand, individual actors might only require reflexive behaviour (*i.e.* "I am being shot, I shall move away.") Currently in these types of games (especially strategy games) there is little intelligence at the artificial player level, merely consisting of such static tactics as "build up a force of x units, and send them along y path". Observation of such tactics in has shown that there is a reliance on some form of state analysis. By considering the hierarchical nature of the player and the actors under that player's control, suitable mechanisms can be introduced. First to provide adequate high-level strategic planning for the artificial player. Secondly to provide low-level tactical planning for the artificial player's actors.

2.3.4 Single Player vs. Multi-player

Single player games were the original type of video games - it was one human player at a time playing the game - and the "single" aspect refers to that single human player (as opposed to a single player overall, including artificial players). This is the key type of game that still exists today where good NPC AI benefits the immersion into the game of the player, and subsequently their enjoyment of that game. These games rely on involving, long story-lines, usually with a large number of optional sub-quests to lengthen the time of play. This is the type of game where unintelligent NPCs are most noticeable.

As with single-player games, multi-player games refer to the amount of *human* players in the game. Despite the fact that these games are primarily designed to be played with other human competitors, almost all allow the use of *bots* for off-line playing, where the human player combats artificial players. These bots are artificial actors; and intelligent, human-level AI in these bots would allow offline players to still hone their skills when human opponents are not available.

Lastly, there is an increasingly popular type of game called "massively multi-player on-line" games (MMOs), where there can be tens of thousands of players all playing on one server cluster. This type of game provides a rich and ever changing set of data which could be used to train intelligences for that game.

Multi-player games present an ideal laboratory for testing the effectiveness of any artificial player being developed. We can create a Turing test where we can put an artificial player into the game without the knowledge of the human players. If the human players can see no difference between a human player and the artificial player under test, the test could be considered a success.

3 THEORETIC CONCEPTS

The game types defined in the first layer can be used to specify game theoretic concepts in the second layer of the taxonomy. These concepts are based on game theory and information theory and provide a bridge between games and game types to AI and AI methodologies.

3.1 Players and Actors

Only one concept from game theory applies to players and actors of relevance in this level, and that is the concept of co-operation. *Co-operative* games are those where the participants can form binding agreements on strategies, and there is a mechanism in place to enforce such behaviour [21]. *Non-co-operative* games are where every participant is out to maximise their own pay-off. Some games may have elements of both co-operative and non-co-operative behaviour: coalitions of participants enforce co-operative behaviour, but it is still possible for members of the coalition to perform better, or receive better rewards than the others if working alone. These are *hybrid* games.

In this taxonomy, if there is a mechanism strictly in place to prevent co-operating participants from breaking away and conducting their own behaviour then we can class that as a *co-operative* game.

Player-as-manager games naturally fall into the *co-operative* category, as the player is managing a team of actors with a shared goal, and with restrictions in place as to acting in a detrimental fashion to fellow team members. Conversely, *player-as-actor* games naturally fall into the *non-co-operative* category, as most games of this type pitch the player against all other actors.

3.2 Game World

3.2.1 Discrete vs. Continuous Actions

Discrete action games within game theory consist of a finite number of participants, turns, or outcomes, resulting in a finite set of strategies which can be plotted in a matrix format for evaluation. *Continuous action* games, however, can have participants joining and leaving the game, or the stakes changing between actions, resulting in a continuous set of strategies. This represents a subset of the potential actions that the game world allows.

Within our taxonomy, this relates directly to *environmentally discrete* and *environmentally continuous* game worlds.

3.2.2 Simultaneous vs. Sequential

In direct relation to *turn-based* versus *real-time* games, *sequential* games have all the players within a game make their moves in sequence, and one at a time [12]. *Simultaneous* games are those where any or all players may make their moves at the same time. Classically, sequential games are also called dynamic games [26]: however this would cause confusion in our taxonomy. *Sequential* games allow the construction of the *extensive form* of the game - essentially a hybrid decision tree of all players and all possible moves with their rewards.

3.2.3 Information Visibility

It is not necessary for all participants within a game to have access to all information about the state of the game at any point. The available information can be *perfect*, where all participants have access to the current state of the game, all possible strategies from the current state, and all past moves made by all other participants. The latter implies that all games that impart perfect information to the participants are by their nature also sequential games [14, 19]. *Imperfect* games impart partial information about the game to at least one participant. A special case of imperfect information visibility is *complete* information where all participants are aware of all possible strategies and the current state of the game, however the previous moves by other participants are hidden.

In this taxonomy, this relates to *accessible* and *inaccessible* game worlds. However, a common observation in games is that artificial participants have access to *perfect information* of the game world, whereas the human player only has *imperfect information*. This breaks the immersion of the game.

3.2.4 Noisy vs. Clear

Noisy game worlds are those in which there is a significant amount of information that an intelligence must analyse to either make a decision on what to do next, but where not all of that information is appropriate to the goal. Both *dynamic* and *non-deterministic* games provide levels of noise: the former due to the fact that the state of the game keeps changing even during the times when the intelligence needs to make a decision or form behaviour from learning; and the latter where there is no clear progressive state from which to base rules and analyse the game world. Although these two definitions provide the clearest example of noise, the level of information available can also create noise. Even in *perfect information* game worlds it is possible that the available information is an overly large data set for any intelligence, and thus noise is introduced.

3.3 Other Theoretic Considerations

As with our game type concepts, there are other theoretical concepts that can be used to create more concrete solutions, independent of the individual concepts above.

3.3.1 Nash Equilibria

Within game theory, a Nash equilibrium (NE) exists where an overall highest level of pay-off for all players takes place [20]. There may be many such equilibria within game strategies for a particular game, or there may be only one - in which case it is a *unique* NE.

Although the NE theorem has its problems, finding an NE for any particular state within a game is considered the accepted way of finding a strategy for game playing. The majority of NE finding algorithms are inefficient, however they could potentially be used by an intelligence to choose strategies.

3.3.2 Zero-sum

These are a type of game in game theory that are a special case of general sum games - ones where there is a fixed overall value to winning (or losing) the game [20]. The specific case where for any winning value v, there is a losing value of $0 - v$, is a zero-sum game. In other words, what one player wins, the other loses.

This does not relate directly to elements within our game world definitions above, but it does relate to the nature of games as they are played. Given there is no actual "value" in winning a computer game, even assigning arbitrary values of (1, -1) to the winning and losing values of the game will only provide a zero-sum game for two players - when subsequent players are introduced to the same game, we no longer have a zero-sum game unless we then re-assign arbitrary values.

Some team-based games implement the zero-sum concept for scoring within the game, where a winning strategy is eliminating the opposing team from the game. Where participants face a situation of i versus i team sizes for $2i$ players, the scoring of the game then becomes $(i, -i)$ upon elimination of one team by the other.

Some MMOs implement a zero-sum economy, where the game world provides resources which can be extracted and then through combat or trade can move from one hand to another, but will not leave the economic system. Note that this can lead to static gameplay where the majority of the game's wealth is in the hands of a very small proportion of the players.

3.3.3 Symmetry

In game theory, a two-player game is classed as symmetrical if, for player 1 and player 2, and the matrices A and B containing the state-action utility values for the strategies for player 1 and player 2 (respectively), $A = B^T$ *i.e.* if player 1 and player 2 swap positions, they can follow the same strategies, with the same pay-offs. This can be generalised to any number of players, where no-matter which players are swapped, the game can be played as it was before. Asymmetrical games rely on the strategies of whichever player's turn it is (*i.e.* first-player-wins-game-or-draws, such as *Noughts and Crosses*).

This could be an important test for a game, as it will allow game designers to ensure that artificial participants and human players are playing the same game, so that the human player will not consider any artificial players to be "cheating". If a game is not symmetrical for all players, then depending on the suitability tests designed for the artificial players, any solution could automatically be considered a failure.

4 ARTIFICIAL INTELLIGENCE METHODOLOGIES

Using the game theoretic concepts in the second layer of the taxonomy, we can connect these to AI methodologies in the third layer of the taxonomy. This section is not segregated into a participants and game world sub-sections as many of the methodologies cross this boundary and can be applied to both. It deserves mention that the AI methodologies listed here are by no means complete. We hope however that using the general concept of the taxonomy, those methodologies not mentioned will be able to fit into the taxonomy accordingly.

4.1 Agents

Our definitions of game worlds are based around rational agents, and as such we can apply agent-based methodologies to our taxonomy, by applying the metaphor *a participant is an agent* (see [24]).

4.1.1 *Reflex Agents*

These agents use a conditional statement to provide the "intelligence". Currently, most actors within games follow reflex systems, to the extent that players can monitor the input-output action pairs of specific actors. Once a pattern has emerged, the human player can modify their strategy sufficiently so that the opponent artificial actor will make a significant loss whilst the human player will make a significant gain. *Reflex agents* can fall into infinite loops, as there is no concept of context within if-then statements.

It has been stated that *reflex agents* can only exist within an *accessible* environment [24], but this would be true only if the scope of the condition covered the entire environment. Smaller local conditions can still be met and actions then carried out, irrespective of knowledge of the wider environment.

Temporal agents can be considered a special sub-group of *reflex agents*, where actions are carried out after measuring the passing of time. This specific type of agent would be applicable in *dynamic* and *semi-dynamic* game worlds, where time is a factor.

Reflex agents, are useful in situations where a high level of complexity is not required by a participant. The more limited the scope of possible actions in a *discrete actions* game world, for example, the less complexity is required in decision making. In some cases *reflex agents* might present the best compromise of complexity versus believability.

4.1.2 *Model-Based Agents*

An agent that monitors the environment and creates a model of it on which to base decisions is called a *model-based agent*. This type of agent would be best applied to *dynamic*, *real-time* games where constant monitoring of the environment is required on which to base decisions on actions. This would also be highly beneficial in a *co-operative* game, where although the actions of other actors are independent, they are inter-related, and so a broader monitoring range covering other co-operating actors can be introduced.

4.1.3 *Goal-Based Agents*

Using a model of the environment, goals can be created and planning carried out to achieve those goals, even within *inaccessible* game worlds and with other participants. Although the artificially controlled participants will generally have broad goals built in to determine their over-all behaviour (such as "stop the human-player at all costs"), there is still scope within that command to create sub-goals (such as "find the player") *Goal-based agents* are also highly beneficial in *inaccessible* game worlds, as they can change their own sub-goals as the information they are made aware of changes.

4.1.4 *Utility-Based Agents*

A further refinement on the *model-* and *goal-based agent* methodologies is the ability to manage multiple goals at the same time, based on the current circumstances. By applying utility theory to define the relative "best" goal in any situation, we have *utility-based agents*. These would be especially useful in *player-as-manager* games, or in an *inaccessible*, *real-time* game world, where we can apply the steadily changing state of the available game world to the various goals as the information becomes available, and then alter behaviour in a gradual manner.

4.2 Computational Intelligence

4.2.1 *Fuzzy Systems*

Fuzzy systems utilise a qualitative approach to information, whereby the incoming data is banded into groups, and it is the membership within a group that is used as a value. For example actors within games usually have some sort of "health" determining their ability to take damage before being removed from the game. By using fuzzy systems, such terms as "low", and "high" health can be used to calculate utility values of strategies.

Fuzzy systems fit naturally into *continuous action* game worlds - given that the input of a fuzzy system is a bounded range, this works even with the example before of turning to any direction: if another goal is "not too far off" an actor's current goal course, then it might provide greater efficiency by achieving the local goal first, even if it is of a slightly lesser priority.

4.2.2 *Neural Networks*

Neural networks (NNs) [13] have already been used successfully in a game world, as in the case of NERO. The team behind NERO are actively researching in this direction. Uses can be seen for *NNs* in *model-* and *utility-based agents*, where the current state of the agent is monitored all the time. *NNs* are particularly good for *noisy* game worlds: through many data sets the irrelevant information can be weeded out.

4.2.3 *Evolutionary Computing*

Elements of *evolutionary computing* (EC) [9] have been used as an aid to finding NE [6]. This is because an NE is considered a global maximum problem - something that *evolutionary computing* is especially good at.

Given that finding an NE is considered a solution to a game for a player, finding such equilibria would allow interacting actors to maximise their game playing. If we consider actors taking actions as a game or sub-game, this would let the intelligence behind the actor make the best action based on the results of finding a NE. NE are not always easy to find, and current algorithms are mostly inefficient [20].

Specific to the taxonomy, EC methods in general are good at optimising in a noisy environment and are thus well suited to *noisy* game worlds, and their use of individuals in a population makes them well suited to model *co-operative* game play.

4.2.4 *Swarm Intelligence*

Swarm intelligences [8] have also been used to compute NE [22]. Ant colony modelling provides a strong methodology for actors to explore the game world to complete goals by providing path-finding around obstacles and creating search patterns to achieve their goals - something that has been seen to be lacking in games. Exploring the game world is important in *imperfect information* worlds, and such group goal finding resulting from ant colony optimisation is useful in *co-operative* game play.

4.3 Classical AI

Classic AI [23] is based around symbolic representations of knowledge to create decision trees and model knowledge-based systems.

4.3.1 Backtracking

Turn-based, discrete action, static, perfect information games are perhaps the easiest types of games to solve - *sequential* play with a fully visible game world allows the easiest construction of the extensive form of the game, which is a type of decision tree, allowing such methods as minimax and A* searching to be used.

4.3.2 Knowledge-Based Systems

Knowledge Based Systems (KBSs) provide a catalogue of information from which deductive reasoning can take place, and are primarily used as expert systems to augment the deductive capacities of certain fields, such as medical diagnosis. Certain games rely on working out the solution to a problem before an artificial participant does, and such reasoning would help in this context. Given the nature of deductive reasoning, KBSs would only be applicable in *accessible* game game worlds.

4.3.3 Rule and Induction Systems

Unlike *KBSs*, *inductive and rule based reasoning* can work in *inaccessible* game worlds, as they rely on what information is available only to reason with. These systems are however labour intensive to create, and highly volatile to error, and such would not be applicable in *real-time, dynamic* games, where there is a large range of possible elements from which to induct reason.

4.4 Other Artificial Intelligence Considerations

There are a number of other areas within AI that can relate to AI in games, although they are not specific to any one classification within the taxonomy.

4.4.1 Believable Agents

These agents are expected to behave as a human would in similar situations. Given this is one of the core purposes of developing better AI in games, any agents that are developed should fall into this category, unless (through the narrative) the actor is expected to behave differently. Even then they should be consistent in their behaviour which could be construed as providing believability in the behaviour across all actor types. Specifically, *player-as-manager* games require a great deal of believability given the large number of actors available to the player, and the semi-autonomous nature of those actors. *Player-as-actor* games will also require a high level of believability, as all the interactions with other actors in the game world must provide a sufficient level of immersion, given the player is essentially existing within an artificial social system.

Believable agents as artificial players must use the same rule set as a human player, otherwise it would be unfair to the human player. More importantly if the artificial player learns within what it considers to be its game world, then it may carry out actions that will alert the human player to the artificial nature of their opponent. This would then break the immersion and thus the believability of the agent. If we take Laird and van Lent's analysis that game intelligence will help with general intelligence, this also means that we could not generalise any intelligence that makes use of programming loopholes when it comes to other domains.

4.4.2 Machine Learning

Realistic NPCs can be made to learn, and machine learning can be used to achieve this: there is a large amount of literature on the field of machine learning as this is one of the core components of artificial intelligence, and as such there are numerous methodologies for learning in any particular situation.

Supervised learning involves giving a system a specific scenario, and then telling it what the desired outcome is. This is done repeatedly over manned training sets to allow the intelligence to learn the correlation between the two. The scenarios are carefully constructed to ensure that the correct behaviour is learned, and takes a great deal of time. In our case this would consists of creating a scenario with dumb actors who need to be specifically controlled, and then carry out the actions that the player would want in that situation, and doing this repeatedly with slightly differing scenarios to reduce the noise of the input. Neural networks are trained this way.

In the majority of cases, players do not want to spend a significant amount of time training the behaviour of what could potentially be disposable actors, thus it would be desirable for the actors to learn the optimal behaviour in any given situation without the player's direct interaction, which leads us to *unsupervised* learning. There is, however, a problem with this: such a method requires a large number of training sets that might not be available by merely monitoring each game as it is being played. A possible solution would be to record scenarios of the player playing, and then analyse these scenarios outside of the game itself.

Even this solution has problems, namely that the optimal behaviour that the learning algorithm develops might not be the desired behaviour of the player for any co-operative or subordinate actors. In which case it would be useful to implement *semi-supervised* learning, where the player can put the actors into a supervised learning mode, which would then let them monitor exactly what the player desires of them, so that they can develop behaviours based on that.

Reinforcement learning has been used to find NE within a given game context to solve strategies within multi-agent systems [4] (and thus could be used in conjunction with agent theory). The solutions are applicable to *dynamic*, *simultaneous* games: as the agents (or in this case generalised actors) learn, the strategies available may alter, and also the strategies being followed by other actors may alter as well. NERO is an example of reinforcement learning. As mentioned in the NERO paper [7], this could introduce a new type of game playing where the aim of the game is to actively train the actors within the game with the desired behaviour for the scenarios in which those actors will be placed. Operatives within NERO are given rewards and punishments for successful behaviour. It is this reinforcement of correct (or incorrect) learning that gives rise to the name of this methodology. The transition to game strategies is a simple one, as each game has a reward, and maximising the reward maximises the reinforced learning.

Quite often actors will be faced with a set of problems at once. *Multi-objective* learning techniques takes this whole set and learns how to solve each problem by analysing the whole set, and extrapolating common themes between the problem which allow a greater categorisation of problems in the future [5]. This is pertinent in *dynamic, real-time* games, which - especially when used with utility-based agents - might present a set of problems or objectives to be completed.

4.4.3 Natural Language Processing

The aim of natural language processing (NLP) is to create a language interface between AIs and humans that seem clear and realistic. Most current games that implement a dialogue system between human players and actors most provide a *discrete action* game world to do it in. There are only a limited number of options available to the player to discuss with the actor. Whilst this is not normally a problem, it can break immersion if a conversation topic is available for discussion that is seemingly unrelated to the actor - something that keen observers would note and then assume that this actor had a bigger part to play than initially thought. NLP can bypass this by allowing a continuous conversational interaction with between the human player and actors in the game, such that the human player would have to put in some effort into the conversation to gain the information needed - or to discover that that actor did not know anything about what the player needed. This is would greatly increase the immersion of the game.

It is useful to note that most text-based adventure games mostly have an open way of playing the game. The human player could type entire actions into the game to be carried out, and the game would interpret these actions, albeit using keyword analysis.

5 USING THE TAXONOMY

Having now created the taxonomy, it would be useful to discuss the envisioned use of the taxonomy, taking a number of examples: first we will illustrate the overall mechanism for using the taxonomy between game developers and AI researchers; then we will illustrate the reverse mechanism, from AI researcher to game developer. We further show the use of the taxonomy from a purely research-based approach, and from a purely game development approach. Lastly, we also provide an example of a specific game, showing the identification of concepts within our taxonomy. Figure 3 shows the bi-directional traversal of the taxonomy.

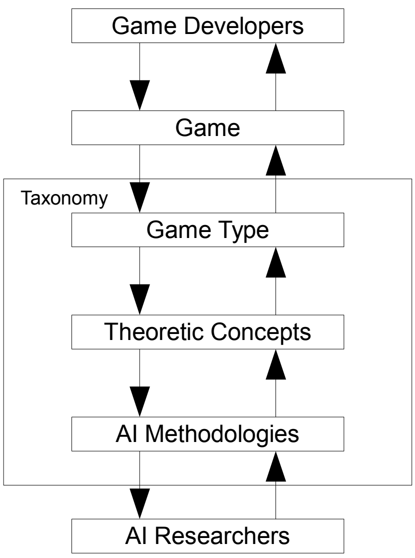


Figure 3. Bi-directional traversal of the taxonomy

5.1 Game Developers to AI Researchers

The most likely scenario for use will be where a game developer wants to incorporate sophisticated AI into their game, and wishes to develop this with the assistance of an expert in the field. The game developer will be able to apply the *game types* layer directly to their proposed game, picking out the interaction and game world concepts identified above: for example the proposed game is a *dynamic, real-time, environmentally discrete, deterministic, accessible, player-as-actor* game.

Using this information, the game developers map the identified concepts from the first layer to the second: again using our example, this gives us the theoretic concepts *non-co-operative, simultaneous, perfect information, discrete actions, noisy*.

Lastly, the game developer identifies a number of AI areas that might be beneficial from this analysis, mapping from the second layer to the final layer. Potential methods can be used, such as *neural networks* or *evolutionary computing* to cope with the *noisy* game world; *tree searching*, given the *perfect information* available and the *discrete actions*; the use of *utility based agents*, given the *noisy* environment and the *non-co-operative* nature of the game-play; or any combination thereof.

With this as a base, the game developers can then approach experts in the field focussed on the methodologies identified to collaborate on specific implementations within the game.

5.2 AI Researchers to Game Developers

The reverse of the above would see an AI researcher develop a novel approach to a particular method within AI, for example utility based agents. The researcher can then identify which of the concepts in the second layer are pertinent to their research. We can see that utility based agents are beneficial for *non-co-operative, imperfect information, noisy* games.

From this, the researcher can map from the second layer to the first layer to identify the overall type of games that their method could be applied to. We can see that this would be *player-as-actor, inaccessible, dynamic, non-deterministic* game types. This allows the researcher to approach various game developers that develop those types of games, to see if there is any scope for collaboration.

5.3 Code Library Selection

Game developers, and developers in general, find the use of various code libraries to be beneficial when creating games - by using a pre-developed set of interfaces, development time can be reduced. It is possible that certain libraries of AI methods could be implemented to provide intelligence for game actors and players, to again reduce the development time associated. As before the game developer would traverse through the taxonomy and identify the types of AI methods that could be used. They could then identify the various libraries that would be beneficial for their completed product. AI researchers may still be needed to provide their expert knowledge, as even though these libraries may provide the technicalities, they will need to be adapted for each instance - in which case the use of the taxonomy is still beneficial in providing common concepts for discussion.

5.4 Experimental Laboratory Selection

AI researchers also benefit from the taxonomy for pure research purposes. As mentioned in the introduction, games can be used as a laboratory in which to conduct experiments and research into various AI methodologies. Using our taxonomy a researcher can identify a type of game to test their ideas, and through analysis of various games available short-list a number of alternatives that will provide them with the environment they need.

5.5 Real Game Example

Fallout is a role-playing game set in an alternate-universe future. The game is single-player, focused on the actor that the player is controlling. The player may meet other actors within the game who join them in their mission and aid them, however these are uncontrolled by the player, thus the interaction is *player-as-actor*. The entire world cannot be seen at once, thus the game world is *inaccessible*. The primary actions of the character take place in *real-time*. The game world is divided into a hexagonal map through which the game actors move - an *environmentally discrete* game world. The game is *dynamic*, as other actors in the game may carry out actions whilst the player is thinking; given this, and the fact the game world is inaccessible, the game world is also *non-deterministic* - the player (or actors) cannot explicitly determine what will happen next. Normal game-play is *simultaneous*, and other actors in the game can be seen to have regular objectives depending on the time of day or other external stimuli. Despite the fact that other actors may join with the player's actor, there are no mechanisms in place to explicitly restrict non-co-operative action: *e.g.* the player may choose to engage in combat with the friendly actors; thus making the game *non-co-operative*. As the world is both *dynamic* and *non-deterministic* it is also *noisy*.

Fallout has a number of different levels of game-play: primary game-play (as described above); combat game-play; and fast travel game-play. Combat game-play changes to be *turn-based*, and also reinforces the non-determinism of the world by using random numbers for such mechanisms as accuracy and damage of particular weapons on each hit. Fast-travel game-play changes to be *static*, as the game state does not change whilst the player chooses where to travel; and *environmentally continuous*, as the fast-travel map presented to the player allows the player to move their actor anywhere on the map.

Using the taxonomy, we can see that the most applicable AI methods to implement in *Fallout* would be *evolutionary algorithms*, *neural networks*, or to use a *utility-based agent* approach. Other methods are available, but when taken as a whole there are competing aspects of the game world that mean we must take the most widely suited: for example, although the *inaccessible* nature of the game world could point to the use of *inductive reasoning*, the *noisy* nature of the environment precludes this.

6 CONCLUSION

Given both the interest in the game industry for realistic actors within games, and the opportunity for involvement by academic researchers, we hope that the taxonomy described in this paper will provide a starting point by which to facilitate collaboration between the two sides in order to further both agendas. The taxonomy presented here is in the early stages of development. It is hoped that a refined version of the taxonomy will provide developers with a framework within which they can discuss AI techniques with relevant experts. Conversely, a fully developed taxonomy should also provide AI researchers with a formal process for utilising AI techniques in a gaming context.

Future work is aimed at further developing the taxonomy and in evaluating its utility. We aim to conduct a comprehensive study of the AI literature in order to map existing techniques to our taxonomy, refining it as necessary. Although we have highlighted the most obvious examples of existing work in the initial taxonomy, this process will no doubt also identify additional approaches utilised in AI that can be useful in a games environment. An additional feature of the taxonomy development is likely to be directed towards the construction of an *ontology*, as referred to in the introductory section, to be used alongside the taxonomy. This will facilitate dialogue between developers and researchers by formalising the concepts within the taxonomy, leading to greater understand and better communication on both sides.

An important aspect of our future work is to validate that the proposed taxonomy is both *useful* and *correct*. In order to achieve this, we are working towards identifying a series of metrics which can be used to measure the success of the taxonomy. Close involvement with industry is critical to this process. From a practical perspective, we aim to illustrate the effectiveness of the taxonomy by identifying a number of case-studies which will enable the implementation of an effective game by following the taxonomy.

The taxonomy was initially developed in order to select a relevant game environment for using as a laboratory in which to experiment with AI techniques. However, we hope that a more fully developed version will have a much wider scope, proving useful across the spectrum of game development and AI research.

REFERENCES

[1] E. Aarseth, 'Playing research: Methodological approaches to game analysis', in *spilforskning.dk Conference*, (August 2003).

[2] Thomas H. Apperley, 'Genre and game studies: Toward a critical approach to video game genres', *Simulation Gaming*, **37**(1), 6–23, (March 2006).

[3] Jay D. Bolter and Richard Grusin, *Remediation: Understanding New Media*, The MIT Press, February 2000.

[4] M. Bowling and M. Veloso, 'An analysis of stochastic game theory for multiagent reinforcement learning', Technical report, Computer Science Department, Carnegie Mellon University, (2000).

[5] Richard A. Caruana, 'Multitask learning: A knowledge-based source of inductive bias', in *Proceedings of the Tenth International Conference on Machine Learning*, pp. 41–48, (1993).

[6] *Evolutionary Computation in Economics and Finance*, ed., Shu-Heng Chen, Studies in Fuzziness and Soft Computing, Physica-Verlag Heidelberg, May 2002.

[7] T. D'Silva, R. Janik, M. Chrien, K. O. Stanley, and R. Miikkulainen, 'Retaining learned behavior during real-time neuroevolution', in *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, (2005).

[8] Russell C. Eberhart, Yuhui Shi, and James Kennedy, *Swarm Intelligence*, The Morgan Kaufmann Series in Artificial Intelligence, Morgan Kaufmann, March 2001.

[9] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Natural Computing, Springer, November 2003.

[10] C. Fairclough, M. Fagan, Mac, and P. Cunningham, 'Research directions for ai in computer games', in *Proceedings of the Twelfth Irish Conference on Artificial Intelligence*, (2001).

[11] G. Frasca, 'Simulation versus narrative: Introduction to ludology', in *The Video Game Theory Reader*, chapter 10, Routledge, (2003).

[12] Yoav Freund and Robert E. Schapire, 'Game theory, on-line prediction and boosting', in *COLT '96: Proceedings of the ninth annual conference on Computational learning theory*, pp. 325–332, New York, NY, USA, (1996). ACM.

[13] K. Gurney, *An Introduction to Neural Networks*, Ucl Pr Ltd, 1997.

[14] J. C. Harsanyi, 'Games with incomplete information played by "bayesian" players, i-iii. part i. the basic model', *Management Science*, **14**(3), 159–182, (November 1967).

[15] A. Kleiner, 'Game ai: The possible bridge between ambient and artificial intelligence', *Emerging Communication*, **6**, 143–158, (2003).

[16] L. Konzack, 'Computer game criticism: A method for computer game analysis', in *CGDC Proceedings*, (2002).

[17] John E. Laird and Michael van Lent, 'Human-level ai's killer application: Interactive computer games', in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 1171–1178. AAAI Press / The MIT Press, (2000).

[18] Craig A. Lindley, 'Game taxonomies: A high level framework for game analysis and design', *Gamasutra.com*, (2003).

[19] Oskar Morgenstern and John Von Neumann, *Theory of Games and Economic Behavior*, Princeton University Press, May 1944.

[20] Noam Nisam, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, September 2007.

[21] Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory*, The MIT Press, July 1994.

[22] N. Pavlidis, K. Parsopoulos, and M. Vrahatis, 'Computing nash equilibria through computational intelligence methods', *Journal of Computational and Applied Mathematics*, **175**(1), 113–136, (March 2005).

[23] E. Rich and K. Knight, *Artificial Intelligence*, McGraw Hill Higher Education, 2 edn., March 1991.

[24] Stuart J. Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach (2nd Edition)*, Prentice Hall, December 2002.

[25] J. Schaeffer, V. Bulitko, and M. Buro, 'Bots get smart', *IEEE Spectrum*, (December 2008).

[26] James N. Webb, *Game Theory: Decisions, Interaction and Evolution*, Springer Undergraduate Mathematics Series, Springer, December 2006.

[27] Mark J. P. Wolf, *The Medium of the Video Game*, University of Texas Press, June 2002.

A Critical Review of Interactive Drama Systems

Maria Arinbjarnar and **Heather Barber** and **Daniel Kudenko**[1]

Abstract. This paper provides a critical review of existing interactive drama systems. An interactive drama takes place within a virtual world in which the user has a high degree of freedom to physically and mentally interact with non-player characters and objects within a dramatically interesting experience which is different on every play, and adapts to the user's interactions. Those criteria which should provide the basis of interactive drama are detailed and discussed in this paper. The main existing interactive drama systems are then surveyed. The techniques used in each system are discussed, as are the contributions, and shortcomings, of each system. There is great potential for interactive drama systems, and this paper considers how best to achieve this.

1 INTRODUCTION

Storytelling is appreciated by many people, as both teller and audience. In the past stories were only told orally, with audience participation. It is still true that listening to a friend narrating a story is an enjoyable way to spend time. However, as storytelling has evolved - through drama, writing, print, film and television - interactivity has been neglected. Receivers (listeners, readers or viewers) of a story will often want to become more involved in the storyworld, perhaps even to become a character. An interactive drama offers a world in which the participant can have a real effect - both long and short term - on the drama which they are experiencing.

Many computer games involve a story, which in most cases is an essentially linear story or series of stories (multi-linear). The most complete stories can typically be found in Role-Playing Games (RPGs), First-Person Shooters (FPSs) and Adventure Games (AGs). This linear element constrains game development because it limits the user to following one of the pre-defined story-lines. However, the incorporation of a story enriches the game, by providing a cause and a motive for the game and the user's actions, thus greatly increasing the potential for immersion and engagement.

A linear or multi-linear story is clearly not an interactive drama, because it cannot satisfy the need for interaction which has a clear effect on the drama development a sufficently large number of times (the number of fundamentally different narratives which can be generated is very limited). There are games with no explicit story structure (simulations) in which the user is encouraged to perceive their own stories within the world. These stories are truly interactive, but lack the structured drama development required to ascertain satisfaction of the need for a dramatically interesting experience. This tends also to lead to a lack of ability to emphathise with characters, an ability which may increase immersion.

There are various terms used for the research field, for example 'interactive drama', 'interactive storytelling' and 'interactive narrative'. The term used by each research group tends to reflect their short-term presentation technique, for example 'interactive narrative' is often used for systems which currently generate a high-level plot outline. These systems can be discussed within the same evaluation as their core aims are the same. The term 'interactive drama' is used here as this was the term which was first used, and thus is the most reflective of the extensive coverage of systems by this paper, which incorporates systems which have been developed throughout the course of research in this field. As discussed in section 2, drama is also the most appropriate term for the ultimate aims of research in this field.

The paper provides a critical review of the current state of the art in interactive drama. There is a great disparity in the field and a lack of consolidation. With a basis such as that provided by this paper it is possible to consider the state of research in this area in a more unified manner. This should enable research in the field to move forward with a basis in common understanding. An evaluation of this type enables researchers to be able to easily identify the shortcomings and contributions of previous research in the field, which they will then be able to build upon in their own future research and contributions.

Since Mateas's 1997 Oz-centric review of interactive drama systems [40] there has not been a comprehensive summary of the main research in this area despite this having been extensive. Roberts et al [56] summarised that subset of interactive drama systems which utilise a drama manager, but does not take into consideration the other methods which can be used to generate interactive dramas.

This paper begins by defining interactive drama as it is ultimately required to be (section 2). In this discussion the major aspects which should be considered essential in creating an interactive drama system are discussed in detail and justified. These are: a virtual world in which the narrative will take place; interaction with objects; social interaction; dramatic structure (which supports dramatic interest); fundamental difference in the narratives generated. This is followed by an overview of existing interactive drama systems (sections 3 and 4) with similar aims. Each system uses its own technique for drama generation. However there are fundamental similarities between many of these methods. A plot graph structure is often used, as discussed in section 3. This may be used in combination with planning techniques. Those systems which use other methods for generating interactive drama are discussed in section 4. This is followed by a discussion of those systems which do not allow the user to have first person control of a character (section 5). Each system's level of achievement in accordance with the basic requirements discussed in section 2 is summarised in section 6. The paper finishes by considering the possible future for interactive drama and research in this area.

2 INTERACTIVE DRAMA

There are various definitions and conceptions of interactive drama (or narrative), which provide the basis for research in the field. These

[1] University of York, York, email: {maria,hmbarber,kudenko}@cs.york.ac.uk

include those found in the following work: DED [4], FatiMA [8], NOLIST [9], GADIN [10], the OZ project [16], I-storytelling [18], Erasmatron [19], OPIATE [24], Virtual Theater Project [29], Laurel [32], IDA [37], Façade [41], U-DIRECTOR [44], Murray [46], SASCE [47], Bards [50], IN-TALE [55], Ryan [57], DEFACTO [59], IDtension [64], PaSSAGE [65], and Mimesis [69].

The various definitions have core similarities and identify the same essential requirements. Having considered these, as well as definitions found in narratology and drama theory, interactive drama as it will be considered in this research can be defined. This identifies the ideally required components of an interactive drama system, and is given and elaborated on here.

An interactive drama takes place within a virtual world in which the user has a high degree of freedom to physically and mentally interact with non-player characters and objects within a dramatically interesting experience which is different on every play and adapts to users interactions.

Drama when used in this paper refers to "moment by moment action, a scenic rendering of speech and behaviour of characters, careful detailing of specific events, commonly contrasted by panorama" [51].

Virtual worlds The exact depiction of the world will depend on the genre of story to be experienced. For example a *Dungeons and Dragons* RPG needs to include dungeons and monsters. The world in which the drama will take place needs to have an appearance of completeness which is sufficently high to allow the users to feel that they are experiencing freedom within the virtual world.

There are many existing virtual worlds in which an interactive drama could take place. These include game worlds, such as Fallout 3 [34] and Neverwinter Nights [17]; and virtual realities, such as Second Life [31] – in which the user creates their own avatar and is considered to be a resident of the world. The use of an open source 3D environment for an interactive drama would give the programmer greater overall control, but may not be compatible with other virtual worlds.

Interaction with objects The user of an interactive drama is likely to become frustrated if they do not perceive themselves to be able to freely select their actions within the virtual world, within reasonable limitations. At a minimum this freedom should be adequate for user-friendly interaction with characters, objects and scenes of the drama. As Laurel [32] explains, it "is difficult to imagine life, even a fantasy life, in the absence of any constraints at all", for example gravity within a game world is not seen as limiting a user's freedom. Providing that any constraints are consistent with the user's perception of the game world the user will still believe that they are free within that world.

This can include interaction with other characters as if they were objects. This is a representation of actual physical interactions in the 'real' world. There is not a clearly defined boundary between this requirement and the next, that of social interaction. Many actions may combine both, for example assaulting another character involves interacting with them as an object, but also has a strong underlying social component.

Social interaction Social interaction involves interaction with other characters within the virtual world on a social level. For example gestures, spoken and emotional communication and expressions are all forms of social interaction. These should all be available to characters within the virtual world. Each character should be able to interact with all of the other characters in each of these ways. The use of language communication is discussed further in this section, as this is the most frequently researched of the social interactions.

The user will ideally be able to communicate freely within the virtual world, and be understood. This is frequently interpreted as requiring natural language processing (NLP), as for example in Façade [41]. This relies on the assumption that NLP provides the highest level of freedom and interactivity. However, this is not necessarily true. Current NLP technology will not allow characters to fully understand natural language, which means that only a restricted set of sentences will have the expected interpretation and thus the user must either know, or guess, the required input for their desired action.

A method which presents the user with a clear set of possible options may be considered to be more user friendly. These options must cover a wide range, to allow the user to identify a suitable representation of their desired action, otherwise it will be seen as limiting the user's freedom. An additional advantage of this method is that the user will be presented with options which they may not otherwise have considered, and thus supplements their imagination. This means that the user is still free to act but is not relying solely on their own creativity. In addition this method increases the level of mutual understanding between characters.

Within the drama other characters will mentally interact with one another. They should also initiate interaction with the user. This is as would be expected in a natural unfolding of a drama, whether interactive or otherwise.

Dramatic structure For the interactive drama to be successful the experience must be dramatically interesting for the user. The use of a dramatic structure supports the dramatic interest of the experience. Through history, storytelling and drama have captured the interests of many theorists. This began with Aristotle [5] in ancient Greece, and has continued with modern theorists including: Barthes [14], Esslin [21], Propp [52], and Todorov [66].

As a result of this research there are structures which can be used to aid in the development of an interesting drama. Freytag [27] proposed a graphical form for the analysis of plots, which is known as Freytag's Pyramid, as shown in figure 1. This referred to as a 'dramatic arc' [32] in this paper. The dramatic arc outlines the basic rise and fall typically found in an interesting drama. This begins with an inciting incident, which provides the mood and motive for the drama (*a*). The suspense will then be expected to steadily climb due to the increase in complications in the unfolding plot (*b*). This will cease at the 'climax' point (*c*). Following this the dramatic arc steadily descends (*d*) as the complications are resolved, and the drama reaches closure (*e*).

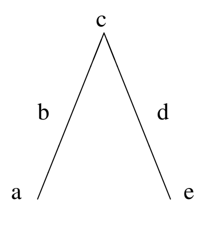


Figure 1. Freytag's plot structure (as given in [27]).

It is possible for an interesting drama to occur without a dramatic arc being followed, for example there may be a lack of closure. However, the more closely a drama conforms to the dramatic arc the more difficult it becomes to claim that it is not dramatically interesting.

Dramatic arcs have been utilised in previous interactive drama research. For example the Oz Project and IDtension require generated narratives to follow a dramatic arc [16, 64]. Façade uses a structure which they call neo-Aristotelian, an adaptation of the Aristotelian structure to interactivity [5].

Propp's morphology of the Russian folktale [52] provides a structure for Russian folktales, in the Aarne index [1], using a specific set of functions and characters. Propp's functions have been exploited by many systems, such as OPIATE [24]. With the exception of [3] very little attempt has been made to extract the morphology of a different set of tales, or story genre, to aid in the unfolding of interactive drama.

Although not all interesting dramas conform to a specific morphology, it is more likely that a drama which does conform to a well known dramatically interesting strcture will be of interest to a wide audience. For example the Field's [25] morpohology has helped to provide the structure of Hollywood films for many years, and is still one of the most used script structures in that domain.

Esslin [21] explains that any drama needs to capture and maintain the involvement of the audience by being constantly interesting. The audience are likely to frequently lose interest in the main storyline. It is thus essential to have sub-stories within the overall story, as these will ensure the continued attention of the audience. Such sub-stories will add to the complication of the plot, thus increasing the overall suspense within the drama. The events within these sub-stories must also follow a clear structure, as this will ensure the continued engagement of the audience. Sub-stories may be nested.

For example there may be a science fiction in which the main plot involves rescuing a spaceship and its crew, who are stranded in deep space. Possible sub-stories which could occur within this drama include: two crew members falling in love; or the captain's quest for a novel solution to reduce the energy consumption of the spaceship. The structure of the second sub-story could involve the engineer proposing a new way of conserving energy (the inciting event), a discussion of this method (the rise), the captain's decision as to whether to follow this proposal (the climax), and possibly the reasoning behind this decision (the closure).

Fundamental difference Each time the user participates in an interactive drama they should identify the story which they experience as being an essentially a new story. To achieve this the main storyline will need to significantly differ every time the user participates, which requires changes in the background and the inciting incident. Insignificant changes, such as only the ending differing, or characters having a different trivial conversation, will not be sufficient. The unfolding story should vary each time the user participates, in such a way that the user would identify it as essentially a new story each time. The difference in the story needs to be apparent from the outset of the drama.

For example, in a murder mystery the uniqueness of the drama occurs in the set-up, the characters, murder and scenes. This means that two mysteries with the same set of characters, but with a different character being the victim, will be essentially different stories, as this will cause a fundamental change to the actions of the characters, the identity of the murderer, and the clues required to discover the murderer and their motive. A change to the set of characters, or the scene, will also lead to a fundamentally different story. However if the set of characters, the scene and the murder remain constant, with the only difference occuring in the identity of the murderer, then the stories cannot be considered to fundamentally differ. As there will only have been subtle difference in the drama which determines that character's identity as the murderer rather than another.

The variation in the story must be highly responsive to the user's interactions. The user is essentially finding a narrative path through the storyworld in which an unnoticed dramatic interest guide will be the computerised playwright. The user should be able to act as and when they desire in ways which will have a perceivable long and short term effect on the narrative.

3 PLOT GRAPH STRUCTURE

A plot graph structure can be used to generate an interactive narrative. In this there are certain stages of interaction. These vary in length depending on the specifications of the system, but can be whole scenes or only a few seconds of action. Following these stages there will be pre-defined actions or sequences of actions which will lead to a new stage of interaction. Different actions (by the user or other characters) will cause variance in the narrative, as they may result in a different stage of interaction being the next to occur.

The major shortcoming of all systems which use a plot graph structure is their lack of extendability and generality. This means that there is also a lack of replayability. Each possibility for the narrative must be pre-defined, in itself and in the context of the stages of interaction which can precede or follow it. This involves a large amount of pre-definition. There will also only be a limited number of possible paths through the storyworld, as the plot graph must be followed. Repetition will occur after only a few experiences with the system, and thus the requirement of a fundamentally different narrative cannot be realistically satisfied – as such a large volume of material will need to be pre-defined.

3.1 The Oz Project

This was the first main interactive drama research group. The Oz Project [16] group created simple characters known as woggles. Research focused mainly on the creation of believable agents. The user could give instructions to one of these characters and play with them. These characters interacted with each other in the game world.

The group's work also included generation of interactive stories which were based on a plot graph structure. The path which the user took through this structure was dependent both on the user's choices and a pre-defined evaluation function, which biased the experience towards 'good' story-lines.

3.2 Virtual Theater Project

The work of the Virtual Theater Project uses the concept of 'directed improvisation', in which improvisational actors follow directions (and constraints), and provide the detail. For example an actor could be instructed to walk to a table, and if they are playing an energetic character they may rush there. The virtual worlds are populated by actors who take the part of characters.

The group worked on a number of different projects. In the Little Red Riding Hood system the user could destroy the story but would not be presented with a new story as a result, instead they were able to observe how their actions would move the story away from its predetermined course. The group's Master-Servant scenarios involved the servant, through a series of postures, switching places with the master [29]. In the cybercafé scenario there are a number of customers and a waiter in a café. The user gives directions to one of the characters, which they will improvise (in accordance with the individualities of their assigned character) to follow. The actions of the characters, whether instructed by the user or the system, are incorporated into the plot graph structure.

3.3 Façade

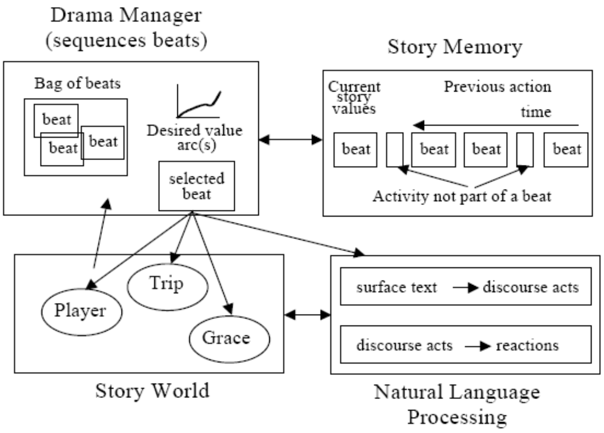


Figure 2. The Façade architecture [41]

In the Façade [41] system the user is invited to the house of some friends, a couple. While there they become immersed in the couple's marital difficulties and battles. The user is able to speak to the other characters and what they say - as well as how and when they say it - will affect the story they experience. The user's actions will determine the outcome of the story and thus the final state of the couple's marriage.

Façade comprises: a drama manager, beats, characters, story values, actions and natural language processing, see figure 2. Beats are short sequences of action which occur throughout the drama. They are explicitly pre-authored, with all actions within the beat being fully defined, and the actions of all roles being assigned to allow for multi-agent coordination [38]. The order in which beats occur can vary, but each has preconditions and effects of other beats. This is a plot graph structure in which each of the plot points is very short.

All higher level goals and behaviours that drive a character are located in the beats. The characters retain autonomy in achievement of base-level goals and in performing actions such as facial expressions or personality moves [38]. The authoring of Façade took 3 man-years and included 27 beats [39]. This has led to a game which lasts between 20 and 25 minutes and which can be experienced 3 to 5 times while still experiencing novelty in the story.

3.4 IDA

At the start of an experience with the Interactive Drama Architecture (IDA) [37] system, the user finds their own dead body. As a ghost they must find their murderer and subsequently manipulate another character into finding the body and also realising who committed this murder. The user has become a ghost, thus explaining their lack of freedom.

In IDA the author is required to pre-define: the story, any domain dependent functions of the director, the environment and art content, and character behaviours. The characters are semi-autonomous

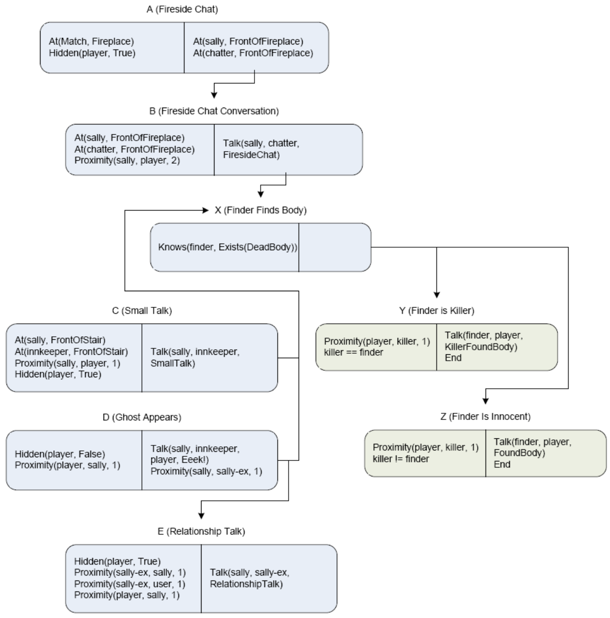


Figure 3. The partial-ordered plan for IDA [36]

in that they will act while they have no instructions from the director, for example drinking. Following commands from the director takes priority over all of their other goals. These commands can be high level, for example 'explore', or very specific, for example 'perform dialogue #131 with John in the library and then run away to another room' [36].

The story consists of plot points in a partially ordered graph, see figure 3. This uses STRIPS with pre- and post-conditions. There is limited variation in these plot points, such as where a certain scene can take place. The user's murderer is pre-determined and fixed.

The user is modelled to enable the drama manager to guide them through the storyworld as subtly as possible. Director actions that modify the plot to accommodate user actions are:

- Deniers, which permanently or temporarily make certain plot points inaccessible.
- Causers, in which the system initiates a plot point.
- Creations, which cause the appearance of new things in the game to replace destroyed items.
- Shifters, which move plot points.
- Hints, such as some noise from a room.

3.5 SASCE

SASCE [47] is an adapted TD-learning method for interactive drama. This method determines, based on a pre-defined evaluation function, the apparent best route for the story, depending on the actions the user is expected to take at each stage, and thus that which will lead to the highest overall score. The routes for the drama are selected from the possible routes through a pre-defined plot graph. The actions the user is expected to take are determined by a computer simulated user. These simulations provide the training data.

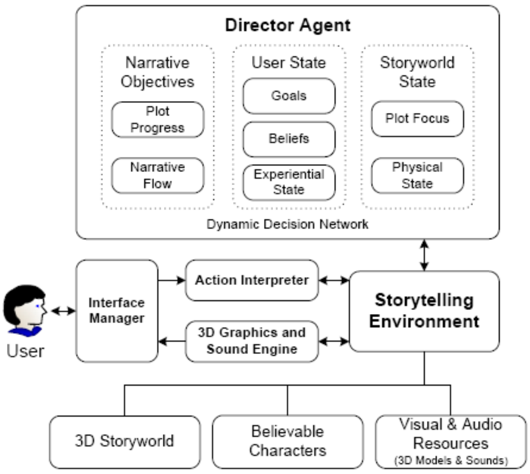


Figure 4. The U-DIRECTOR architecture [44]

3.6 U-DIRECTOR

U-DIRECTOR [44] uses HTN planning and dynamic decision networks to implement a medical mystery story that takes place on a secluded island, see figure 4. The story is pre-authored and follows a fairly strict plot. A Bayesian inference mechanism is used to decide how to manipulate the user into following the plot. This enables achievement of the desired ending, the solution of the mystery.

The director attempts to engage the user in the drama by providing hints which will lead them towards following the plot. If the hints are not sufficient then the director will become less subtle, for example by instructing another character to take the initiative in the required action. The director uses extended Bayesian networks to select a directive action based on the aim of maximising expected narrative utility.

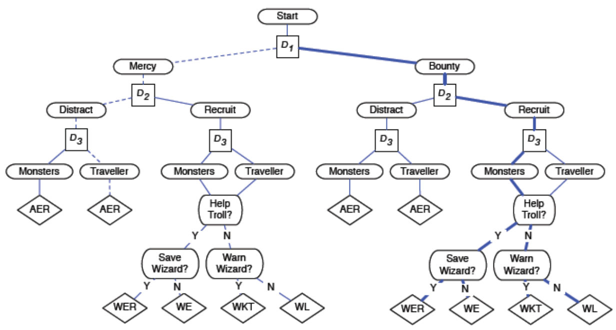


Figure 5. The current PaSSAGE plot graph [65]

3.7 PaSSAGE

(Player-Specific Stories via Automatically Generated Events) [65] focuses on the user-specific adaptation of the story. There are a number of possible 'encounters' which involve characters in interactions with one another. These follow a particular order depending on their type. The encounter chosen depends on which type of game player the user has been modelled to be, which is based on their choices in an introductory phase.

The encounters form a plot graph, see figure 5. However since the path taken is dependent on the user model there is likely to be a linear story experienced by the same user on subsequent experiences.

3.8 IN-TALE

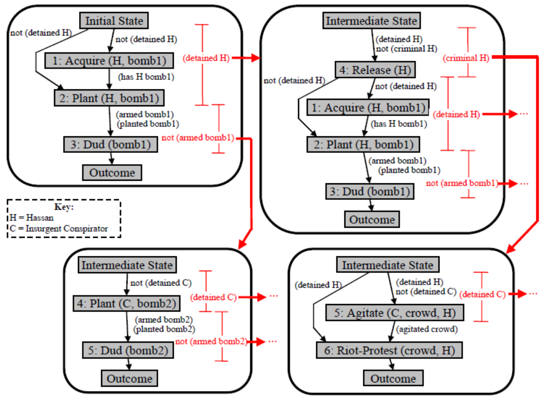


Figure 6. The narrative plan for the current IN-TALE training scenario [55]

The IN-TALE (Interactive Narrative Tacit Adaptive Leader Experience) [55] system is designed for training soldiers. The user will find themselves in a scenario which could occur in the line of duty. They will be able to act as freely as they would in reality and their actions will determine whether they are able to successfully diffuse the situation. The ending will adapt to ensure that the problematic events will always occur – however the user chooses to act.

The drama is generated based on a plot graph. Planning is used to determine whether the current path being followed is likely to be successful or if this needs to change, and the action adjusts appropriately, see figure 6.

3.9 Mimesis

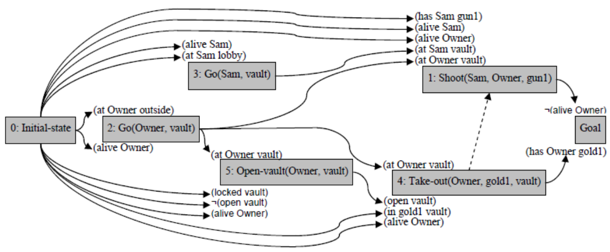


Figure 7. The narrative plan for a bank robbery in Mimesis [53]

The Mimesis [68, 53, 69] system was created as part of the work of the Liquid Narrative Group. It is designed as a general architecture and thus to work with any game engine.

An attempt is made to give the user of the Mimesis system the illusion of complete freedom. Following each user action not in accordance with the current plan the system decides whether the user's action can be "accommodated" or must be "intervened" with. An accommodated action must be incorporated into a new plan to achieve the story's goal. In the group's bank robbery scenario if the user opens the bank vault – which the plan requires another character to open – re-planning can accommodate this inconsistency by creating a plan in which that character does not open the vault but finds it open, see figure 7. If accommodation is not possible the system must intervene with the user's action. This could mean causing the user to miss when they attempt to shoot a character who must perform some role for the story's goal to be achieved, or perhaps a nuclear reactor's "control dial momentarily jamming ... [to] preserve the apparent consistency of the user's interaction while also maintaining safe energy levels in the story world's reactor system." [69]

When the Mimesis system receives a plan request it creates a directed acyclic graph (DAG) to achieve the story ending. This is a plot graph structure in which the plot graph can be redrawn within the narrative, but this variation is not sufficient for fundamentally different dramas to be generated.

4 OTHER TECHNIQUES

This section discusses those systems which do not use a plot graph structure for the generation of interactive drama. There are various other methods which have been used. The strengths and shortcomings of each of these methods, and the systems, are discussed here.

4.1 NOLIST

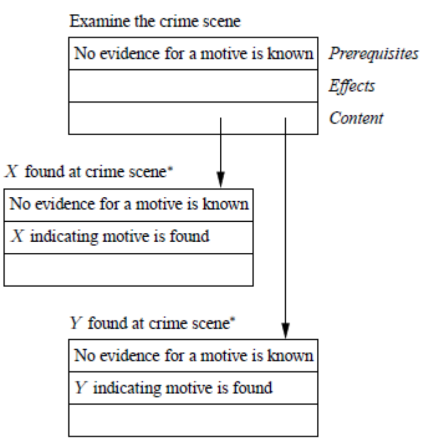


Figure 8. The action hierarchy for examining a crime scene in NOLIST [9]

In the non-linear interactive storytelling game engine (NOLIST) [9] a Bayesian network is utilised in creating a murder mystery. The Bayesian network dynamically changes in response to actions and observations made by the user. It is not preset but combines the user's actions and logical inference to determine details of the story, including the identity of the murderer. For example if the user finds a body and a gun lying beside the body then the probability that the murder weapon was the gun increases. Thus NOLIST creates the past of the story in response to the user's interactions.

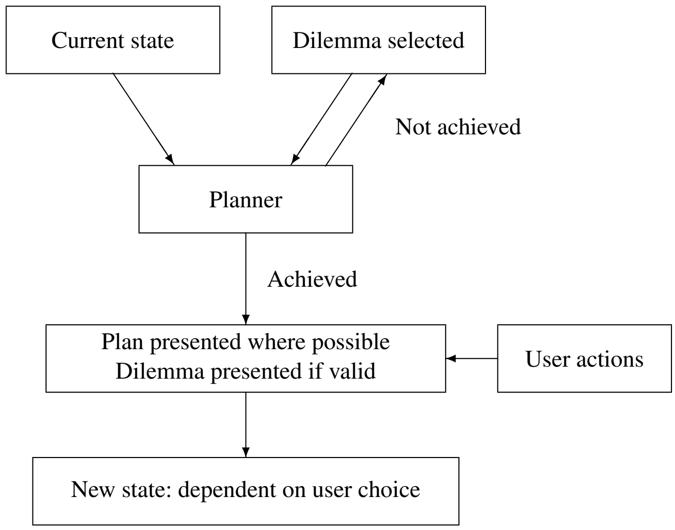


Figure 9. An overview of the GADIN system moving between states dependent on plans, dilemmas and user decisions [10]

NOLIST is highly adaptive to user interaction. However since users are likely to play games in a similar manner each time (in accordance with their player type [15]) they will probably experience a story with insignificant differences on subsequent experiences.

4.2 GADIN

The generator of adaptive dilemma-based interactive narratives (GADIN) system [10] generates narratives based on dilemmas. These represent fundamentally difficult decisions for characters within the storyworld, who can include the user. These dilemmas provide dramatic interest within the narrative.

GADIN uses planning to achieve dilemmas within the course of a story. When presented, such a plan constitutes a sub-story of the generated narrative. The user is able to freely select their own actions, which are incorporated where possible into the plan. If this is not possible then re-planning will occur. The user is able to freely make their own decisions when presented with dilemmas. A user model is employed to increase the dramatic interest of the dilemmas for the individual user [12].

Figure 9 shows an overview of GADIN's narrative generation process. Depending on the domain in which the narratives are to be generated this will continue indefinitely (for example in soaps [11]) or until a storygoal has been achieved [13]. This storygoal is dynamically selected and the user may cause it to change throughout the narrative, although it will always provide a clear and satisfactory ending.

The disadvantage of this system is in the planning bottleneck. With an increased number of actions, dilemmas and characters the planning becomes too slow for a real-time experience of the narrative.

4.3 Erasmatron

Chris Crawford's Erasmatron [19] system presents the user with a number of action options, generally relating to specific speech acts. Once the user has made their choice, the system or a character responds appropriately. This turn based action selection continous until the story is finished. This is a text-based system.

This system has a range of storyworlds available. Within these the user can become involved in the creation of a story by choosing from low-level action options. Characters have emotions and personalities. There is a drama manager which acts as "Fate".

As the user is being presented with a list of low-level action options in the Erasmatron system they will be unlikely to feel that they are free to act or to become immersed, particularly since the stories tend to return to the same choice points multiple times within the same experience.

4.4 DEFACTO

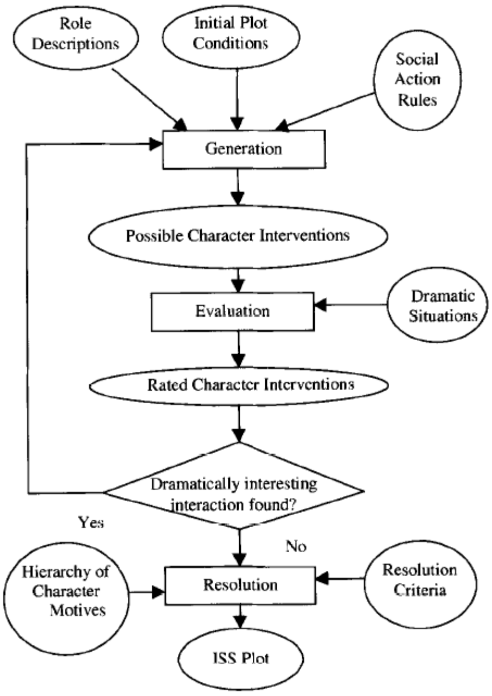


Figure 10. The plot manager architecture of DEFACTO [58]

While in the DEFACO [59] system the user becomes a character in an Ancient Greek world. They are able to specify their actions within that world and will then be shown graphically the story created. Until the graphical output is produced the user will not know the consequences of their actions.

DEFACTO allows the user to participate in stories incorporating murder, marriage, sacrifice and gods. A series of rules control the drama generation within the world, see figure 10. These stories are dynamically created in a text-based system, with user interaction. Following the interaction phase the drama is presented graphically with a twist to the user – after all of their action choices have been made, but they will not discover the outcome of their actions until the presentation phase.

The specificity of the DEFACTO system to a particular storyworld limits its applicability to other domains and the nature of its twist means that the outcome will be predictable on subsequent experiences.

4.5 OPIATE

The open ended Proppian interactive adaptive tale engine (OPIATE) [24] system creates stories based on Propp's [52] general structures for fairy tales, see figure 11. Characters other than the user have flexible roles in the story. In each state the system chooses appropriate Proppian functions using case-based planning. The story director guides the actors by giving them goals relevant to the selected function[23, 22]. The user's actions are integrated into this wherever possible.

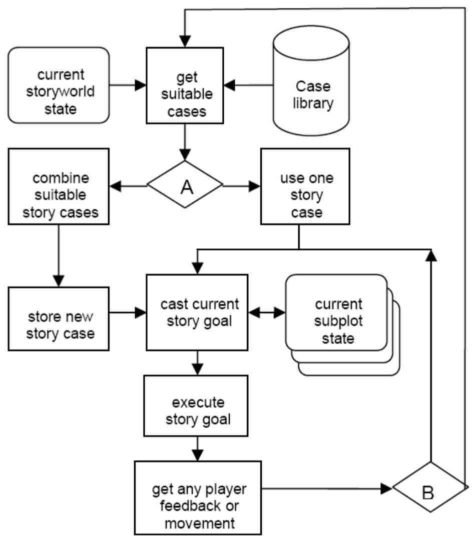


Figure 11. A flowchart showing the OPIATE planning process [24]

The story emerges from character interactions and events initiated by the story director. The engine has a gossip system which connects the characters, and spreads news and opinions about the user and their actions. The characters also communicate news of storyworld events between themselves.

The test bed for OPIATE was fairly limited with pre-scripted puzzles. It is thus unknown how it would scale, particularly given the complexity of the planning algorithm. OPIATE has a strong reliance on the generality of Propp's functions, both within the scope of a restricted fairy tale and in its potential for applicability to further domains, which is unlikely to be the case.

4.6 DED

The directed emergent drama (DED) [4] engine has a director agent that uses schemas to structure an emergent drama. There is a set of actor agents, who play characters in the unfolding drama using the schemas as a guide. Schemas are structures which contain: goals, a knowledge base; and actions for the actors and the user of the drama. The basic DED architecture can be seen in figure 12. This figure shows that all communication between the director and the actors is through schemas. The director never interacts directly with the user or actors. The user will have all the same options for interaction as the actors have. All of the interaction options available to other characters will also be possible for the user.

The characters of the drama are played by autonomous actor agents who use belief networks as their core decision mechanism.

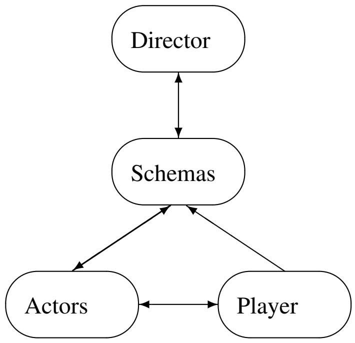


Figure 12. The DED architecture [4]

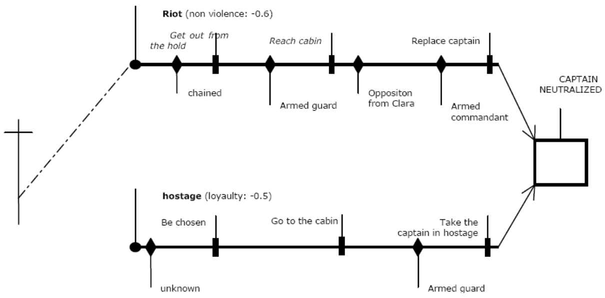


Figure 13. This figure shows the IDtension task-goal structure [63]

The actor agents use the Rational Dialog (RD) engine introduced in [2] which has now been extended and optimised for use by the actors in DED. The RD engine uses extended object-oriented Bayesian networks and Multi-Agent Influence Diagrams [30]. This is a game theoretic approach to a single agent decision problem in a multi-agent environment which provides linear growth with respect to the number of actions considered.

The schemas structure the emergent drama by giving actors goals, a knowledge base and appropriate actions to choose from. Schemas are generic structures which used by the director to structure improvisational acting, they are not small pre-authored stories. This means that an actor receives goals to accomplish and relevant actions from which they can choose. The actions are further supported by a knowledge base which the actor can use to determine appropriate actions with respect to the character's emotion, situation and personality. This facilitates the emergence of a drama in which a user can interact with the actors and storyworld freely and directly influence the unfolding drama.

The drama emerges from the interaction of the user and the actors interactions within the schemas deployed by the director. At the outset DED draws a basic plot, using the dynamic plot generating engine (DPGE) [3] to create a past for characters and their relationships. This provides a background story for the drama.

This is recent research and has jet to be fully implemented with a complete drama, set of characters and a user.

5 RELATED WORK

Not all interactive drama systems allow the user to have first-person control of a single character. Such systems are not able to achieve the requirements detailed in section 2, but the techniques should still be considered. These are discussed in sections 5.1 - 5.4. Section 5.5 briefly introduces systems which focus solely on non-interactive story generation.

5.1 IDtension

IDtension [63, 64, 62] bases its approach on narratology such as Propp's functions [52], Bremond's process, Greima's actant model and Todorov's transformations [64]. The interactive narrative is divided into three layers [64]:

- The *discourse* layer, which contains the message or theme of the story.
- The *story* layer, which gives the succession of events and character actions, following rules based on structuralism and narrative sequences.
- The *perception* layer, which determines how the narrative is presented to the user.

The IDtension system is authored by defining and scripting a set of tasks that need to be completed, in a causal order, to complete a certain goal, as shown in figure 13 when writing a novel the author will often envision a certain type of reader. Similarly, IDtension utilises a user model which contains the following criteria [64]:

- "*Ethical consistency:* The action is consistent with previous actions of the same character, with respect to the system of values."
- "*Motivational consistency:* The action is consistent with the goals of the character."
- "*Relevance:* The action is relevant according to the actions that have just been performed. This criterion corresponds to one of the Grice's maxims."
- "*Cognitive load:* The action opens or closes narrative processes, depending on the current number of opened processes and the desired number of opened processes (high at the beginning, null at the end)."
- "*Characterization:* The action helps the user to understand characters' features."
- "*Conflict:* The action either exhibits directly some conflict (like for example an incentive that is in conflict with the inciting character's values), or the action pushes the user towards a conflicting task (for example by blocking a non-conflicting task, if a conflicting task exists)."

5.2 I-Storytelling

A user of the I-Storytelling [18] system will see a graphically depicted story. They can make suggestions to characters which may or may not be followed, and can move certain key objects. This group's system equips characters with Hierarchal Task Networks (HTNs). The characters are initially positioned in random locations around the storyworld. The story is then created through the characters' interactions.

5.3 BARDS

The BARDS system uses a Heuristic Search Planner (HSP) with RTA* to plan for emotional development in the characters, rather than for actions [50, 49]. The group use an ontology created by Gustave Flaubert as the basis for the planner. Flaubert's novel, *Madame Bovary* [26], provides the test scenario. The user can use natural language to make comments which may cause other characters to react

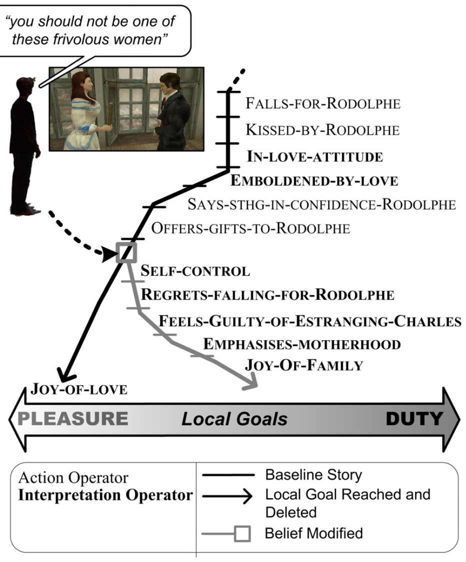


Figure 14. The influence of a NL utterance in BARDS [49]

emotionally and thus change the story, see figure 14. For instance a woman in love with a character other than her husband may feel guilt when reminded of her children. The effect will vary depending on the characters' feelings.

This is a novel approach, in which the user takes the role of an audience rather than a user, but an audience able to influence the generated story.

5.4 FAtiMA

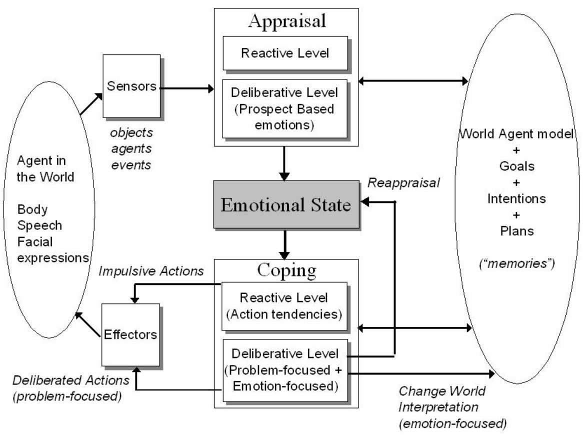


Figure 15. The FAtiMA architecture [8]

FAtiMA (FearNot! affective mind architecture), see figure 15, is a character based emergent drama system [8, 48]. The drama emerges around character actions. The test base is FearNot!, an educational game which helps children aged between 6 and 12 to learn to cope with bullying situations. The characters are reactive to the interactions of other characters, the environment and the user.

When reacting, characters use a set of emotional reaction rules, based on appraisal values such as: desirability, desirability-for-other, and praiseworthiness [8, 20]. The rules have preconditions which are compared to the current situation and the optimal match is chosen.

The characters are also goal driven. For this a STRIPS-based partial-order continuous planner is used. Characters evaluate the probability of success and the importance of the actions in accordance with whether the actions are expected to generate hope or fear. The action likely to generate the strongest emotion is chosen.

FAtiMA employs a Game Manager (GM) which uses 'narrative actions'. These affect the environment and are primarily dedicated to story management [7]. Narrative actions select episodes with respect to a plan of episodes that can be represented as a state machine. The episodes are structured as follows [7]:

- " *Name*, a unique name for the episode.
- *Set*, the set is the location in the virtual environment where the events of this episode will take place.
- *Characters*, the characters of the story, defined through a set of properties like their name, position on the set, etc
- *Preconditions*, a set of conditions that specify when is the episode eligible for selection.
- *Goals*, character goals that are communicated to the agents in this particular episode.
- *Triggers*, a condition that when satisfied will cause the execution of a set of narrative actions.
- *Finish Conditions*, a set of conditions similar to the preconditions that when satisfied indicate that the episode is finished.
- *Introduction*, a set of narrative actions introducing the episode and characters, some introductory text."

FAtiMA also applies *theory of mind* [6, 35]. This consists first of a 'double appraisal', which means that when the agent has chosen the action that would cause the strongest emotion, all of the generated actions are returned to the appraisal system, to determine which of the actions evokes the strongest emotional response from that agent. Additionally, the agent performs a 're-appraisal' by testing the actions against the emotional systems of all other characters in the scenario, to determine which action causes the strongest emotional reaction in others.

Fear Not was tested by an empirical study on 345 children, 172 male (49.9%) and 173 female (50.1%) between the ages of 8 and 11 [28]. The results showed that the children were able to empathise with the characters. There was a positive correlation between the children believing in the characters and whether they found them interesting, empathising with the characters. If the users believed that they had an high impact on the characters' behaviour then they were more likely to empathise with them. Girls were more likely than boys to feel sorry for victims that they were successful in helping.

5.5 Story generation systems

There have been various attempts to design a computer which is capable of writing stories. The major contributing systems to this research area are discussed here.

The first of these was James Meehan's Tale-Spin [42] in 1980. This system produces original fairy tales with morals. These are purely text-based and have a large number of inconsistencies. There is still dramatic interest to the stories generated by the system, all of which are set in a standard fairy tale world – with, for example, trees, rivers and fields. The system contains a large amount of background knowledge of possibilities for the world, which is created as the story is told. Characters have goals, emotions and relationships and are semi-autonomous within the game world.

Planning was used to create infinite soap opera style stories in Lebowitz's UNIVERSE [33]. In this it was necessary for the author to provide goals to the story-telling system. UNIVERSE used these goals and existing plot fragments to create a summary of a soap opera plot. System-created stereotypical characters are dynamically assigned roles in these fragments, with new characters being added if no existing character is able to take on a particular role. Character relationships are central to the interwoven storyline. The system is reliant on the reader assuming characters' motivations.

Turner's Minstrel [67] uses case-based reasoning to generate stories about knights and ladies in the days of King Arthur. The cases are existing stories and these are matched to desired stories – replacing variables where necessary – and recombined to create new stories. The system utilises its awareness of what is consistent within a world to ensure that the generated stories have this feature, and tries to present a twist at the end of each story. Both the characters and the story have goals, which are entered by the user before story generation begins.

More recent story generation research [54] has similarly focussed on generalising story segments. The system makes use of a number of short story segments, known as vignettes, which are assumed to be good. It then uses pre-defined mappings to apply these segments to new domains, where they can be joined in the generation of a new story. This technique strongly relies on the undemonstrated generality and dramatic interest of the story segments.

The Virtual Storyteller [60, 61] generates emergent stories from character interactions. Autonomous character agents have individual emotions and beliefs. The characters improvise using techniques from improvisational theatre. The stories emerge from character interactions, which are guided by a plot agent. The resulting story is then sent to a narrative agent. The story is processed by a natural language processor and then synthesised. Special rules have been developed to transform the synthesised speech to be presented, for example with the expected emphasis that a storyteller would use to provide suspense or excitement.

Although it is not strictly a story generation system, Daydreamer [45] is a system which creates daydreams. The idea is that these will be generated when a computer is idle. These daydreams will be affected by previous events and will either reflect on these – to rationalise or learn from the experiences – or create idealised alternatives to them. Experiences are at this stage input by the user. Relaxed planning is used by Daydreamer, in combination with goals and domain knowledge.

6 SUMMARY

Table 1 shows the level of satisfaction by existing systems of each of the components which are required in the creation of an interactive drama system (as discussed in section 2). This does not include story generation systems as these do not allow the user to interact within the world. The first column, (interaction with objects), shows whether or not it is possible for the user to interact with objects (which can include interacting with characters as objects) within the virtual world. In the second column, (social interaction), the ability to socially interact is identified. The dramatic interest of the drama is supported by the dramatic structures used, and the third column, (dramatic structure), identifies the structure used by each existing system. In the third column the method of presentation of the virtual world, and actions within that world, to the user is given. The final column, (fundamental difference), shows the number of fundamentally different narratives which the system is capable of creating within an application domain. It is not possible to give an exact figure for this, so an order is instead given, for example a system which is able to produce 15 fundamentally different narratives would be able to generate different narratives in the order of magnitude of 10, this is given in the table as O(10). These figures are overestimates of the potential, as exact numbers are not known.

7 FUTURE WORK

There are many possible directions in which interactive drama may develop in the next few years, or decades. It is not possible to predict the future, but in this section promising approaches and possible applications are speculated on.

Thus far there has been an strong emphasis on plot graph and planning-based approaches. Alternative decision algorithms, such as Bayesian networks, have not often been utilised. Taking advantage of alternative approaches, as is common in many other areas of Artificial Intelligence, could significantly advance interactive drama. The problems which are associated with the scaling-up of most existing systems could be successfully resolved by investigating approaches beyond the confines of symbolic planning.

Another potential way to overcome scaling-up problems would be to move from centrally generated to distributed and emergent drama. In the latter approach, decision-making is distributed amongst autonomous actors, and thus the computational bottleneck of fully centralised control is removed. To achieve a consistent story with a dramatic progression some control over the actors' actions needs to be retained. Hybrid approaches combining both perspectives on decision control, such as those suggested by [4, 6], show great promise.

The creation of interfaces for interactive drama is another area which has received very little attention thus far. Natural language appears to be the most natural choice, but the technology available today is far from perfect. Using natural language (NL) interfaces can lead to reduced enjoyment of the drama experience and frustrate the user, as has been noted in the evaluation of systems such as Façade (see for example [43]). However, future advances in NL research should make this technology more applicable to interactive drama.

An increased focus on human-computer interaction (HCI) aspects should also lead to an advance in the state of the art in system evaluation. Thorough evaluations of systems has not been easily achievable, and as a result (and as this paper shows), comparisons between interactive drama systems are not a trivial task. Nevertheless, it is to be hoped that evaluation will become more standardised and expected in this research area as it develops.

While research in interactive drama is flourishing (judging by the high number of conference and workshop submissions), very little (if any) developed technology has been incorporated into commercial games. The main reason is that this is still a relatively new research area, and that there is a question of reliability. Trust is also a major factor. Game developers are very reluctant to give control of the final product to an automatic narrative generator which cannot guarantee a consistently high story quality. In addition there has not as yet been proposed a convincing method of integrating storytelling into existing game genres, such as first-person shooters. Perhaps the best way to move forward in this is to create a whole new game genre – a method which shows great promise, judging by the attention the Façade system [41] generated both in the research and the player community.

System	Virtual world	Interaction with objects	Social interaction	Dramatic structure	Fundamental difference
Oz	Simple graphics	Yes	Some	Plot graph	O(10)
Virtual Theater Project	Text	Some	Yes	Plot graph	O(1)
Façade	Simple graphics	Some	Some	Plot graph	O(10)
IDA	Simple graphics	No	Some	Plot graph	O(1)
SASCE	None	Some	Some	Plot graph	O(10)
U-DIRECTOR	Simple graphics	Some	Some	Bayesian networks	O(1)
PaSSAGE	Neverwinter Nights graphics	Yes	No	Plot graph	O(10)
IN-TALE	Graphics	Yes	Some	Plot graph	O(10)
Mimesis	Simple graphics	Yes	No	Plot graph	O(1)
NOLIST	Text-based	Yes	Some	Bayesian networks	O(∞)
GADIN	Text-based	Some	Yes	Planning and dilemmas	O(∞)
Erasmatron	Text-based	No	Yes	Dramatic interest rules and general patterns	O(10)
DEFACTO	Text-based and simple graphics	Some	Some	Dramatic interest rules and general patterns	O(10)
OPIATE	Simple graphics	Yes	Some	Proppian structures	O(10)
DED	Second Life	Yes	Yes	Schemas and emergence	O(∞)
IDtension	Text-based	No	No	Planning and tasks	O(10)
I-Storytelling	Simple graphics	No	Some	Character HTNs	O(10)
BARDS	Virtual reality	No	Some	HSP	O(10)
FAtiMA	Simple graphics	No	Yes	Character goals and emergence	O(10)

Table 1. Summary table

This research additionally has high applicability to education, therapy and entertainment which could be investigated further.

REFERENCES

[1] A. Aarne, ‘Verzeichnis der märchentypen’, *Folklore Fellows Communications No. 3*, (1911).

[2] M. Arinbjarnar, ‘Rational dialog in interactive games’, in *proceedings of AAAI Fall Symposium on Intelligent Narrative Technologies*, Westin Arlington Gateway, Arlington, Virginia, (2007).

[3] M. Arinbjarnar, ‘Dynamic plot generation engine’, in *proceedings of the Workshop on Integrating Technologies for Interactive Stories*, Playa del Carmen, Mexico, (2008).

[4] M. Arinbjarnar and D. Kudenko, ‘Schemas in directed emergent drama’, in *proceedings of the 1^{st} Joint International Conference on Interactive Digital Storytelling ICIDS08*, Erfurt, Germany, (2008).

[5] Aristotle, *Poetics*, The Internet Classics Archive, 350 B.C.E. Translation by S. H. Butcher available at http://classics.mit.edu/Aristotle/poetics.html. Last referenced 01/10/08.

[6] R. Aylett and S. Louchart, ‘If i were you: double appraisal in affective agents’, in *Proceedings of the Autonomous Agents and Multi-agent Systems AAMAS*, (2008).

[7] R. Aylett, S. Louchart, A. Tychsen, M. Hitchens, R. Figueiredo, and C. D. Mata, ‘Managing emergent character-based narrative’, in *The Second International Conference on Intelligent Technologies for Interactive Entertainment*, Cancun, Mexico, (January 2008).

[8] Ruth Aylett, Joao Dias, and Ana Paiva, ‘An affectively driven planner for synthetic characters’, in *In proceedings of ICAPS06 International Conference on Automated Planning and Scheduling*, UK, (2006).

[9] O. Bangs, O. G. Jensen, F. V. Jensen, P. B. Andersen, and T. Kocka, ‘Non-Linear Interactive Storytelling Using Object-Oriented Bayesian Networks’, in *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, (2004).

[10] Heather Barber, *Generation of Adaptive Dilemma-based Interactive Narratives*, Ph.D. dissertation, University of York, 2009.

[11] Heather Barber and Daniel Kudenko, ‘Dynamic generation of dilemma-based interactive narratives’, in *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference*, Stanford, California, (2007).

[12] Heather Barber and Daniel Kudenko, ‘A user model for the generation of dilemma-based interactive narratives’, in *AIIDE’07 Workshop on Optimising Player Satisfaction*, Stanford, California, (2007).

[13] Heather Barber and Daniel Kudenko, ‘Generation of dilemma-based interactive narratives with a changeable story goal’, in *2nd International Conference on Intelligent Technologies for Interactive Entertainment*, Cancun, Mexico, (2008).

[14] R. Barthes. An essay. trans. richard miller, 1974.

[15] Chris Bateman and Richard Boon, *21st century game design*, Charles River Media, Hingham, Massachusetts, 2006.

[16] Joseph Bates, ‘Virtual reality, art, and entertainment’, *Presence: The Journal of Teleoperators and Virtual Environments*, **1**, (1992).

[17] Bioware. Newerwinter nights. http://nwn.bioware.com/, 2002.

[18] Marc Cavazza and Fred Charles, ‘Character-based interactive storytelling’, *IEEE Intelligent Systems*, **17**, (2002).

[19] Chris Crawford, *Chris Crawford on Interactive Storytelling*, New Riders, 2004.

[20] Joao Dias and Ana Paiva, ‘Feeling and reasoning: a computational model for emotional agents’, *Springer*, 127–140, (2005).

[21] M. Esslin, *An Anatomy of Drama*, Sphere Books, London, 1976.

[22] C. R. Fairclough, *Story Games and the OPIATE System*, Ph.D. dissertation, Department of Computer Science, University of Dublin, Trinity College, October 2004.

[23] C. R. Fairclough and P. Cunningham, ‘AI Structuralist Storytelling

In Computer Games', *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, (2004).

[24] Chris Fairclough, *Story Games and the OPIATE System*, Ph.D. dissertation, University of Dublin - Trinity College, 2004.

[25] Syd Field, *Screenplay: The Foundations of Screenwriting*, Delta Trade Paperbacks, New York, 2005.

[26] G. Flaubert and M. Cavazza, *Madame Bovary*, Paris, France, 1856.

[27] Gustav Freytag, *Technique of the Drama*, Benjamin Blom, 1863.

[28] L. Hall, S. Woods, R. Aylett, L. Newall, and A. Paiva, 'Achieving empathic engagement through affective interaction with synthetic characters', in *Proceedings of the International Conference on Affective Computing and Intelligent Interfaces*, pp. 731–738. Springer, (2005). LNCS 3784.

[29] B. Hayes-Roth, R. van Gent, and D. Huber, 'Acting in character', in *Creating Personalities for Synthetic Actors*, eds., R. Trappl and P. Petta, Springer-Verlag, Berlin, (1997).

[30] D. Koller and B. Milch, 'Multi-Agent Influence Diagrams for Representing and Solving Games', *Games and Economic Behavior*, **45**(1), 181–221, (2003). Full version of paper in IJCAI '03.

[31] Linden Lab. Second life, June 2008. http://secondlife.com/.

[32] Brenda Laurel, *Computers as Theater*, Addison-Wesley Publishing Company, 1993.

[33] Michael Lebowitz, 'Planning stories', in *Ninth Annual Conference of the Cognitive Science Society*, Seattle WA, (1987).

[34] Bethesda Softworks LLC. Fallout 3, 2008. http://secondlife.com/.

[35] S. Louchart and R. Aylett, 'Building synthetic actors for interactive dramas', in *Proceedings of the AAAI Fall Symposium on Intelligent Narrative Technologies*, pp. 63–71, (November 2007). ISBN 978-1-57735-350-8 FS-07-05.

[36] B. Magerko, *Player Modeling in the Interactive Drama Architecture*, Ph.D. dissertation, The Department of Computer Science and Engineering, University of Michigan, 2006.

[37] Brian Magerko, 'Story representation and interactive drama', in *1st Artificial Intelligence and Interactive Digital Entertainment Conference*, Marina Del Rey, California, (2005).

[38] M. Mateas, *Interactive Drama, Art, and Artificial Intelligence*, Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, December 2002. Technical Report CMU-CS-02-206.

[39] M. Mateas and A. Stern, 'Build it to understand it: Ludology meets narratoloy in game design space', in *Proceedings of the Digital Interactive Games Research Association Conference*, Vancouver B.C., (June 2005). Included in the Selected Papers volume.

[40] Michael Mateas, 'An oz-centric review of interactive drama and believable agents', Technical report, (1997).

[41] Michael Mateas and Andrew Stern, 'Façade: An experiment in building a fully-realized interactive drama', *Game Developers Conference, Game Design track*, (2003).

[42] James Richard Meehan, *The Metanovel: Writing Stories by Computer*, Garland Publishing, New York and London, 1980.

[43] David Milam, Magy Seif El-Nasr, and Ron Wakkary, 'Looking at the interactive narrative experience through the eyes of the participants', in *proceedings of the 1^{st} Joint International Conference on Interactive Digital Storytelling ICIDS08*, Erfurt, Germany, (2008).

[44] Bradford W. Mott and James C. Lester, 'U-DIRECTOR: A decision-theoretic narrative planning architecture for storytelling environments', in *Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, Hakodate, Japan, (2006).

[45] Erik T. Mueller and Michael G. Dyer, 'Towards a computational theory of human daydreaming', in *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, Irvine, California, (1985).

[46] Janet Murray, *Hamlet on the Holodeck: the future of narrative in cyberspace*, MIT Press, Cambridge, Massachusetts, 1998.

[47] Mark J. Nelson, David L. Roberts, Charles L. Isbell, and Michael Mateas, 'Reinforcement learning for declarative optimization-based drama management', in *Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, Hakodate, Japan, (2006).

[48] Ana Paiva, Joao Dias, Daniel Sobral, Ruth Aylett, Polly Sobreperez, Sarah Woods, Carsten Zoll, and Lynne Hall, 'Caring for agents and agents that care: Building emphatic relations with synthetic agents', in *Proceedings of AAMAS 2004*, (2004).

[49] D. Pizzi and M. Cavazza, 'Affective storytelling based on characters' feelings', in *Proceedings of the AAAI Fall Symposium on Intelligent Narrative Technologies*, Arlington, Virginia, (November 2007).

[50] D. Pizzi, F. Charles, J. Lugrin, and M. Cavazza, 'Interactive storytelling with literary feelings', in *Proceedings of the Second International Conference on Affective Computing and Intelligent Interaction (ACII)*, Lisbon, Portugal, (September 2007).

[51] G. Prince, *A Dictionary of Narratology*, University of Nebraska Press, 2003.

[52] Vladimir Propp, *Morphology of the Folktale*, Austin: University of Texas Press, 1968. Translation by Laurence Scott.

[53] M. O. Riedl, C. Saretto, and R. M. Young, 'Managing interaction between users and agents in a multi-agent storytelling environment', in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, (2003).

[54] Mark Riedl and Carlos Leon, 'Toward vignette-based story generation for drama management systems', in *Proceedings of the INTETAIN'08 Workshop on Integrating Technologies for Interactive Stories*, Playa del Carmen, Mexico, (2008).

[55] Mark Riedl and Andrew Stern, 'Believable agents and intelligent story adaptation for interactive storytelling', in *TIDSE'06*, Darmstadt, Germany, (2006).

[56] David L. Roberts and Charles L. Isbell, 'A survey and qualitative analysis of recent advances in drama management', *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning*, **4**, (2008).

[57] Marie Laure Ryan, 'Narrative and the split condition of digital textuality', *Dichtung Digital*, **5**, (2005).

[58] N. M. Sgouros, 'Dynamic generation, management and resolution of interactive plots', *Artif. Intell.*, **107**(1), 29–62, (1999).

[59] Nikitas M. Sgouros, 'Dynamic, user-centered resolution in interactive stories', *International Joint Conference on Artificial Intelligence*, **2**, (1997).

[60] I. Swartjes and J. Vromen, 'Emergent story generation: Lessons from improvisational theater', in *Proceedings of the AAAI Fall Symposium on Intelligent Narrative Technologies*, Arlington, Virginia, (November 2007).

[61] I. Swartjes, J. Vromen, and Niels Bloom, 'Narrative inspiration: Using case based problem solving to support emergent story generation', in *Proceedings of the International Joint Workshop on Computational Creativity*, Goldsmiths, University of London, (June 2007).

[62] N. Szilas, 'Stepping into the interactive drama', in *Proceedings of the International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, pp. 14–25, Darmstadt, Germany, (June 2004). SPRINGER. S. Gbel et al. (Eds) LNCS 3105.

[63] N. Szilas, O. Marty, and J. Rty, 'Authoring highly generative interactive drama', in *Proceedings of the International Conference on Virtual Storytelling*, pp. 20–21, Toulouse, France, (November 2003).

[64] Nicolas Szilas, 'IDtension: A narrative engine for interactive drama', in *1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, Darmstadt, Germany, (2003).

[65] David Thue, Vadim Bulitko, Marcia Spetch, and Eric Wasylishen, 'Interactive storytelling: A player modelling approach', in *AIIDE'07*, Stanford, California, (2007).

[66] T. Todorov, *The Poetics of Prose*, chapter The Typology of Detective Fiction, 42–53, Basil Blackwell, Oxford, 1977.

[67] Scott R. Turner, *The Creative Process: A computer model of storytelling and creativity*, Lawrence Erlbaum Associates, 1994.

[68] R. M. Young and M. Riedl, 'Towards an architecture for intelligent control of narrative in interactive virtual worlds', in *Proceedings of the International Conference on Intelligent User Interfaces*, (January 2003).

[69] R. Michael Young, Mark O. Riedl, Mark Branly, Arnav Jhala, R. J. Martin, and C. J. Saretto, 'An architecture for integrating plan-based behavior generation with interactive game environments', *Journal of Game Development*, **1**, (2004).

Using Neural Networks for Strategy Selection in Real-Time Strategy Games

Thomas Randall[1], Peter Cowling[1], Roderick Baker[1] and Ping Jiang[1]

Abstract. Video games continue to grow in importance as a platform for Artificial Intelligence (AI) research since they offer a rich virtual environment without the noise present in the real world. In this paper, a simulated ship combat game is used as an environment for evolving neural network controlled ship combat strategies. Domain knowledge is used as input to the Artificial Neural Networks (ANNs) through scripts that run in parallel and feed their decisions to the ANNs. The ANNs then interpret these scripts and decide what strategy to perform. The results are compared to ANNs that have no such knowledge and tested to see how well the ANNs generalise.

1 INTRODUCTION

Computer games are increasingly becoming a platform for academic Artificial Intelligence (AI) research as they offer virtual environments without the inherent noise present in the real world, supporting the prediction of Laird and van Lent [1] that computer games could become a killer application for AI research.

Artificial Neural Networks (ANNs), a technique that uses a computational analogy with the human brain, is an active area of research with a lot of work being carried out in computer games [2 – 9]. Research shows that not only can ANNs be used in games, they can be the focus of the gameplay itself, as in the freely available NERO[i] combat simulation game [3, 6 - 8].

Evolutionary Algorithms (EAs) are also an active area of research with a lot of work carried out in games [2 – 14]. EAs use an analogy with biological evolutionary process. Spronck uses Genetic Algorithms (GAs) to evolve script selection in a process called Dynamic Scripting [13, 14]. Dynamic scripting uses a population of scripts to control agents in a Real-Time Strategy (RTS) game. The scripts are initially picked at random and after each encounter each script element is allocated a fitness based on the performance of evolved scripts which use that element. This fitness is used to alter the probability of the scripts being selected in the next encounter. Spronck also describes a model for creating reliable adaptive game intelligence for use in commercial videogames [15] which Dynmaic Scripting adheres to.

One common use of EAs is to evolve ANNs. Stanley and coworkers created a technique called **N**euro**E**volution of **A**ugmenting **T**opologies (NEAT) that evolved both network weights and network topology at the same time [3 – 8]. NEAT uses historical markings of topology changes, which allows NEAT to use operators to crossover different ANN topologies. NEAT starts off with a simple network topology adding connections and nodes through mutations in a process called complexification. These mutations often reduce the fitness but NEAT protects new innovation through speciation. Networks are assessed to see how similar they are to each other; if similar, they are placed into the same species, otherwise a new species is created. Individual networks do not compete against each other in NEAT; instead, species compete against each other. NEAT is particularly well suited to problems where it is difficult to determine appropriate network topologies, such as in the NERO game [3, 6 - 8] mentioned previously.

Incremental evolution is a process of increasing the complexity of the problem as the evolutionary process progresses [16, 17]. This can be done in different ways from altering the fitness measure to altering the environment that evolution takes place in. The aim is to use domain knowledge to guide evolution in the right direction for problems that are too complicated or impossible to evolve solutions directly.

The use and acquisition of domain knowledge is an active area of research, in particular Louis uses a **C**ase-**I**njected **G**enetic **A**lgo**R**ithm (CIGAR) to learn how to play strategy games [9, 10]. CIGAR injects chromosomes which encode previously successful strategies in the current population of strategies. CIGAR effectively learns from experience: as it plays a game, it keeps track of good strategies which might be used for future problems. CIGAR can be viewed as an incremental evolution technique as it uses prior knowledge to drive the evolutionary process.

In this paper we are controlling units in the ship combat game called DEFSIM, written in C#.NET, that simulates ship combat using the rules of the commercial video game DEFCON[ii], by Introversion Software[iii]. We aim to evolve intelligent behaviours for individual units in an RTS game in order to create intelligent players for the game overall. We use domain knowledge of strategies as input to ANNs to merge or produce a strategy based upon the input strategies. Most related research directly evolves ANNs that receive input from the game state [3, 6 – 8] or evolves mechanisms to select hand-written scripts in a specific order to create dynamic players [13, 14]. In this paper we pre-trained ANNs to carry out specific tasks and used them as scripts. Hence we can use evolution to combine a mixture of scripts, ANNs, and other techniques that aid the creation of intelligent-looking unit behaviours.

DEFSIM simulates DEFCON ship combat, whilst allowing us complete control over the source code, allowing us to tweak features as required. This is much more straightforward than directly using the DEFCON code, although this may shortly

[1] School of Computing, Informatics and Media, Univ. of Bradford, Bradford, BD7 1DP, UK. Email: {t.w.g.randall, p.i.cowling, r.j.s.baker, p.jiang}@brad.ac.uk.

change with [23]. DEFSIM also allows us to run experiments thousands of times more quickly than is possible than most commercial games available as it removes features unnecessary for AI research such as graphics rendering. This speed advantage offers a suitable environment to train/evolve ANNs. DEFSIM also allows for the addition of features as needed and allows us to run simplified rules of the game. As games become increasingly complex environments offering realistic physics, graphics and interactions the presence of noise increases, increasing the difficulty of using machine learning approaches. The use of a simulator helps remove some of this noise allowing us to concentrate on research. Once solutions have been found they may then be imported into the original game.

Ship combat works in a probabilistic manner in DEFSIM; combat between two battleships yields a 50% chance for either ship to be successful in battle. However, if a player has a local firepower advantage against an opponent in a battle, then these odds are changed in favour of the opponent with more ships [18, 19]. Learning to locally outnumber the opponent is a challenge for machine learning because the units must be made to cooperate and coordinate with each other, taking into account the number of opponents and allies in the nearby vicinity and moving strategically.

Evolution of solutions with a complex fitness function can often be too slow or render it impossible to find a solution without some additional guidance and domain knowledge. We investigate evolution of ANNs to interpret different scripts running in parallel and produce a strategy based upon them.

In section 2, we will discuss our experimental design and the different terrains we will use for evolution. We also describe the behaviour of the scripted AI against which the ANN players are tested. In section 3 we present our results and discuss our findings. In section 4 we present conclusions and plans for future work.

2 EXPERIMENT DESIGN

We use a population of 20 ANNs to select from different strategies to control a fleet of battleships in a game of DEFSIM against a team of scripted opponents. In DEFSIM, each ship has a 10% chance of hitting the closest opponent that is in the ship's attack radius (set to the same distance as the view radius) per frame. As the ANNs have to learn to approach the opponent ships and the opponent ships start outside the view radius, we have made it so that the ANNs have complete information (i.e. they have prior information as to the location of the opponent ships).

In this work we use SharpNEAT [20 - 22], a C#.NET implementation of NEAT [3 – 8] to evolve both the topology and weights of a population of ANNs. SharpNEAT also adds another process that is not available in the original NEAT design called pruning [22]. Pruning simplifies network topologies if the performance of a species in NEAT stagnates for a given period (i.e. no improvement in the species best fitness), which in these experiments is 200 generations.

The ANNs are evaluated on five different terrains with different strategies needed to win. We use five terrains to get a range of environments as units in an RTS game would need to operate on multiple terrains that the developers had not created, especially if the game allows user created terrains. These terrains are shown in Figure 4 to Figure 8 where ships marked with an 'E' are enemy (scripted) ships, ships marked with an 'F' are friendly (ANN controlled) ships and flags marked with a 'T' are targets. In these games, targets represent areas of importance, i.e. opponent bases. Damage inflicted on targets is worth significantly more than damage inflicted on opponent ships. The score for a game is shown in Equation (1). In these experiments the scripted players are playing a defensive role and the ANNs are playing an offensive role.

$$F_1 = \alpha \sum E_i + \beta \sum T_j \quad (1)$$

Where α and β are constants set to 1 and 5 respectively. E_i is the probability of enemy ship i being alive and T_j is the probability of enemy target j being alive.

```
foreach ship in myShips
  target = mostVulnerableTarget()
  if distance(ship, target) <
       (viewDistance(target) / 2) then
    ship.moveTowards(target)
  else
    if isInTargetViewDistance(opponent,
        target) then
      attacker =
        getStrongestAttacker(target)
      moveShipBetweenTargetAndAttacker(
        ship, target, attacker)
    else
      circleShipAroundTarget(ship, target)
    end if
  end if
end foreach

function mostVulnerableTarget()
  vulnerableTarget = nil
  targetDanger = 0
  foreach target in myTargets
    maxDistance = viewDistance(target) * 4
    danger = 0
    foreach op in visibleOpponents(target)
      dist = distance(op, target)
      health = getHealth(op)
      danger = danger +
        max((maxDistance - dist) *
        health / maxDistance, 0)
    end foreach
    danger = (danger /
    amountOfVisibleOpponents(target)) *
      getHealth(target))

    if danger > targetDanger or
        vulnerableTarget = nil then
      targetDanger = danger
      vulnerableTarget = target
    end if
  end foreach

  return mostVulnerableTarget
end function
```

Figure 1. Pseudocode of scripted ships.

All of the terrains in these experiments use the same scripted opponents that try to defend the targets whilst the ANNs try to do as much damage to the targets as possible. The idea is to simulate a generic game strategy where the ANN-controlled player tries to attack the scripted player's base which has targets representing different building/important points that the opponent needs to defend. The targets are incapable of fighting, thus they do not inflict damage against the ANN controlled ships. The scripted player's pseudocode is shown in Figure 1. Note that to defend a target a ship stands between the target and the attacker. This is because in DEFSIM, combat is performed automatically and the attacker attacks the closest opponent with a 10% chance of hitting. Also note that health represents the probability of the ship being alive in the range [0 1] and uses Equation (2) to calculate damage inflicted As a ship receives damage, the amount of damage it inflicts in a fight is reduced. This implicitly encourages the ANNs to minimize damage received as the fitness measure used in Equation (1) will be small when the ANN controlled ships take heavy damage.

$$H_t = H_{t-1}(1 - PA) \quad (2)$$

Where H_t is the defending ship's new health, H_{t-1} is the defending ship's current health, P is the probability of the attacking ship hitting the defending ship (in these experiments 0.1), A is the attacking ship's current health.

The ANNs are evolved to select/merge different strategies to maximize their fitness using Equation (1) as the fitness measure. In these experiments, four different strategies are used as input for the ANNs. These are:

1. Attack opponent ships.
2. Attack opponent targets.
3. Group together with friendly ships.
4. Have more life than the opponent.

These four strategies are performed by pre-evolved ANNs. We leave the evolution of these behaviours outside the scope of this paper, but note that the ANNs could be replaced by hand-coded equivalent scripts with no fundamental difference to our results. We refer to these strategies as "ANN scripts" in the subsequent text Strategy 4 represents a complex strategy where the ANNs will try to maximise their health whilst minimizing the opponents' health, which could include behaviours such running away when the ANNs health is small, and pressing an advantage with superior firepower in a particular location. Each ship is controlled individually by an identical ANN. All the ANN scripts output what a ship should do in terms of vertical (up and down) and horizontal (left and right) movement. The ANNs that use domain knowledge (via ANN scripts) do not receive direct input about the world state, instead receiving "advice" from the scripted behaviours. The inputs and outputs to the script interpreting ANNs are shown in Table 1 and Figure 2. The ANN scripts output an action, i.e. the output of the scripts is the degree of up/down and left/right movement.

This work has parallels with Dynamic Scripting [13, 14] and CIGAR [9, 10] as it uses domain knowledge in the evolution. The work presented here is different to Dynamic Scripting in that ANNs are evolved to interpret the scripts whereas Dynamic Scripting optimizes selection and ordering of hand-written scripts. CIGAR extracts domain knowledge from playing a game or watching someone play a game to create cases which are injected at suitable points into the population of an evolutionary algorithm. Here, we use knowledge in the form of ANN scripts. All the scripts fed into a master NEAT network in parallel and the ANNs are evolved to relate the scripts' actions to final ship actions.

The ANNs will be compared to ANNs evolved without domain knowledge. We do this to see how the domain knowledge works against ANNs evolved directly to play the game. The inputs and outputs for the ANNs evolved without any domain knowledge are shown in Table 2 and Figure 3.The sensors used in ANNs without domain knowledge (also shown in Table 2 and Figure 3) use Equation (3) as their input signal as represented in Figure 9.

In these experiments, all units have complete information about ship locations. Ship/target density in each direction is measured using

$$F_3 = \sum \frac{D - d_i}{D} \quad (3)$$

Where D is the maximum possible distance in a given direction (Up/Down and Left/Right) and d_i is the distance from the ANN controlled unit to a unit i. The sum is over all other friendly/enemy ships.

The distances of the ships are summed, as in [6, 7]. Summing the inputs allows for scalability to an unknown number of ships.

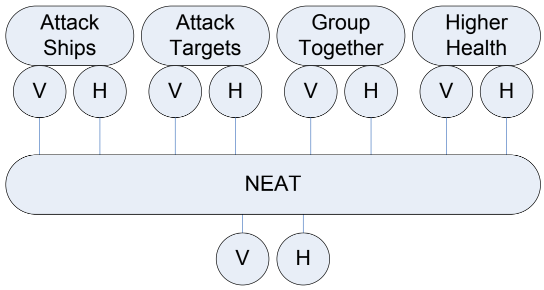


Figure 2. Knowledge base ANNs receive information from scripts. V represents vertical movements and H represents horizontal movements.

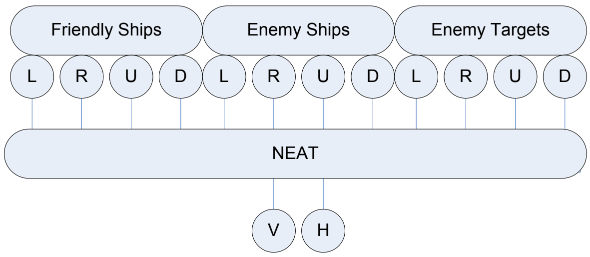


Figure 3. Game state ANN receives input about units in a game. L, R, U, D, represents the up, down left and right sensors. V and H represent vertical and horizontal movements.

Input	Description
0	Bias
1	Vertical movement from Script 1
2	Horizontal movement from Script 1
3	Vertical movement from Script 2
4	Horizontal movement from Script 2
5	Vertical movement from Script 3
6	Horizontal movement from Script 3
7	Vertical movement from Script 4
8	Horizontal movement from Script 4
Outputs	
0	Horizontal velocity
1	Vertical velocity

Table 1. List of Inputs and Output for the ANNs Recieving Input From Previously Trained ANNs

Input	Description
0	Bias
1	Left friendly ship Sensor
2	Up friendly ship Sensor
3	Right friendly ship Sensor
4	Down friendly ship Sensor
5	Left enemy ship Sensor
6	Up enemy ship Sensor
7	Right enemy ship Sensor
8	Down enemy ship Sensor
9	Left enemy target Sensor
10	Up enemy target Sensor
11	Right enemy target Sensor
12	Down enemy target Sensor
Outputs	
0	Horizontal velocity component
1	Vertical velocity component

Table 2. List of Inputs and Outputs for the ANNs Taking into Account Friendly Ships, Enemy Ships and Enemy Targets

Having a ship above and to the left of an ANN controlled ship will bring input to two sensors (the up and left sensors). It is possible for the ANNs to determine the direction of opponent ships when the inputs are compared. Two inputs are used for each direction to indicate density of hostile and friendly units in a particular direction. In both experiments, the networks are cloned so that each ship uses its own separate copy of the network. This is necessary since NEAT creates recurrent networks so each ship needs its own copy or recurrent input could potentially pollute the ANN "memory". It also allows for scalability as different terrains have different amounts of ships in the game [3 – 8], and makes evolution easier.

In these experiments ANNs will be evolved against four of the five terrains shown in Figure 4 to Figure 8 playing 10 games on each of the four terrains per generation (in the sequence of 10 games on the first terrain then 10 on the next terrain and so on). The ANNs are also tested during evolution in the terrain they were not evolved against to see how well they generalise (with their progress on the test terrain not being considered during evolution). Both sets of experiments use Equation (1) as the fitness measure. The different experiments are:

- *Experiment One:* ANNs evolved using Figures 5, 6, 7, 8 and tested on Figure 4.
- *Experiment Two:* ANNs evolved using Figures 4, 6, 7, 8 and tested on Figure 5.
- *Experiment Three:* ANNs evolved using Figures 4, 5, 7, 8 and tested on Figure 6.
- *Experiment Four:* ANNs evolved using Figures 4, 5, 6, 8 and tested on Figure 7.
- *Experiment Five:* ANNs evolved using Figures 4, 5, 6, 7 and tested on Figure 8.

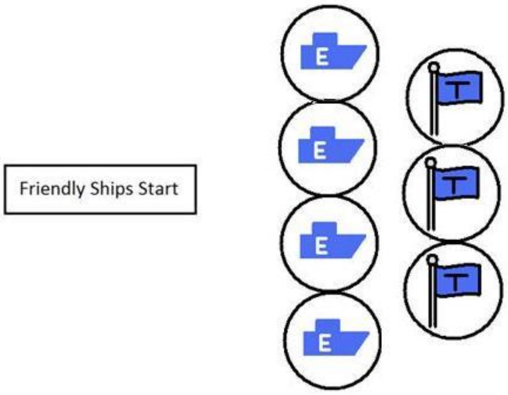


Figure 4. The ANN controlled ships start to the left with targets to the right defended by enemy ships.

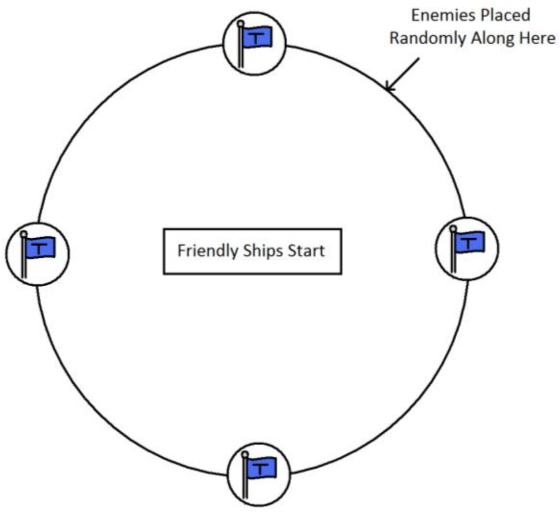


Figure 5. The ANN controlled ships start in random positions in the middle with four targets around them. The Enemy ships start in random positions on the circle that the targets sit on.

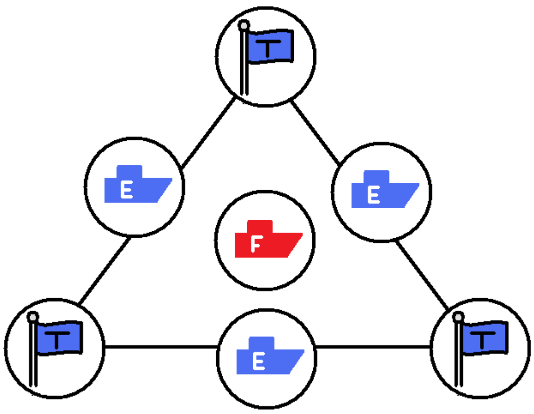


Figure 6. Three ANN controlled ships start in the centre (diagram only shows one) and three targets in a triangle around the ANN controlled ships. The enemy ships start on the lines between the targets.

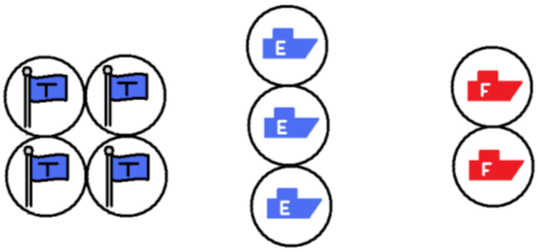


Figure 7. The targets are grouped together so that it is easier for the enemy ships to defend them.

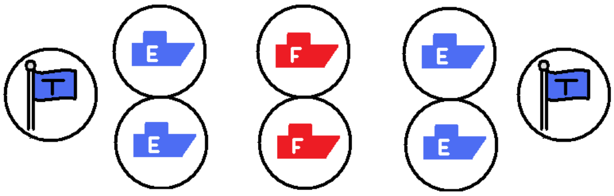


Figure 8. The ANN controlled ships start in the centre with targets at either side defended by enemy ships.

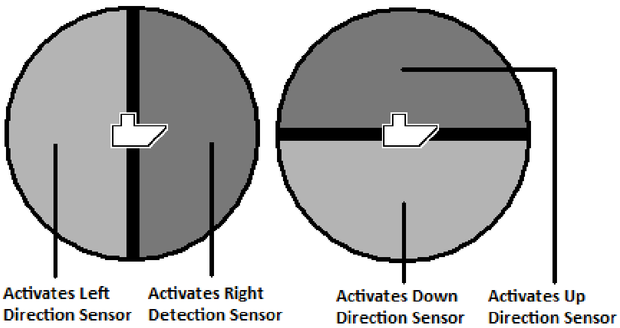


Figure 9. The zones that cause network inputs to the directional network. The circle around the ship is the ship's view radius.

3 RESULTS

Figure 10 to Figure 14 show the results of evolution for 500 generations, using Equation (1) as the fitness measure, for experiments one to five respectively. The default SharpNEAT parameters [20] were used for all experiments with the exception of the initial connection probability (the probability that a connection exists between two nodes for the initial starting population) which was set to 0.4 so that the initial population started with some connections rather than starting with few or no connections. As a small population (of 20) was used, many generations could occur with a lower initial connection probability before enough connections were made for the ANNs to perform well. All results show the 5th best genome of the population and are averaged over 10 runs of 10 games per evaluation with a population size of 20. Over a good deal of empirical investigation we have found the 5th best to be a robust statistic as it gives a good description of behaviours learnt whilst removing most of the chance inherent in experimental noise. We do not use the mean which includes 'lucky' and 'unlucky' outliers. Also note that the different terrains have different amounts of units and consequently the different results are not to the same scale as it is possible to reach a higher fitness if there are more opponents (although not necessarily easier).

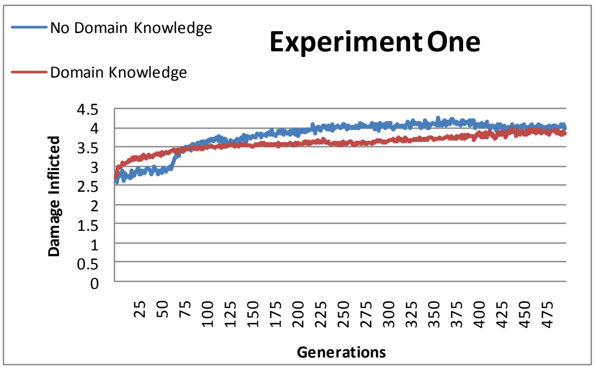


Figure 10. Experiment one evolved using Equation (1) and against Figure 5, Figure 6, Figure 7 & Figure 8.

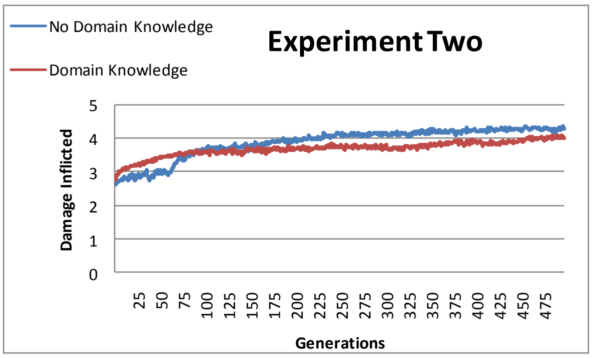


Figure 11. Experiment two evolved using Equation (1) and against Figure 4, Figure 6, Figure 7 & Figure 8.

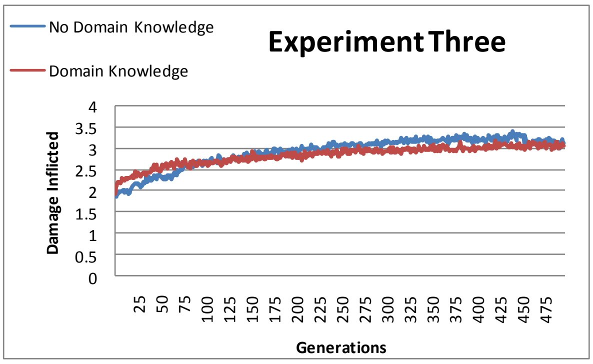


Figure 12. Experiment three evolved using Equation (1) and against Figure 4, Figure 5, Figure 7 & Figure 8.

It is clear from these results that generally ANNs evolved without domain knowledge, but have knowledge of the world state, produce a higher fitness than ANNs with domain knowledge (and no access to raw state information). However, ANNs that use domain knowledge evolve quicker and produce better initial results in early iterations. This happens for all experiments except experiment five (Figure 14) where both perform equally. ANNs with domain knowledge appear to evolve faster because they already have strategies that tell them how to play the game.

Once the ANNs with knowledge of the world state learns an initial strategy, they start producing good results that perform better than ANNs that use domain knowledge, presumably through optimisation of their strategy, something which is harder for the ANNs with domain knowledge because they do not have raw information about the game state and can only optimise the strategies from their knowledge base. We can also see that evolution had not finished at the end of the experiments. In all the experiments, it takes the method without domain knowledge a while to find an initial plausible strategy (so that domain knowledge has value when CPU time may be limited). Absence of a plausible initial strategy can often hinder evolution as is the reason for much research into incremental evolution [16, 17]. In order to perform well, the ANNs need to evolve a general purpose strategy that works across multiple terrains. This would also explain the sudden jump in fitness for ANNs that do not use domain knowledge in experiments 1-4 (Figure 10 to Figure 13) as the ANNs find an appropriate initial strategy.

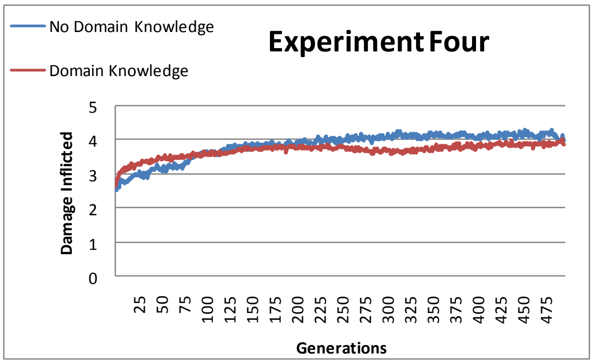


Figure 13. Experiment four evolved using Equation (1) and against Figure 4, Figure 5, Figure 6 & Figure 8.

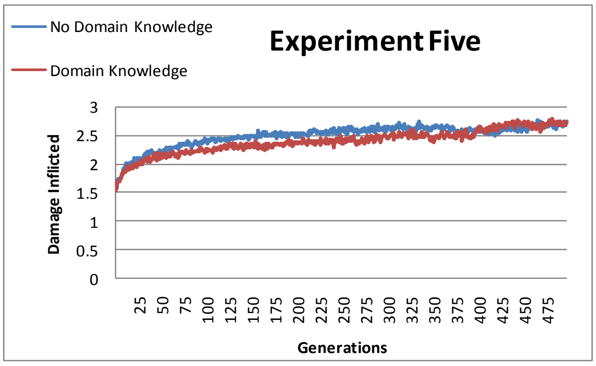


Figure 14. Experiment five evolved using Equation (1) and against Figure 4, Figure 5, Figure 6 & Figure 7.

During evolution the ANNs were also tested against an unseen terrain (although the results of this were not taken into account during evolution) to see how well they generalize on unseen test terrains. Results are shown in Figure 15 to Figure 19. The ANNs were not evolved on the unseen terrains, instead, they were tested during their evolution on the training terrains to investigate their performance changes.

Figure 15 shows the results for experiment one on the unseen terrain shown in Figure 4 using (1) as the fitness measure. It is clear from these results that once again ANNs that domain knowledge starts off slightly better but is eventually overtaken by ANNs that do not use domain knowledge, but have more complete knowledge of the world state. Towards the end of the evolution cycle, the ANNs without domain knowledge start to decrease in fitness whilst ANNs with domain knowledge start to improve. This could be down to strategies learnt throughout evolution having a negative effect on this terrain as the scripted opponents defending the targets better against the strategies used. Note that whilst ANNs with domain knowledge start to decrease in fitness towards the end of evolution, they still perform significantly better than the initial starting fitness. As evolution is ended after 500 generations it is not clear if evolution of ANNs that use domain knowledge would have continued improving but it strengthens our previous hypothesis that if more time was given, the ANNs that use domain knowledge would start to outperform ANNs that do not.

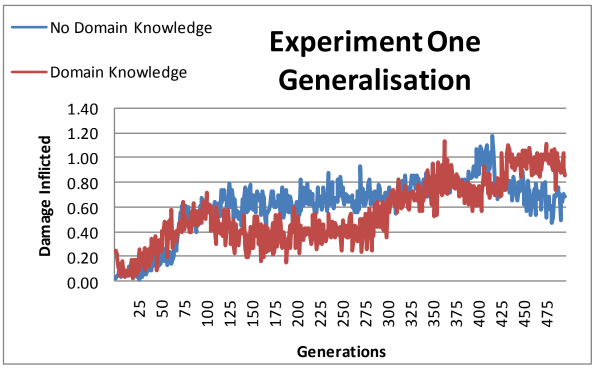


Figure 15. Experiment one generalizing on Figure 4 using Equation (1) as the fitness measure.

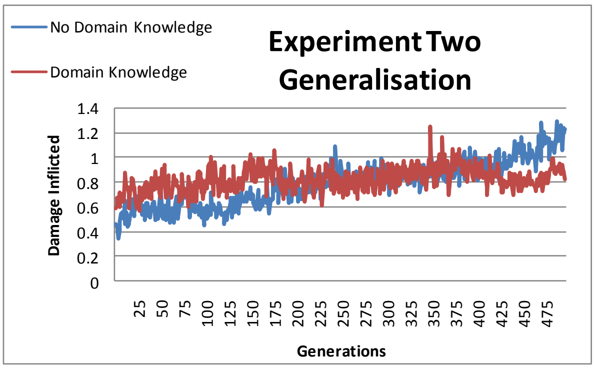


Figure 16. Experiment two generalizing on Figure 5 using Equation (1) as the fitness measure.

Figure 16 shows the results of experiment two's of generalization on the unseen terrain shown in Figure 5. It is interesting to note that once again the use of domain knowledge begins by producing better generalisation abilities although towards the end of the evolution cycle this tails off. ANNs that utilise domain knowledge does not really improve during the evolution cycle indicating that strategies evolved for training terrains do not really apply to the terrain shown in Figure 5. Figure 17 to Figure 19 show the remaining results for experiment three to five respectively at their ability to generalise. It is clear that the ability to generalise does not improve much during evolution for both types of ANNs with the differences not being statistically significant. The lack of improvement could indicate that the strategies learnt during evolution do not apply to the unseen terrains. It could also indicate that the strategies learnt could potentially be terrain specific. The fact that ANNs without domain knowledge performs well on unseen terrains agrees with our previous hypothesis that the ANNs have learnt a general purpose strategy rather than a specific strategy for a terrain.

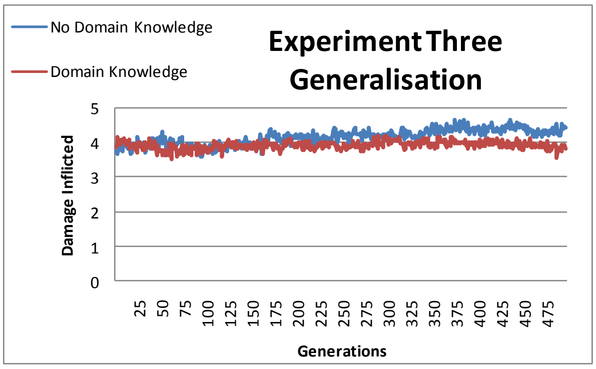


Figure 17. Experiment three generalizing on Figure 6 using Equation (1) as the fitness measure.

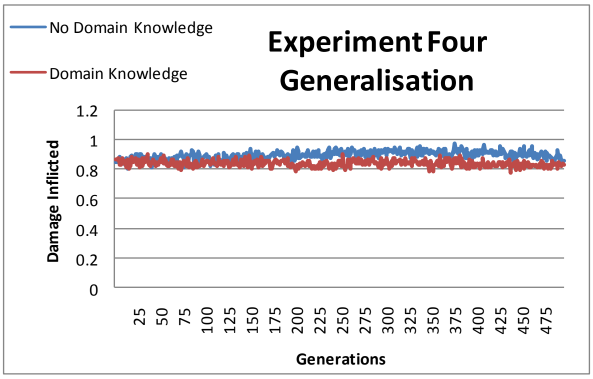


Figure 18. Experiment four generalizing on Figure 7 using Equation (1) as the fitness measure.

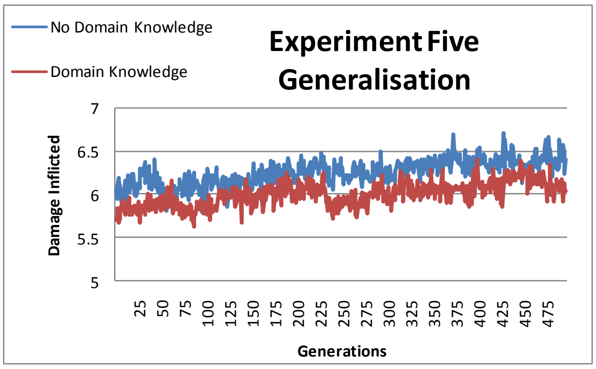


Figure 19. Experiment five generalizing on Figure 8 using Equation (1) as the fitness measure.

4 CONCLUSION & FURTHER WORK

We created DEFSIM, a C#.NET implementation of DEFCON in order to successfully perform the experiments presented here. This allows us to create many game scenarios for testing. We have shown that ANNs can be used to select/merge different scripted strategies to produce an effective player.

Whilst Dynamic Scripting can be used to select appropriate scripts for a given scenario, we have shown that ANNs can be used to “merge” different strategies that are running in parallel. In a sense, we have used an ANN in place of a Finite-State Machine, which is normally used to make a “hard” selection of which scripts to run, to make a “soft” selection via the network. The ANNs that used the scripts evolved good players quickly on terrains they were trained for, often in less than 60 generations. With a population of 20 like in these experiments, this does not take long to evolve. On a P4 1GHz processor, it took 1 CPU day to evolve 500 generations for one run of the experiment; the vast majority of this time was taken up by the ANNs playing DEFSIM.

Generally, we observed that ANNs with domain knowledge perform better than ANNs without domain knowledge at the start of evolution, but after many iterations the players without domain knowledge (but with full state information) produce a higher fitness. The players with domain knowledge only had information from the ANN scripts and did not receive information about the game state. Another problem is that the ANNs could only ever come up with strategies that are in the knowledge base, in this case, they could only ever use one of the four strategies or a mixture of them. The ANN that does not use domain knowledge could potentially be using strategies that we, the writers, have not thought of. With all these limitation the domain knowledge based ANNs continued to evolve and, if more time was given, could potentially out perform ANNs without the knowledge, presumably through novel, unforeseen, combinations of these strategies.

For future work we intend to consider ANN players which have information about the game state as well as domain knowledge via ANN scripts to see if it aids the evolution process. We also intend to increase the rule base for the scripts selection to allow for more complex strategies. We also intend to improve the DEFSIM environment to allow new units and rules to further enrich the space of possible strategies.

ACKNOWLEDGEMENTS

This research was funded by the Engineering and Physical Science Research Council (EPSRC). We gratefully acknowledge the contribution of Introversion Software Ltd who donated the DEFCON source code for use in this work. We would also like to thank Stephen Remde and Colin Ward for proofreading.

REFERENCES

[1] Laird, J.E., and van Lent, M. 2000. Human-level AI's Killer Application: Interactive Computer Games. *AAAI Fall Symposium Technical Report,* 80-97. North Falmouth, Massachusetts.

[2] Bryant, B.D., and Miikkulainen, R 2006. Evolving Stochastic Controller Networks for Intelligent Game Agents. *Proceedings of the*

2006 Congress on Evolutionary Computation (CEC 2006), 3752-3759.

[3] D'Silva, T.; Janik, R.; Chrien, M.; Stanley, K. O.; and Miikkulainen, R. 2005. Retaining Learned Behavior During Real-Time Neuroevolution. *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2005).*

[4] Stanley, K. O., and Miikkulainen, R. 2002. Efficient evolution of neural network topologies. In *Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02*, 1757-1762. CEC. IEEE Computer Society, Washington, DC.

[5] Stanley, K. O. and Miikkulainen, R. 2004. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*.

[6] Stanley, K. O.; Bryant, B.D.; and Miikkulainen, R. 2005. Evolving Neural Network Agents in the NERO Video Game. *Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG'05),* 182-189. Piscataway, NJ: IEEE Press.

[7] Stanley, K. O.; Bryant, B.D.; and Miikkulainen, 2005 Real-time neuroevolution in the NERO video game. *IEEE Trans. Evolutionary Computation* 9(6): 653-668 (2005)

[8] Stanley, K.O.; Bryant, B.D.; Karpov, I.; and Miikkulainen, R. 2006. Real-Time Evolution of Neural Networks in the NERO Video Game. *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06),* 1671-1674. Menlo Park, CA: AAAI Press.

[9] Louis S.J., and Miles, C. 2005. Playing to learn: Case-injected genetic algorithms for learning to play computer games. *IEEE Transactions on Evolutionary Computation. v9 i6. 2005,* 669-681.

[10] Louis S.J., and Miles, C. 2005. Combining Case-Based Memory with Genetic Algorithm Search for Competent Game AI. *ICCBR Workshops 2005,* 193-205.

[11] Miles, C., and Louis S.J. 2006a. Co-evolving real-time strategy game playing influence map trees with genetic algorithms. In *Proceedings of the Congress on Evolutionary Computation*, Vancouver, Canada, 2006. IEEE.

[12] Miles, C.; Quiroz, J.; Leigh, R.; and Louis, S.J. 2006b. Co-evolving influence map tree based strategy game players. In *Proceedings of the 2006 IEEE Symposium on Computational Intelligence in Games*, IEEE Press, 2007.

[13] Spronck, P.; Sprinkhuizen-Kuyper, I.; and Postam, E. Enhancing the Performance of Dynamic Scripting in Computer Games, *Proceedings of the 4th International Conference on Entertainment Computing* (ICEC 2004), 2004.

[14] Spronck, P.; Ponsen, M.; Sprinkhuizen-Kuyper, I.; and Postma, E. 2006. Adaptive Game AI with Dynamic Scripting. *Machine Learning, Vol. 63, No. 3,* 217-248.

[15] Spronck, P. 2005. A model for reliable adaptive game intelligence. *In IJCAI-05 Workshop on Reasoning, Representation, and Learning in Computer Games*, 95,100.

[16] Christensen, A.L., and Dorigo, M. 2006. Incremental Evolution of Robot Controllers for a Highly Integrated Task. *Proceedings of The Ninth International Conference on the Simulation of Adaptive Behavior SAB'06,* 473-484. Rome, Italy.

[17] Gomez, F., and Miikkulainen, R. 1997. Incremental evolution of complex general behavior. *Adapt. Behavior. 5, 3-4 (Jan. 1997),* 317-342.

[18] Lanchester, F., 1916. *Aircraft in Warfare: the Dawn of the Fourth Arm, Constable and Co.* Ltd, London.

[19] Pettit, L.I.; Wiper, M.P.; Young, K.D.S. 2003. Bayesian inference for some Lanchester combat laws. *European Journal of Operational Research, Volume 148, Number 1,* 152-165(14). 1 July 2003.

[20] C. Green. *SharpNEAT homepage*. http://sharpneat.sourceforge.net/, 2003–2009.

[21] D'Ambrosio, D.B.; Stanley, K.O. A Novel Generative Encoding for Exploiting Neural Network Sensor and Output Geometry. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007).* New York, NY: ACM, 2007. 8 pages.

[22] Lockett, A.J.; Chen, C.L.; Miikkulainen, R. Evolving explicit opponent models in game playing, *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, July 07-11, 2007, London, England

[23] Robin Baumgarten. DEFCON API http://www.doc.ic.ac.uk/~rb1006/projects:api, 2009.

[i] Available at http://www.nerogame.org/

[ii] http://www.everybody-dies.com/

[iii] http:/www.introversion.co.uk/

Automated Event Recognition for Football Commentary Generation

Maliang Zheng[1], **Daniel Kudenko**[1]

Abstract. The enjoyment of many games can be enhanced by in-game commentaries. In this paper, we are focusing on the automatic generation of commentaries for football games, using Championship Manager as a case study. The basis of our approach is a real-time mapping of game states to commentary concepts, such as "dangerous situation for team A". While in some cases it is feasible to provide such a mapping by hand-coding it, in some cases it is not straight-forward because the meaning of the concepts can't be easily formalized. In these cases, we propose to use inductive learning techniques that learn such a mapping from annotated game traces.

1 INTRODUCTION

Watching game action on the screen can be made more exciting by providing additional commentary with the pictures. This is the case for many game genres. For example, in-game battle reports that comment on the current situation in real-time strategy games and highlight especially heroic actions by individual troops clearly can enrich the game experience and make it more immersive.

Even though passionate commentary is a desiring merit for the games, generating the automated commentary has to confront enormous technical challenge. In football domain, ROCCO[1], from DFKI, and MIKE[2], from ETL, are the two of such attempts in generating the live commentary from simulator data. However, ROCCO focuses on incrementally recognizing pitch events with combining the elementary predicates, while MIKE emphasizes that cooperating with the six analysis modules to generate a range of remarks.

In this paper, we present our work on providing in-game commentary in real-time for simulated football games within Championship Manager, a highly popular title developed by Beautiful Games Studios (Eidos). In this game, the player steps into the role of a football team manager, purchasing and selling players, as well as overseeing their training progress. The football matches themselves are simulated based on player statistics. The player only watches the simulation and can't interfere.

To generate the commentary, we first collected commentary concepts, such as "dangerous attack" from football news reports and other sources. We then attempted to manually implement direct mappings from game state to the set of commentary concepts (i.e., an *event recognition* mechanism). While for some concepts we were able to do so, for other concepts it turned out to be infeasible. Instead, we used an inductive learner to map hand-annotated game states selected from trace data, to generate the mapping function.

In this paper, we discuss our approach and present the empirical results on a few case studies, which show the successful application of machine learning to commentary concept generation. While our research has been focused on football and Championship Manager, the methods can be easily transferred to other games, as long as game trace data is readily available.

The paper is structured as follows. We first present the Championship Manager game, and the structure of the game traces. We then discuss our general approach, followed by examples of hand-coded commentary mappings and inductively learned mappings. We finish the paper with a summary and an outlook to future work.

2 CHAMPIONSHIP MANAGER

This section briefly summarizes our experimental domain, the Championship Manager game (CM[2]) 2008, presenting both the trace data specification and the game log simulation aspects. The data extracted from each simulated football match are packed by a file (Figure1) that specifies the respective *spatial information* and *action description* of moving objects (i.e. *players, officials,* and *ball).* The pitch coordinate has its origin at centre mark, the x axis runs across the pitch from top to bottom, y axis points up to the sky, and z axis runs from left goal to right goal. Therefore, for every *sample period* (stated in the file's header part), the moving object's *position* is indicated in three-dimensional space[3] as a decimal fraction with sign. In addition, the player's *facing angle* is recorded and measured in degrees, which starts from the z axis and clockwise increases in the xz plane.

```
---------General Information
Version:1
Sample period:0.1■■■

---------Position & orientation samples for referee (Index:0)
(Index, Position, Facing angle)
0, (50, 0, 0), 270
1, (50, 0, 0), 270
2, (50, 0, 0), 270
3, (50, 0, 0), 270
   .
   .
   .
69232, (45, 0, 0), 110
69233, (45, 0, 0), 110
69234, (45, 0, 0), 110■■■

---------Action type for player:(Squad type:Home, Index:0)
(Time, ActionType)
(0.001, eActionTypeWalking1)
(1.067073, eActionTypeWalking1)
(2.202635, eActionTypeWalking1)
(3.257802, eActionTypeWalking1)
   .
   .
```

Figure 1. Sample Data File Segment

[1] Department of Computer Science, University of York, York YO105DD, UK, *kudenko@cs.york.ac.uk*

[2] CM2008 is a football-management simulation belongs to Edios Interactive Limited. http://www.championshipmanager.co.uk/

[3] CM2008 only provides a two-dimensional simulation, so, the value for the y-axis always equals 0.

The samples capturing the above spatial information are recorded sequentially and at regular time points from the start of the match until its end. On the other hand, action descriptions are sampled only at the occurrence of a defined event. Each record thus logs the event *time* and related *action type*. The CM offers more than 200 action types for describing the pitch events. Actions with prefix 'eActionType' are dedicated to identify the agents' (i.e. officials and players) elementary events, such as, eActionTypeLinesmanSignalCorner, eActionTypeShotStraight-JoggingRightFoot, and so on. Other actions that are prefixed with 'ePersonBallAction' characterize the ball's actions, which include kick, drop, dribble kick, head, and pick up.

Similar to a real football match, there are four officials operating on the field and two teams competing to get the ball into the opposing goal. Each team consists of a maximum of eleven players, but without limiting the number of substitutions during the play. A number of common occurrences (e.g. corner ball) are also implemented. The existing commentary system of CM appears to be sensitive and precise in describing the play-by-play events, which almost make up the bulk of commentary. The evaluation of play, like, 'good save', fails to be mentioned, however, there seem to be a few of generic statements about the match, such as, 'Man Utd's passing was superb' as long as Man Utd keep the ball for more than 4 passes without the opponent getting a touch.

3 GENERAL APPROACH

The main process of event recognition comprises the collection of commentary concepts and the mapping between trace data and target concepts. In order to enrich the communicable information and minimize the potential ambiguity, the concepts are rigorously selected from multiple sources, such as, the live text commentary offered by Sky Sports, the BBC live in EURO 2008, the Laws of the Game published by FIFA, etc..

Since the football events vary from each other in terms of recognition difficulty, we divided them into two groups. One group collects the events/concepts that are recognizable solely by hand-coded rules, and the other group requires more sophisticated approaches (in our case inductive learning). To make a proper partition, every candidate event must be firstly redefined as a set of attribute-value pairs, in which the attributes are expected to expressively summarize the major characteristics of the events and vary between commentary concepts (details are presented in section 5). Consequently, for those events that are constantly dominated by certain attribute(s), it is unnecessary to seek advanced methods, but simply evaluate the related value(s) of attribute(s) (see the following section for the example), while the rest are passed to machine learning process. Therefore, the attribute definition and its value calculation are two decisive factors in event recognition, especially, for the mappings which are hand-coded. The game state attributes to be used by machine learning to generate the mapping (classifier) are difficult to be decided optimally at a glance, because the attribute selection procedure must take its consideration all applicable attributes with various combinations, and the performance variance in applying different machine learning techniques on the same attribute set may even increase the complexity.

Our approach copes with this uncertainty in attribute selection. Relying on a good domain understanding, all event-related attributes are declared at first. Meanwhile, the classified instances are randomly sampled to constitute a dataset in reasonable size, and whose values for predefined attributes are thereby calculated. All selected attributes are combined in various forms, but excluding those that are redundant in attribute semantics. This results in a batch of training files that are different in the properties of instances. The Weka[4] workbench sequentially operates on these files according to three representative machine learning techniques [3, 4, 5]: C4.5, Naive Bayes, and K-Nearest Neighbor.

C4.5 is a typical Decision Tree method, which forms a general approximation of the target function by a decision tree when training examples are given, and all instances are classified by sorting them down from the root to some leaf node. Thus, the final model is expressed in understandable rules with low computation consumption for classification process. However, the training process could be computationally expensive as the amount of training examples must proportion to the complexity of problem. Otherwise, the trees are prone to errors.

Rather than explicitly search through the space of possible hypotheses, Naive Bayes, manipulates probabilities for hypotheses by estimating the frequency of various data combinations within the training examples. Although its effectiveness has been approved by many practical trials, there are two potential factors may affect its performance: 1) the assumption, the values of the attributes are conditionally independent given the target value, may be overly restrictive for some cases. 2) The computational cost on finding the optimal hypothesis could be raised when candidate domain is large.

In distance weighted K-Nearest Neighbor method, the global approximation never being performed until a new instance must be classified. Since the target function is estimated locally and differently for each new query instance, it is robust to noisy training data and effective when plenty training data are available. Nevertheless, the cost of classifying new instances can be high due to local approximation is taken place at classification time. Even worse, this mechanism might take some irrelevant attributes into account when comparing the instances and misclassify the similarity.

After obtaining the classifier model of each learning technique, cross-validation [3] and the standard error computation [6] are applied to estimate the accuracy of the learned hypotheses over unseen instances. Unless the classifier achieves an acceptable error rate, the attribute set is refined. This improvement strategy focuses on removing irrelevant attributes, or alternatively adjusts the training dataset (e.g. collect more training instances). Finally, the model with the highest statistical accuracy is accepted, and the process proceeds to *practical refinement*. All instances that fail to be recognized are classified and added to the training dataset, as the supplements, for subsequently rebuilding the classifier.

Figure 2 shows a screenshot of our commentary system. Once a game log file has been loaded, the simulation components in top right of screen are accordingly initialized: two groups of points on the pitch stand for the competing teams in blue and red colours, four of yellow points denotes the officials, and another point symbolizes the football. As long as the match proceeds, the recognized events (i.e. the commentary) will instantly pop up at

[4] Weka is a data mining software contains tools for the classification task, available at http://www.cs.waikato.ac.nz/ml/weka/

left part of the screen. The match status data (e.g. substitutions) and system control options are also shown on the screen.

Figure 2. Screen Shot of the Football Commentary System

4 HAND-CODED COMMENTARY CONCEPTS

This section uses two examples to illustrate how the mapping between trace data and target concepts is implemented by hand-coding. One of the elementary concepts in football commentary is the *ball controller*. The relevant attributes to decide whether a player A controls the ball are:

- A is acting on the ball
- The last ball action was issued by A & A is still in ball's zone (circle region centred at ball's location with radius at 1.8 meter)

Because the decision only depends on either of these two attributes to be true, the state-to-concept mapping can be generated by hand-coding. The first attribute is determined by the ball actor at a specific moment, and the information can be retrieved from the ball action section of source file. However, the second attribute's value requires further data processing: a queue is used to store the playing history of the ball actor, and it is checked whether the distance between player A and the ball is not greater than the given radius threshold.

Another commentary concept, "player A is the *offside offender*", can be defined as:

- Line referee has raised the flag for an offside offence
- Player A's teammate has kicked the ball
- When the ball was kicked, player A is standing at offside position

Similar to the former case, this concept can be announced when all three attributes are evaluated as true. The value for the first attribute can directly refer to the referee action section of the source file. The second attribute is decided by the squad type of the most recent ball actor, which is stored in the queue. For the third attribute, the process firstly sorts the players' record by their coordinate value in z-axis team by team. Then, it only has to compare the largest (smallest) value of the attacking team with the first two largest (smallest) values of the defending team at the kicking time in order to detect the offside.

5 CASE STUDY FOR CONCEPT LEARNING

In this section we present a case study[5] of constructing the state-to-commentary mapping with machine learning. The first case concerns learning role of the player, e.g. left back. Because the locations are different even if the players are taking the same role, it is difficult to tackle it by hand-coding a mapping from attributes, such as player's location. When the process begins, possible relevant attributes are chosen, as shown in Table 1.

Index	Attributes	Rationale
1	The distances between player's position and the mean line of the squad (z-axis)	The recognition of player's role usually has to refer to his relative position in the team. However, because the number of team players is variable in a match, it is not adaptable to describe their mutual relationship with a fixed number of attributes. These attributes are therefore used as an alternative measures to summarize the relative spatial relationship between player and the whole squad.
2	The distances between player's position and the mean line of the squad (x-axis)	
3	The variance of the player's positions in z-axis	Measures may be useful to describe the width of the team's standing in both horizontal and vertical directions. When players are standing wide, the distance between player's position and the mean line may be longer.
4	The variance of the player's positions in x-axis	
5	Player's coordinate X	The locations capture individual player's absolute spatial information, which may be considered together with the relative ones.
6	Player's coordinate Z	
7	Squad Side	To identify the sign difference of the position value in terms of opposite squad side

Table 1. Attribute Set for Player Role

The training instances are manually picked out while watching the visual simulation. Meanwhile, the corresponding attribute values are also computed. As we can see from Table 2, the distribution of classes of training instances reflects the fact that the 4-4-2 formation is popular among all the teams in this game.

Role / Item	GoalKeeper	RightBack	LeftBack	CentreBack	CentreMidfield	LeftMidefield	RightMidfield	CentreForward	Sweeper
Proportion in Sample	9%	9%	9%	18%	18.18%	9%	9%	12.12%	6%

Table 2. Classes Distribution in Sample Dataset (size:60)

We applied the C4.5 decision tree learning algorithm to the datasets which are summarized by different attribute combinations, the results are illustrated in Table 3. The first row,

[5] The case study only uses CM2008's output data for sample generation, and the events to be recognized are not part of the simulator

the Average Error Rate (AER), is obtained by performing the 10-fold cross validation. After taking the Confidence Interval into account, the Estimated Error (EE) reveals that the first model built from attribute combination (1,2,3,4)[6] outperforms others.

Combinations / Statistics	1,2,3,4	1,2,5,6	5,6,7	1,2,3,4,5,6	1,2,3,4,5,6,7
Average Error Rate	24.24	27.27	25.75	30.30	31.81
Standard Error	[-0.45,0.45]	[-0.51,0.51]	[-0.93,0.93]	[-0.75,0.75]	[-1.02,1.02]
Confidence Interval	[23.78,24.45]	[26.76,27.78]	[24.82,26.68]	[29.55,31.05]	[30.79,32.83]

Table 3. Statistics Data for C4.5 (size:60)

One characteristic shared by the first and the third combinations is that they are free of the ambiguity in the value of coordinate location (i.e. the attributes in first one only represent the relative distance, and the third combination uses attribute 7 to identify the squad side). Within the coordinate system, the positions of teams in both sides are symmetric according to the origin (centre mark). As a consequence, all calculations must anyhow guarantee the consistency of the sign of values. However, after observing the decision tree, we noticed that attribute 7 is redundant with relation to attribute 6 (i.e. z>0 is coherent to side=right and vice versa). Instead of indicating this side difference by a variable, the succeeding experiments attempt to automatically map all the positions of right-side-squad players into its opposing side prior to building the model.

Table 4 shows the results after redoing the experiment. The second model tends to be much more accurate in prediction, and the worst error rate is significantly reduced to 10.88%. This experiment proves the importance in terms of coordinate consistency.

Combination / Statistics	1,2,3,4	1,2,5,6	1,2,3,4,5,6
Average Error Rate	25	10.17	11.36
Standard Error	[-1.15,1.15]	[-0.71,0.71]	[-0.34,0.34]
Confidence Interval	[23.85,26.15]	[9.46,10.88]	[11.02,11.7]

Table 4. Statistics Data for C4.5 (size:60)

Although training on the same datasets, the performance for the learning models could differ significantly as the algorithm is changed from C4.5 to K-Nearest Neighbor (KNN). Table 5 presents the results for the models (the best classifier of each attribute combination obtained from intensive trails) that follow the latter algorithm, as well as their settings for the K. The results show that the second model is far more accurate than the others. Comparing the corresponding attribute sets, the attribute 3 and attribute 4 could introduce a large bias that leads to the classification fault, since the "closet" is defined by the difference of corresponding attribute values of two instances, however, the values of attribute 3 and attribute 4 are highly duplicated.

Combinations / Statistics	1,2,3,4	1,2,5,6	1,2,3,4,5,6
K	14	1	5
Average Error Rate	70.45	15.25	59.09
Standard Error	[-3.37,3.37]	[-1.57,1.57]	[-3.71,3.71]
Confidence Interval	[67.08,73.82]	[13.68,16.82]	[55.38,62.8]

Table 5. Statistics Data for KNN (size:60)

Figure 3 shows the screenshot of classifier errors generated by Weka. The horizontal and vertical axes represent the labelled class and variance in x-axis (attribute 4) respectively. The plots are grouped into three groups, and the "X" denotes correct prediction and the "口" specifies the false classification. When the vote starts, the closet neighbors are always found in the identical group and dominate the classification if this class has comparatively larger amount. In this example, at least one more instance is marked by either turquoise "X" or green "口", especially for the group in the middle. This output is also consistent with the one we have got by using C4.5 (See Table 4).

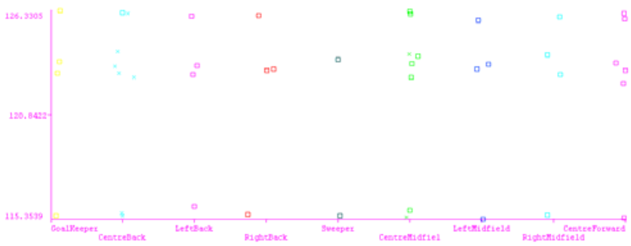


Figure 3. Plot visualization – KNN (attribute set: 1,2,3,4)

The models approximated by Naive Bayes (NB) are relatively weaker on instance classification, when comparing their statistical data (in Table 6) with the former ones.

Combinations / Statistics	1,2,3,4	1,2,5,6	1,2,3,4,5,6
Average Error Rate	29.54	20.33	20.45
Standard Error	[-1.33,1.33]	[-1.29,1.29]	[-0.9, 0.9]
Confidence Interval	[28.21,30.87]	[19.04,21.62]	[19.55,21.35]

Table 6. Statistics Data for NB (size:60)

[6] The attribute index details please refer to Table 1

Because the estimated error rate satisfies the application requirement, the second model of Table 4 is eventually adopted into this commentary system. Figure 4 reveals that the player roles in this match are all classified correctly. However, the model may misclassify particular roles especially at some occasions. The proposed solution for lowering such occurrence rate is continuously extending and refining the training dataset according to these misclassifications.

Figure 4. Player Role Classification

The ball is frequently passed in each football match. Human observers are able to identify the possible path of the pass and recognize its candidate receivers as soon as the ball is kicked. Sometimes, this assumption directly derives the commentary. Such as, "player X runs to the ball". However, this is still a big challenge for the computers. Since such *ball transfer path* recognition more or less relies on certain background knowledge and a global view of the match, we tackled this problem with a machine learning approach. In addition, the understanding on the ball transfer path can be used to address another difficulty in identifying the player's kicking intention for either passing the ball or a direct shot at the goal.

Because a football match is a highly flexible domain, the number of players may vary unpredictably, and makes it difficult fixing this flexible event into the machine learning process. Maintaining a decision structure that includes the mutual relationships between ball holder and all other players cannot be allowed, because the classifier features must be fixed from the beginning. Instead of deciding which teammate this ball will be passed to, every teammate must predict whether the ball is going to be passed to him. Consequently, each player should estimate the potential ball passing from the fixed set of attributes (Table 7). One of the drawbacks of this design is the lack of global information, which may lead to two or more players being recognized as the ball receiver in some specific cases.

Index	Attributes	Rationale
1	Sender's generic pitch position	This attribute aims to manually transform the numeric attribute values into the nominal items by dividing the whole pitch into several partitions. Because the ball kick action may happen anywhere in the pitch, which might results the attribute values widely different from instance to instance, we do not use the player's absolute coordinate location to identify the spatial information
2	Receiver's generic pitch position	
3	Sender's face angle	The player's face angle may be considered together with his kicking skill for deciding the possible ball transfer path
4	Sender's action	This attribute value is given in the data file to notice the corresponding player actions. Different kicking events may represent different power and direct the ball's flying angle
5	Receiver's action	
6	The angle between the player and the ball holder	This is another spatial measures for recording the relative positional relationship between player and the ball holder
7	The risk of interception	This is a factor for ball passer usually has to evaluate
8	The shortest distance between receiver and defender	This is another factor concerning how easy the receiver can touch the ball from the ball holder's perception

Table 7. Attribute Set for Ball Transfer Path

The decision on attribute combinations is made according to their semantic coherence. For example, the combination (1,2,3,6)[7] tries to capture the absolute spatial relations between ball sender and the ball receiver through attribute 1 and attribute 2, while the sender's face angel may be combined with the relative angle between the player and the ball holder to show their relative positional relationship. As long as the training instances and attribute combinations are determined, the first group of models adhere to the C4.5 algorithm can be achieved. Due to the poor ranking in performance (in Table 8), the initially defined attribute combination set can simply remove the third (1,2,3,4) and the forth (1,2,3,6) options from the first iteration.

Statistics / Combination	Average Error Rate	Standard Error	Confidence Interval
1,2,3,4,5,6,7,8	21.88	[-1.18,1.18]	[20.7,23.06]
1,2,3,6,7,8	21.88	[-1.18,1.18]	[20.7,23.06]
1,2,3,4	31.25	[-0.35,0.35]	[30.9,31.6]
1,2,3,6	31.25	[-0.35,0.35]	[30.9,31.6]
3,4,5,6,7,8	20.31	[-2.2,2.2]	[18.11,22.51]
5,6,7,8	21.88	[-1.57,1.57]	[20.31,23.45]

Table 8. Statistics Data for C4.5 (size:60)

[7] The attribute index details please refer to Table 7

Because the confidence intervals for the models in Table 8 overlap with each other, advance refinement would be better to attempt to filter both attribute and training instances. However, only the last model's performance is improved among the results shown in Table 9

Combination \ Statistics	Average Error Rate	Standard Error	Confidence Interval
1,2,3,4,5,6,7,8 (ClassOrder)	21.87	[-1.18,1.18]	[20.6,23.05]
1,2,3,6,7,8	20.31	[-1.18,1.18]	[19.13,21.49]
3,4,5,6,7,8	25	[-2.2,2.2]	[22.8,27.2]
5,6,7,8 (Resample)	18.75	[-0.71,0.71]	[18.04,19.41]

Table 9. Statistics Data for C4.5 with Filter(size:60)

Similar statistical analysis is conducted on the models based on KNN and NB as well. Nevertheless, neither of the solutions could significantly boost the prediction accuracy. The probable reason is the size of dataset is extremely small compared to the large variety in attribute values. Therefore, the forth model of Table 9 could be temporarily added into the system as a compromise of lacking a sufficiently large training dataset.

In most occasions, the choice of teammate for ball passing can be very flexible, even a human observer may fail to make a precise prediction. However, for some specific situations, like there is only a player standing in open space, the recognizer nevertheless can make the effective estimation.

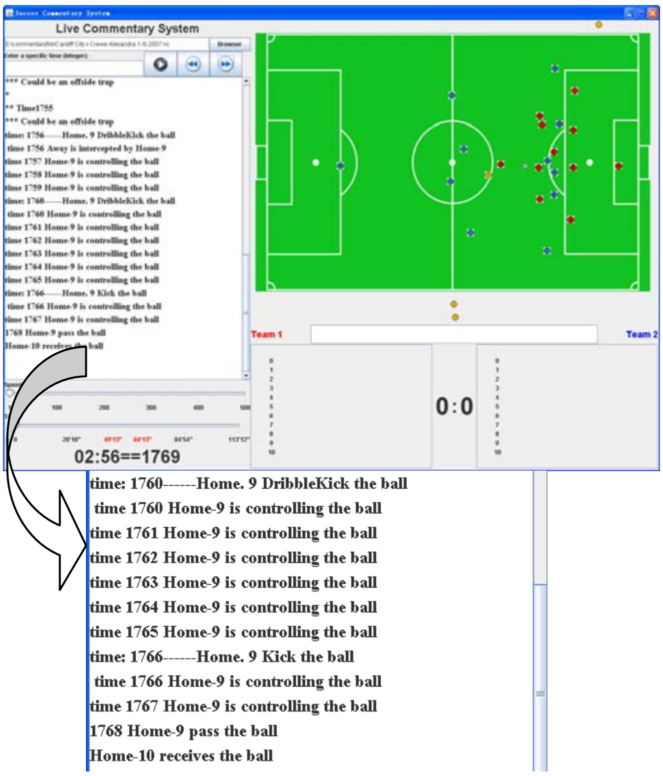


Figure 5. Ball Transfer Path Classification

The Figure 5 shows an example of correct prediction: the ball is predicted as to be passed to the centre forward when playing the counter attack.

Offside trap is very often mentioned in the real commentary, and it is also the precondition for deciding if the attacker successfully beats the trap. The reasonable solution for the computer to understand this event is that the system must have some sorts of background knowledge on its occurrence occasion, and have ability to decide if the defending team probably has the motives for making the trap. It is a challenge work for traditional approach to take all the possible factors into account, nonetheless, may be solvable by machine learning. All attributes are thought to be relevant are listed in Table 12.

Index	Attributes	Rationale
1	The mean value of defending line (in z-axis)	This attribute is used for occasion description. We believe there may be some implicit relationships between the relative distance from d-line to their goal and trapping the attacker into offside position. For example, when the team's d-line is high, the defenders could play the trap to against the counter attacks and generally cutting down on conceding goals.
2	The distance between the ball location to the mean of defending line (in z-axis)	This figure also works for providing the classifier more information about the match situation. For example, if the ball is already in the defending third, the back's forward motion may be less probably for playing the trap but blocking the attack
3	The distance between forward, and the back who is standing closest to the goal	This is a critical attribute for the final classification. If the distance is too far or forward has already stood at the offside position, it does not make sense to for backs to launch the trap.
4	The face angle deviation of players who is standing on the defending line	This attribute is used for human intention prediction. As playing this trap requires all backs move forward together, they must communicate with each somehow for cooperation.

Table 12. Attribute Set for Offside Trap

Similar to the former cases, we pick out all sensible combinations of attributes. The Table13 illustrates the future error rate prediction for each of the models derived from the combinations. Most of model's average error rates are around 20%, but the fifth one's (combination 1,2[8])reaches at 45.61%, which is definitely less accurate than others. Due to the overlapping in confidence intervals, it is not safe to assert whether the first, the third, or the seventh is the most accurate model.

Combination \ Statistics	Average Error Rate	Standard Error	Confidence Interval
1,2,3,4	19.30	[-1.69,1.69]	[17.61,20.99]
2,3,4	22.81	[-1.48,1.48]	[21.33,24.29]

[8] The attribute index details please refer to Table 12

1,3,4	19.30	[-1.69,1.69]	[17.61,20.99]
3,4	24.56	[-1.31,1.31]	[23.25,25.87]
1,2	45.61	[-1.35,1.35]	[44.26,46.96]
1,2,4	21.05	[-1.93,1.93]	[19.12,22.98]
1,2,3	19.29	[-0.62,0.62]	[18.67,19.91]
2,3	36.84	[-2.02,2.02]	[34.82,38.86]

Table 13. Statistics Data for C4.5 (size:60)

Since the attribute space for this event is small, and the error rates are at a similar level, the refinement process should focus on the dataset. From the last table's result, the further experiments are restricted only on those combinations that probably can derive the best performance, therefore, the second (2,3,4), forth (3,4), fifth (1,2), sixth (1,2,4), and eighth (2,3) combinations must be firstly removed. Consequently, by generating a subsample with random replacement, the results presented in Table 14 reveal that the models of the first and the second combinations perform equally best, the predicted error rates are between 10.95% and 13.97%.

Statisitcs / Combination	Average Error Rate	Standard Error	Confidence Interval
1,2,3,4	12.28	[-1.69,1.69]	[10.59,13.97]
1,3,4	12.28	[-1.69,1.69]	[10.59,13.97]
1,2,4	29.82	[-1.93,1.93]	[27.89,31.75]
1,2,3	17.54	[-0.62,0.62]	[16.92,18.16]

Table 14. Statistics Data for C4.5 (size:60)

From Table 15, a more accurate model is obtained from using the second combination (2,3,4) compared with the best result we have got from Table 14. The possible reason is the combination of attribute2, attribute3, and attribute4 is deterministic for the final classification decision, thus, taking them as the calculation factors for defining the neighbour can be comparatively effective.

Statistics / Combination	Average Error Rate	Standard Error	Confidence Interval
1,2,3,4	8.77	[-1.32,1.32]	[7.45,10.09]
2,3,4	7.01	[-1.09,1.09]	[5.92,8.1]
1,3,4	10.52	[-1.58,1.58]	[8.94,12.1]
3,4	10.52	[-2,2]	[8.52,12.52]
1,2	21.05	[-2.16,2.16]	[18.89,23.21]
1,2,4	35.08	[-1.12,1.12]	[33.96,36.2]
1,2,3	40.35	[-0.93,0.93]	[39.42,41.28]
2,3	42.1	[-2.63,2.63]	[39.47,44.73]

Table 15. Statistics Data for KNN (K=1, size:60)

Table 16 summarizes the best results of NB. The best accuracy is achieved with the second attribute combination (2,3,4), which is also the best set for KNN. In contrast to the poor performance in C4.5, the forth combination (3,4) is much more improved. Consequently, we move on to the performance evaluation.

Statistics / Combination	Average Error Rate	Standard Erro	Confidence Interval
1,2,3,4	10.52	[-0.96,0.96]	[9.56,11.48]
2,3,4	8.77	[-1.16,1.16]	[7.61,9.93]
1,3,4	10.52	[-0.96,0.96]	[9.56,11.48]
3,4	8.77	[-1.16,1.16]	[7.61,9.93]
1,2 (after filter)	36.84	[-1.1,1.1]	[35.74,37.94]
1,2,4	17.54	[-1.32,1.32]	[16.22,18.86]
1,2,3	17.54	[-0.38,0.38]	[17.16,17.92]
2,3 (after filter)	19.29	[-0.7,0.7]	[18.59,19.99]

Table 16. Statistics Data for NB (size:60)

In all cases, except of some faults, the model generated by the NB is found out to be more accurate than the KNN, which proves its effectiveness in classification in this domain. In Figure 6, the text box shows that at 6004 (time index), the system recognizes the backs (in blue) are running forward maybe for playing the offside trap. And the visualization reveals this prediction at 6008 (i.e. the backs keep moving up from 6004 to 6008).

Figure 6. Offside Trap Classification

6 RELATED WORK

To date, we are aware of two other similar attempts at producing the automated football commentary systems in the domain of RoboCup[9]: ROCCO and MIKE. One feature of ROCCO is that it organizes the concepts in hierarchical way, which supports incremental event reorganization. Thus, elementary event and state predicates, which only happen at a single timepoint, are utilized to define those high-level event concepts. Much different from ROCCO, MIKE solves the difficulty by adopting three dedicated agents. One is for simple events, one is for pass work, and another is for shots and goals. Even though the events in both methods are identified by matching predefined propositional models (pattern) [7], the analyzer in ROCCO outperforms the other by the variety of its recognizable events. Intentional concepts, such as, offside trap, is inferred in ROCCO's commentary, while MIKE merely commentates on the ones that have taken place. However, besides the endeavour in event-based analysis, three other agents of MIKE enable it to perform additional state-based analysis. The system is able to automatically comment on the player's action, say, good passes, which benefited from interpreting the knowledge in a mathematical model. For the part of discourse planning and natural language generation, MIKE not only considers the importance of candidate events, but also tends to simulate interruption and abbreviation when generating the template based commentary.

According to the former survey, many strategies have been involved to create the pitch event recognizer but none of them can thoroughly solve the problem. One of the main difficulties is guaranteeing the output accuracy. In both systems, the success of event recognition definitely depends on the correctness and completeness of predefined conditions that are crafted manually. Therefore, their performance may be degraded dramatically when handling tricky occurrences by those static rules [6]. For instance, traditional solutions are occasionally misclassifying a player's kick action as a shot when he merely intends to pass the ball to the teammate who is nearer to the goal. Another weakness concerns the efficiency when generating the model for a large population of recognizable events. Both problems are alleviated by our proposed approach.

7 CONCLUSIONS & OUTLOOK

In this paper, we have described an approach to in-game commentary generation, which is based on the mapping of states to commentary concepts. We showed that while some concepts can be produced by hand-coded mappings, other concepts require a more sophisticated approach. Specifically, we propose the application of inductive learning, and the results of our case studies show the feasibility of this approach.

In order to deploy our approach to a real game, more concept categories need to be tackled, and integrated into the commentary generation system. Also, the sophistication of the text generation can be improved (so far commentary text is based on a small set of simple templates).

In future work, we also plan to extend the approach to other game genres, such as real-time strategy.

ACKNOWLEDGEMENTS

We thank Beautiful Games Studios for providing the data sets used in our study, and specifically Alex Whittaker for his assistance.

REFERENCES

[1] D. Voelz, E. Andrè, G. Herzog, and T. Rist. Rocco: A RoboCup Soccer Commentator System. In M. Asada and H. Kitano, editor, RoboCup-98: Robot Soccer World Cup II, Springer. ISBN 3-540-66320-7 (1998)

[2] K. Tanaka-Ishii, I. Noda, I. Frank, H. Nakashima, K. Hasida, and H. Matsubara. Mike: An automatic commentary system for soccer. In Proceedings of ICMAS-98.

[3] T. Mitchell. Machine Learning. McGraw-Hill Series in Computer Science. ISBN 0-07-042807-7 (1997).

[4] J. Quanlan. C4.5: programs for machine learning. Morgan Kaufmann series in machine learning. ISBN 1-55860-238-0 (1993).

[5] I.Witten and E. Frank. Data Mining: practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann. ISBN 0120884070 (2005).

[6] J. Gosling. Introductory Statistics. Pascal Press. ISBN 1864410159 (1995).

[7] E. Andrè, G. Herzog, and T. Rist. Multimedia Presentation of Interpreted Visual Data. In P. Mc Kevitt, editor, Proc. Of AAAI-94 Workshop on "Integration of Natural Language and Vision Processing", pages 74-82, Seattle, WA, (1994). Also available as Report no. 103, SFB 314-Project VITRA, Universität des Saarlandes, Saarbrücken, Germany.

[9] RoboCup™ is an international joint project uses soccer game as a central topic of research. More information will be available at: http://www.robocup.org/

Serious Games to Teach Ethics

Rania Hodhod[1] and **Daniel Kudenko** and **Paul Cairns**

Abstract. In this paper, we are focusing on digital serious games (edugames) and how they can be utilized in teaching in the ethics and citizenship domain. Our aim is to combine narrative techniques with intelligent tutoring techniques in a single model that adopts and based on educational theories and classroom educational strategies. The model has been used to implement an adaptive educational interactive narrative system (AEINS). AEINS is an inquiry based edugame to support teaching ethics. The AEINS version presented in this paper targets students between the age of 8 and 11. The idea is centered around presenting and involving students in different moral dilemmas (called teaching moments) within which the Socratic Method is the used pedagogy in the teaching process. AEINS monitors and analyzes the students actions in order to provide an individualized story-path and an individualized learning process. The student is an active participant in the educational process and is able to interact with the edugame as a first person player. We claim that such interaction can help in developing new or deeper thoughts about different moral situations. Our aim is to contribute to the design of serious games and help raise awareness of ethics and citizenship in children.

1 Introduction

Computer game worlds have become more complex over the years as computer technology has evolved. Games are a very dynamic field, and they have moved on significantly since the simplicity of Pong with many improvements and expansions. Since the 1950s, computer and cognitive scientists have developed the idea that the computer can be used by a student to learn independently and that computer programs can teach a student. For example, McGrenere started investigating whether games could be utilized to assist learning [25] and others explored the appropriate game types and game elements to be used as educational tools [1]. Some researchers such as Klawe [34] consider these games only effective if the interaction is monitored and directed by teachers, or if the games are integrated with other more traditional activities such as pencil-and-paper exercises. Other researchers believe that effectiveness is related to the features, preferences and behavior of a particular user [25].

In the last few decades, Serious games (edugames) started to emerge. Studies on the use of games in education [1, 14, 15, 28, 29, 30, 32, 33, 41] have proven that games constitute a medium that motivates the learners to try to develop their knowledge while they put it into practice. Games became a strong supplement to teaching by virtue of their concrete experiences leading to learning. Instead of being taught about topics, students are engaged with these topics and play them out. Within such environments players can learn while being engaged in an entertainment activity [31] and thereby students create their own experiences and get feedback on their specific actions in a safe environment [41].

As obvious, not all users share the same preferences or styles when interacting with a game and when solving game-problems. This leads to the importance of adaptation in the sense that behavior of each play-instance of a game depends on the actions of an individual player. The major aim for an adaptive game-based learning system is to support and encourage the learner by considering his needs, strengths and weaknesses. The telling of stories within these environments takes the role of engaging the player and/or transfering tacit knowledge to the learner. Simpson in her article emphasizes the importance of stories in our lives and their role in tightening human relationships [9]

> ”Stories are connections to the past and yet carry us into the future; they speak of relationships, of human connections, and to what gives a quality to our lives.”

Stories and interactive narrative (IN) have been used for long time now to entertain children and to teach them, for example in classrooms for primary and secondary school curricula, both on their own and as a support for other subjects [17, 43]. IN allows teachers to introduce sensitive issues in a safe and stimulating way. Interactive narrative has proven to be successful in creating enriching experiences for its users, sparking problem-solving skills, individual and group decision-making skills, and encouraging pupils to develop strategies to deal with different issues in different disciplines. In addition it has been shown that role playing and discussions can help students to transfer their knowledge and beliefs into actions and can help them to see that their decisions affect other people and things [24].

Interactive narrative has mainly been used as a common tool to teach in ill defined domains such as design, history and ethics. The Socratic Method is the most widely used pedagogy in telling these stories. Ethics and citizenship is one of the important ill-defined domains, where the development of skills of participation and responsible action is a fundamental part of the citizenship curriculum. According to [21], ”By teaching students how to make good choices, we are in effect, educating character.” Character education involves teaching children about basic human values including honesty, kindness, respect ...etc. In addition, according to Kohlberg, if children get engaged in enough independent thinking they will eventually begin to formulate conceptions of rights, values, and principles by which they evaluate existing social arrangements [16].

In such domains, like ethics, it is a generally accepted fact that knowing is different from doing, for example, as Watson clarifies [7]: ”getting high scores in an ethical course does not guarantee at all the actual behavior of that student”. Watsons view can be summarized as follows: 1- identify what is good and what is bad behavior. 2- Instruct people as to what these are. 3- Inspire people to behave in the right ways using examples to imitate. This view presents the

[1] University of York, England, email: rania.hodhod@cs.york.ac.uk

classical view of moral education which is straight forward but unfortunately reality is much more complex. Watson also mentioned a very important point which is the desire for good:

> ”The trick lies not solely with knowing what is right and good but also in building a love for the good and the worthwhile.”

Watson points out that by giving the students the chance to see successful people do what's right and good, chances are better that learners will be biased to follow suit themselves than they might otherwise.

Considering the importance and the challenges the ethics domain provides plus the theory behind serious games and interactive narrative, a hybrid architecture that combines interactive narrative and intelligent tutoring has been designed. Based on this architecture, an edugame called AEINS has been implemented. AEINS aims to use stories and interactive narrative as a source of inspiration and direction for moral conduct. We believe that by allowing the student to be involved in moral dilemmas helps them to express his character through the problem solving, decision making, and conflict resolution present in these dilemmas. These are important parts of developing moral character. In other words, we aim to have strong learning objectives underpinned by effective story telling. Although edugames now is a growing area of research, most educational games to date have been produced without any coherent theory of learning or underlying body of research [13]. In addition, very few formal evaluations have been conducted to evaluate the actual pedagogical values of these games [27].

2 Related Work

Because narrative plays such an important role in cognition and culture, narrative-centered learning environments have been the subject of increasing attention. Learning narrative environments can be viewed as rich generated stories that transfer some educational concept(s) or skill(s) to the student. From the environment designer point of view, he is intentionally placing educational materials in these environments in order to reach certain educational outcomes (hopefully!). On the other hand, from the student point of view, a kind of unintentional learning process happened through an engaging, appealing experience. The student is seen as an active participant in the construction of his own knowledge.

Since the narrative centered learning environments tend to have educational targets so the environments must exhibit certain educational properties, and contain one or more of the following educational components, for example the existence of tutorial planning, student model, learning objectives, domain and pedagogical models. Some environments also contains feedback/hint provider(s). Different ideas have been proposed on the kind of interaction between the tutoring modules and the narrative module. One way is to link the learning objective to the game narrative content as in ELECT BILAT, a game-based system to teach cultural awareness and negotiation skills for bilateral engagements [20]. Each game action is linked as positive or negative to a learning objective. The researchers believe that within an educational context, emergent narrative is not efficient enough as the experiences the learner ends up creating may not contribute to the intended or desired learning goals. A more scripted approach has been followed to achieve greater control of learner experiences. According to Riedl and Young [35], this kind of narrative is not as adaptive as a planned narrative as it limits the learner's freedom in the environment.

Another kind of interaction appears in Mimesis [26], Thomas and Young employ adaptive interactive narrative to guide discovery learning where learning tasks are represented in the interactive narrative plans. Mimesis uses the intervene strategy; a strategy that allows the user action to fail if it would change the world in a way that conflicts with the actual story plan. Actually, we think that if this kind of intervention persists in order to preserve the story unfold, this can lead to the user boredom and loss of interest. Also, this can override the educational goals.

The BAT ILE system is a binary arithmetic tutor that uses narrative to provide context of learning binary arithmetic and logic gates [2]. The narrative is different in this system where there is no evolving story line, and the environment is inhabited by one tutor agent where the communication with the student seems to be in one direction, the tutor student direction. In other words, the effect of the student actions are not obvious on the narrative, although it affects the choice of the subsequent problems and puzzles. It has been shown that students enjoy using BAT ILE, but it was not clear that the system's design was the main reason although it can be considered an important factor.

More systems developed such as Storyteller in the literacy education domain [5] and Crystal Island in the microbiology domain [6, 42]. StoryTeller discusses the idea of helping children to customize stories, they go through iterative cycles of writing and review till their story comes to life. The narrative in Storyteller is pre-defined by the children before starting the game. Crystal Island investigates the incorporation of an automated story director and intelligent tutor into a narrative-based training simulation. The authors take the same approach as ours with respect to the way the tutoring and narrative components interact together, but Crystal Island does not contain a student model, which has proven its importance in providing adaptivity. Another story-based educational game developed for language learning is TLCTS [22], which helps people to acquire communicative skills in foreign languages and cultures. TLCTS combines game design principles and game development tools with learner modeling, pedagogical agents, and pedagogical dramas.

FearNot! is a character-driven learning environment developed for the anti-bullying domain [43]. Although the system does not include the student as a protagonist, the student is still able to affect the story through offering suggestions to the victimized virtual character. The environment emphasizes highly autonomous agents that promote empathic relationships with the student, it does not have any kind of student model and the learning environment is not able of handling more social educational subjects rather than anti-bullying one. Prada et al. (2000) allow the user to act as a protagonist in ”TEATRIX” [40], a learning environment designed to help children and their teachers in the whole process of collaborative story creation. Children start with a selection phase where they set up the scenes for the development of a play and design the characters, in addition to doing the whole performance. In TEATRIX, there is no pedagogical or student models, which helps in achieving the personalized learning. In addition, the story generation is limited to the scenes pre-selected by the children before the beginning of the play. this kind of interaction decreases the flexibility of the story generation and limits the users choices of having different unexpected events in the story which, we think, if considered can increase the immersion, dramatic and emotional effects that in turn may affect the educational outcome.

The interactive narrative learning environment, IN-TALE, developed for military training skills [36]. It integrates an automated story director and reactive autonomous agents in the context of a story-

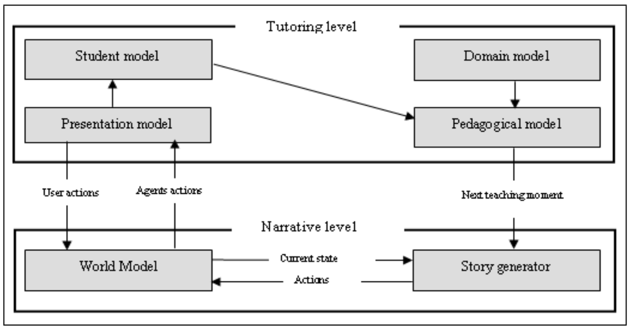


Figure 1. An architecture for a narrative-based system that includes intelligent tutoring

based military leader training scenario. In the scenario, the user plays the role of a military leader in a foreign peacekeeping mission and the virtual characters play the roles of merchants in a volatile marketplace. In-TALE aims to expose the user to dramatic and pedagogical situations, with the use of the surrounded agents believable behavior, to help him to acquire the desired skills for such missions, but also the system lacks adaptivity to the educational process. A further interactive drama prototype is TIME that has been developed in the field of medicine [44]. The system portrays a situation in the life of a virtual patient. The system contains eight decision points that rely on a specific table of probability values to determine the outcome of the situation in question. Although TIME is able to assess the student work at the end of the presented scenario, it has no student model that guides the educational process during real time interaction with the system.

To the extent of our knowledge, no narrative-based educational game has included a student model in order to teach citizenship and ethics. Our goal is to provide an interactive narrative experience that merges the benefits of interactive narrative approach with the benefits of strong adaptive tutoring approach (designated in the presence of a pedagogical and student models)in order to teach in the citizenship and ethics domain.

3 Proposed Architecture

The main idea of the proposed work is to make use of narrative technique and intelligent tutoring in a single architecture. Interactive narrative allows interaction between the student and the story teller, it contains inspiring examples after which the students can model their own behavior. Intelligent tutoring usually infer a student model from student behaviors to adapt the instruction to the students needs. Each module in the system is fully separated from the other parts; all the parts can communicate through inference rules. As we aim to make use of the presented architecture in a learning environment, we believe in the necessity of having the narrative level work in favor of the tutoring level. In principle, the dynamically generated story bits are mainly generated to serve the tutorial goals, see $Fig.1$. As seen in figure, although tutoring and narrative are in separate levels, there are back and forth communications between the various modules present in both levels.

3.1 Tutoring Level

The tutoring level consists of four components: domain model, student model, pedagogical model and presentation module. The domain model is a main component of tutoring systems, it is a dynamic model where a set of rules are implemented by which the system can reason. The student model is a crucial component that mainly aims to provide adaptivity; it involves creating an individual model for every student. The pedagogical model adapts instruction (problem selection, problem difficulty, topic area, choice of activity, choice of help type, and availability of help) following a model of human tutoring expertise that balances motivational and cognitive goals. Finally, the presentation model handles the flow of information and monitors the interactions between the user and the system and vice versa.

3.2 Narrative Level

The narrative level consists of two components: a world model and a story generator. The world model houses the objects and the AI characters'declarative models. Each AI character is implemented as a set of rules that describe the character personality and control the character behavior. All the characters share the same basic knowledge base to support interacting with the world and other characters. The story generator uses continuous planning for the dynamical generation of story beats in run-time.

4 AEINS

Based on the architecture presented, a prototype called AEINS (Adaptive Educational Interactive Narrative System) has been developed. AEINS is an inquiry-based learning environment, that helps 8-11 years old children to be engaged effectively in moral dilemmas. Our approach is to provide customized learning environment and personalized feedback to help to maintain immersion and engagement during the learning process in ethics and citizenship domain. This is achieved through presentation of new insights into combining interactivity with pedagogy for engaging students effectively in moral dilemmas. In AEINS, the student is able to act freely in the environment influencing the path of the story unfold, and at the same time be monitored and guided by the tutor.

AEINS proposes two types of agency. The first kind is complete free agency by which the student is able to influence and control the direction of the story (i.e., before reaching or after finishing a teaching moment). In this case, the story generator uses continuous planning for the dynamical generation of story beats in run-time to provide suitable actions and reactions in response to the student's actions. In other words, the narrative generated in the game through planning where for every possible way the user can violate the story plan, an alternative story is generated. The second type is restricted agency which exists in the entire interaction within a teaching moment; the teaching moments use simple branching planning approach designated by decision points where the student has to act. Restricting the agency in order to preserve the educational targets is acceptable because the teaching moments themselves are relaxed by varying the places and characters that can participate in their worlds; this part is illustrated in [39]. AEINS main aim is to allow students to move from the *making moral judgments state* to the *taking moral actions state*, from the knowing state to the doing state, which is considered a very important step in moral education. The following subsections introduce how the various architecture components are utilized in AEINS.

4.1 Domain Model

The domain model describes the various concepts (values) in the ethics and citizenship domain and their relationships. One part of the domain in model is defining the principles of character education [11] and represent their relationships and dependencies. A frame based representation has been used to demonstrate those relationships and dependencies as seen in $Fig.2$. The other part of the domain model is constructed in the form of a repertoire of moral dilemmas (teaching moments). An example of a branched graph structure dilemma is shown in $Fig.3$. Mapping the domain values to the teaching moments can be imagined as shown in $Fig.4$.

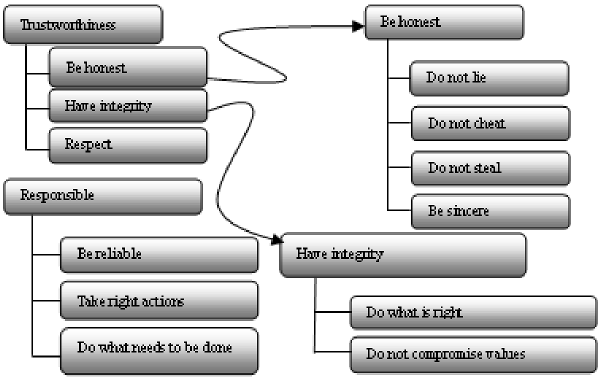


Figure 2. Frame based representation

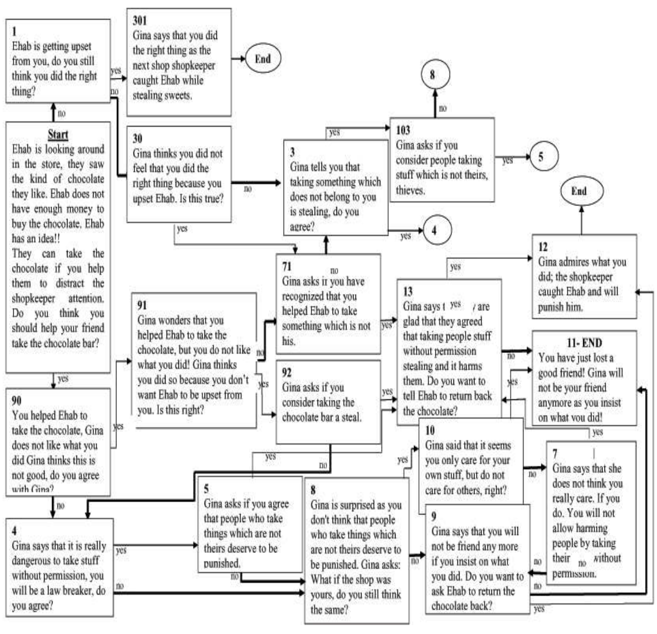


Figure 3. Graph structured dilemma

4.1.1 Moral Dilemmas and Teaching Moments

The ethical argument as a whole is ill structured and it is hard to define the set of right answers or actions. But according to Simon and his explanation of the architects design process [18], "During any

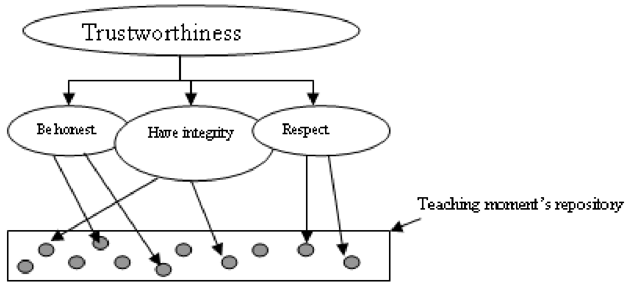


Figure 4. One to many relation between the moral values and the teaching moments

given short period of time, the architect will find himself working on a problem which, perhaps beginning in an ill structured state, soon converts itself through evocation from memory into well structured problem." Simon meant to say that a problem that is ill structured in large can be well structured in small. Upon this, we decided to make use of pre-analyzed moral dilemmas in a way that every analyzed part can act as a separate well defined problem on its own.

Moral dilemmas such as Kohlberg's moral dilemmas and other dilemmas designed specially for school children are used to construct teaching moments. Every teaching moment can be imagined as non-interactive story presentations interleaved with user decision points that allows the story to progress forward [39]. The teaching moment structure is in the form of a directed graph, where each node is a distinct situation of the world and the directed edges link decision points with each other or leads to an end [4].

This kind of representation allows the use of an intelligent tutor to follow the student's actions and assess them in the form of a step by step follow up. Ideally, each teaching moment path describes an inquiry based narrative, a story in which the protagonist is the user in the role of making moral decisions. These dilemmas allow students to pursue different procedures for solving the problem. These various procedures arise from allowing different perspectives based on students perceptions and interpretations of the nature of the problem. Through role playing and discussions, students can see that their decisions affect other people and things [24]. In addition, the students' understanding gained through this process is situated in their experience and can best be evaluated in terms relevant to their experience.

Although the different branches of every teaching moment are hand coded, each teaching moment exhibits variability through allowing different characters and places to present the teaching moment depending on the story world state. Each teaching moment represents a part of the whole story and focuses on a certain concept (value) in a way that the concept mastery is established within. Each teaching moment has certain prerequisites that must be fulfilled before the execution of the teaching moment takes place. Manipulating teaching moments priority are done through the represented rules as follows:

Trigger: teaching moment X_1 has not been presented
and teaching moment X_2 has not been presented
and value Y is not held by the user
and value X is held by the user
Action: set priority to teaching moment X_2

The capital letters in the rules represents variables and the representation denotes that if (a) a specific pattern of teaching moments has not been presented to the student yet and (b) user holds certain values and does not hold others, the action part of the rule executes (the next teaching moment priority is identified).

4.1.2 Interaction with teaching moments

Within every teaching moment, the learning tasks are tightly coupled with the narrative where specific skills (goals) have to be acquired by the student. The interaction of the student with the teaching moments follows the following structure. Such interaction is monitored and evaluated by the cognitive tutor.

- The teaching moment begins with a specific theme to act as a starting point or trigger for learning.
- Generate possible problem solutions "ethically approved."
- Assess the viability of alternative solutions by constructing arguments and articulating beliefs.
- The student is asked about his opinions through series of questions
- Evaluate the students answers and construct arguments that lead the students to examine the validity of his opinion or belief.
- If the student agrees with a desired ethical choice, the teaching moment ends
- Else if the student sticks to an unethical decision, the system raises the ante (what if style questions).
- If the student keeps sticking to the unethical choice, choose another teaching moment and repeat the above steps.

4.2 Pedagogical Model

The pedagogical model is developed in the form of production rules. These rules are used to give the system specific cognitive operations to reason about the student and the teaching process. The model specifies how a student ideally would use the system and how the system reacts to his actions. According to the student's actions, the model is able to assess the student's skills and adjusts the student model accordingly. In order to design the pedagogical model, the problems structure and what exactly needed to be modeled has to be specified. With ill defined problems, development is a change in the way a person thinks not is the case of acquiring more knowledge. So to enable the student model to work effectively, we have specified skills needed to be acquired by the student and the actions that should be taken by the student in order to reflect these skills. The idea is based on analyzing moral dilemmas and transforming them to a story graph structure. The second step is to specify the decision points that reflect the specified skills.

Research suggested that students benefit from being encouraged to consider a collection of evidence and coordinate their theoretical ideas with supporting or contradictory evidence as they engage in argumentation [3, 37]. In addition, students must have opportunities to choose among different options and to reason which criteria lead to the option chosen [12]. AEINS follows these approaches in designing the pedagogical model and uses the Socratic Dialogue as it has been shown to be a highly effective approach [10] to help children develop new ideas and gain new insights. The Socratic Dialog also helps the students to think critically, solve problems non-violently, and make choices based on what's right instead of what they can get away with. It also help learners to face evidence that what they believe to be true is, in fact, false and a misconception, this is such an efficient way as learners often are interested in resolving the discrepancy [8].

AEINS rephrases the question from the perspective of the learner to provide a meaningful context and facilitate the activation of prior knowledge; this technique has shown its usefulness in the learning process as shown in [38]. For example, if we would like learners to realize the effects of stealing, we could pose the problem of shoplifting and raise the stakes, if necessary, in the form of: what if you (the student)is the owner of this shop.

4.3 Student Model

Student modeling is an important aspect of providing adaptivity. It is currently a simple form of the overlay model represented in the form of rules. AEINS builds a model of the students learning process by observing, analyzing and recording the students actions and choices from the generally accepted ethical views. The student model is represented by rules similar to those of the pedagogical model but associated with certainty confidence.

AEINS is capable of providing a summary at the end of the interaction with the learning environment. This summary is based on the student model, it displays the different positive and negative skills the student have. It also shows the teaching moments the student interacted with and the values the student exposed to associated with confidence factors.

4.4 Presentation Model

Based on Gagne, there are four stages in this model: awaken, explain, reinforce, and transfer, each of them is emphasized in the system as follows: At the awaken stage, the interface itself is designed in a way that captures the students attention, as discussed before in $Fig.2$. The playing characters personalities evolve over time, which make their reactions different every time with respect to their current personality. The variance of the narrative experience itself is engaging and helps in gaining the attention of the student and creates new experiences, based on Gagne and Keller [23]. At the explain stage, feedback and explanations are given to student. This helps the student to reflect on her own actions and their consequences. At the Reinforce and Transfer stages, the student has the freedom to see all the previous history of her actions and other playing characters actions. The student is involved in the moral dilemmas and the consequences depend on her choices and actions. This forces the student to make a conscious choice in terms of ethics.

To interact with the story, AEINS offers a GUI as shown in $Fig.5$ where the student is able to open a list and chooses an action from it; actions include: move, invite, persuade etc. The student is then able to click on one of the characters' and places' pictures in the world. for example, the student chooses *"invite"* action from the list and then clicks on *Ziad's* and *house's* pictures. The end result will be "invite Ziad to pub2. Ziad has the freedom to accept or reject the users invitation according to specific set of rules and constraints describing the tactic the playing-characters can take.

The student gets engaged in a conversation that evolves depending on the students actions. The aim is to enable students to test their own intuitions about certain moral value and to perform arbitrary experiments, in so doing it is believed that students will better understand the nuances of the domain. In addition present the student with good models and examples, hopefully, after which they could model their own behavior.

4.5 Story generation in AEINS

According to [35] planning is efficient and able to generate different narratives for different users, and also different narratives for the single user on subsequent play turns. This technique enhances the user's

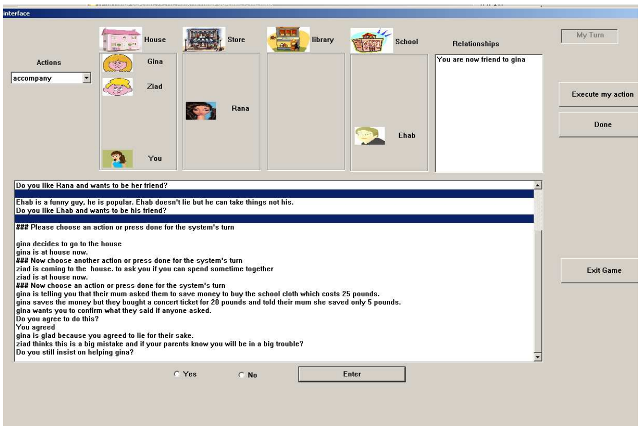


Figure 5. The developed system interface

sense of control in the narrative environment [35]. The main story in AEINS is generated using a STRIPS-like representation planning algorithm, similar to the work of Barber and Kudenko [19], a STRIPS-like representation planning algorithm is used that selects a story event to be executed based on a set of authored story actions. However, unlike the work of Barber and Kudenko, the planning algorithm is more like game-playing algorithm, forward-chaining from the current situation by trying all possible actions from there. The choice of action(s) depends on the nearest to satisfy the narrative goals (preconditions of the teaching moment). The effects indicate how the current situation changes as a result of applying the operator. For every possible way the student can violate the story plan, an alternative story plan is generated.

In a STRIPS-like representation planning algorithm, actions are instances of generic schemata called operators. An operator has preconditions and effects. The preconditions indicate the conditions that must be valid for the operator to be applicable. The effects indicate how the current situation changes as a result of applying the operator. Given a narrative goal (i.e. the pre-condition of the next teaching moment) and the current world state, the story engine selects a story action to execute from the produced plan. Fig.1. presents an example of two action operators, represented with variable argument(s) for which different instances can be substituted.

STRIPS-Like Planning In the STRIPS representation, a problem definition has the following components:

- A finite, nonempty set I of instances.
- A finite, nonempty set P of propositions, which are partial functions of one or more instances. Each application of a proposition to a specific set of instances is called a positive literal. A logically negated positive literal is called a negative literal.
- A finite, nonempty set O of operators, each of which has: 1) preconditions, which are positive or negative literals that must hold for the operator to apply, and 2) effects, which are positive or negative literals that are the result of applying the operator.
- An initial set S which is expressed as a set of literals.
- A goal set G which is expressed as a set of literals.

Although this kind of narrative is used to generate the stories connecting different teaching moments together in a one continuous coherent story, it was practical not to follow the same technique in designing the teaching moments, where it is very important to conserve

action name	*preconditions*	*postconditions*
move(Y,Z)	place(Y), charat(student,Y)	place(Z), charat(student,Z)
invite(X,Y,Z)	char(X), place(Y), charat(X,Y)	place(Z), charat(X,Z)

Figure 6. example of the story world operators

the educational targets. Branching narrative is used to develop stories within the teaching moments as discussed previously in this paper.

4.6 Story World

The story world contains all the information about the characters and the objects, such as their description, location and their state in the game world. The story world is mainly the world current state, and its role is to track and save all the current actions of the student and the agents to be used later by the planner. The world current state is updated after every executed action either done by the student or by one of the agents. The main advantage of having more than one non-playing character is to have the freedom to portray agents who do not share the learner's goals, they can be used to provide negative examples [26]. On the other hand, they can also act according to the moral goals and can give positive examples or help the student to stay on the right track. The story is in effect a narrative describing the story world, the characters' actions and the actions the student is taking and the effect of these actions on the story world.

5 How does AEINS work?

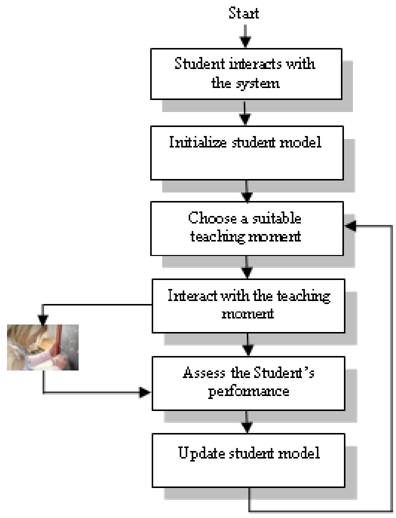


Figure 7. The game model

A model of the game is presented in $Fig.7$. The game starts by presenting the game world to the student. The game gives a brief introduction about the world. The game allows the student to choose his friends to initialize the student model. The pedagogical model chooses the nest teaching moment to present. The student is free to act and the game generates the appropriate narrative according to the student's actions and the specified goals (teaching moment preconditions). Once the preconditions are satisfied, the teaching moment starts and the student starts interacting with it. After finishing the teaching moment the student model is updated, the student regains his freedom to act, and the system generates the narrative accordingly. The cycle continues as shown in the figure.

6 Evaluation

6.1 Analytical Evaluation

AEINS has worked on the drawbacks mentioned in the related work. AEINS succeeded in mixing continuous planning and branching planning approaches. The former was used to generate the outer story that links the teaching moments together and forms coherent continuous story. The latter has been used in structuring the teaching moments that accommodate the Socratic Dialogue. The dynamically generated narrative sustains the freedom of the player and allows him to affect the story and feel control over the environment. The branched narrative helps in preserving the educational goals and allows the cognitive tutor to follow and assess the learner. AEINS succeeded in combining both techniques in a seamless way, where moving from one technique to another is done smoothly and without affecting the learner's experience.

AEINS used the Socratic Dialogue as its teaching pedagogy. In every dilemma, the voice of 'Socrates' comes from one of the involved characters who exhibit certain personality according to their role in this moral situation. This voice usually comes from the learner's friend, this raises conflict moral situations and makes the learner thinks harder to solve the discrepancy. Since, AEINS contains a student model, the feedback provided to the learner is tailored to his history and his current choices.

6.2 Empirical Evaluation

A preliminary evaluation of the AEINS system has been performed: four students aged 8 to 11 years tried the system. The students were of different origins and had different cultural backgrounds, and thus formed a good sample. Responses to the usefulness of AEINS were positive with one exception: "The environment will be more engaging if it contains more animation". The four students felt that AEINS is interesting and the dilemmas were engaging. Verbal statements during and after the session were overwhelmingly positive. One student comments during using the system:"I did not mean to upset my friend, I felt as if it really happened and I had lost my friend who will not talk to me ever again. I think I will be careful next time." Another student comments after the session has ended: "I think this can help me solving school problems".

One of the children asked for password at the very beginning so that no body can access their files and see their actions. This reflects the importance of having such private environment that students can freely act and even purposely did mistakes. All children praised the idea of having the whole experience saved where they can return later and revise it, so they can see what had happened according to their choices and actions. Presently, although these are really encouraging results, the next goal is to develop more substantial evaluation strategies to evaluate AEINS usability and validity.

7 CONCLUSION

This paper presents a framework for using serious games to teach ethics. An educational adaptive system, AEINS, has been implemented. Real life dilemmas and Kohlberg's moral dilemmas are part of the domain and were used to construct the teaching moments. AEINS combines two types of generated narrative; continuous narrative planning and branched narrative. Interactive narrative allows the presentation of ill defined problems, like ethics and citizenship problems and is used to engage the student in the game and allow the student to experiment with various actions in a safe environment. Such interaction aims to provide new thoughts and add new experiences to the student that can be used later or transferred to real life actions.

AEINS incorporates an intelligent tutor that allows monitoring the student and provide personalized learning process and feedback. Two types of student agencies were considered: a high-level agency outside the teaching moments and semi-controlled agency inside the teaching moments. The preliminary evaluation of AEINS showed promising results and we are currently working on a full evaluation with a larger group of children.

REFERENCES

[1] A. Amory, K. Naicker, J. Vincent and C. Adams, The use of Computer Games as an Educational Tool: 1. Identification of Appropriate Game Types and Game Elements, British Journal of Educational Technology 30(4), p. 311-322, 1999.

[2] A. Waraich, Using narrative as a motivating device to teach binary arithmetic and logic gates, SIGCSE Bull.,363, 97101, ACM(Pub.),New York, NY, USA, 2004.

[3] B. Koslowski, Theory and Evidence: The Development of Scientific Reasoning, Cambridge MA: MIT Press, 1996.

[4] B. S. Magerko, player Modeling in the interactive Drama Architecture, Department of Computer Science and Engineering, University of michigan, 2006.

[5] B.W. Mott, C. B. Callaway, L. S. Zettlemoyer, S.Y. Lee, J. C. Lester and M. Rukeyser. Towards narrative-centered learning environments. Narrative Intelligence: Papers from the 1999 Fall Symposium. Menlo Park, CA: American Association for Artificial Intelligence, In M. Mateas and P. Sengers (Edt.), Press(Pub.),7882, 1999.

[6] B. W. Mott and J. C. Lester. U-director: a decision-theoretic narrative planning architecture for storytelling environments, AAMAS 06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, ACM(Pub.) 977984, Hakodate, Japan 2006.

[7] C. E. Watson, Using Stories to Teach Business Ethics Developing Character through Examples of Admirable Actions, *Teaching Business Ethics*, **7** (2), 93–105, Pub. Springer Netherlands, Collection Humanities, Social Sciences and Law, May 2003

[8] D. A. Bergin, Influences on classroom interest, *Educational Psychologist*,**34** (2),87–98, 1999.

[9] D. E. Simpson, Dilemmas in Palliative Care Education, Palliative Medicine Journal,12, 1998.

[10] D. H. Elkind and F. Sweet, The Socratic Approach to Character Education, appeared in Educational Leadership, 1997.

[11] D. H. Elkind and F. Sweet, Good Character, *http://www.goodcharacter.com/*.

[12] D. Kuhn , Science as argument: implications for teaching and learning scientific thinking, *Science Education*,**77**(3), 319–337, 1993.

[13] D. W. Shaffer, K. D. Squire, R. Halverson, and J. P. Gee. Video Games and the Future of Learning. Phi Delta Kappan, 87(2), 104-111, 2005.

[14] D. W. Shaffer (in press). Epistemic frames for epistemic games. Computers and Education. http://coweb.wcer.wisc.edu/cv/papers

[15] D. W. Shaffer, Multisubculturalism: Computers and the end of progressive education. under review by Teachers College Record, 2005. Available at http://coweb.wcer.wisc.edu/cv/papers/multisubculturalism-draft1.pdf

[16] F. H. Van Eemeren, D. N. Walton, C. A. Willard, J. Woods and D. Zarefsky. Fundamentals of argumentation theory: A handbook of historical backgrounds and contemporary developments. Mahwah, NJ: Lawrence Erlbaum Associates, 1996.

[17] G. Bolton. Acting in classroom drama: A critical analysis. Pub. Heinemann, London, 1999

[18] H. A. Simon.The structure of ill-structured problems.AI, 4:181201, 1973.

[19] H. Barber and D. Kudenko. Adaptive Generation of Dilemma-based Interactive Narratives Advanced Intelligent Paradigms in Computer Games Series: Studies in Computational Intelligence, Springer,71, 2007

[20] H. C. Lane, M. G. Core, D. Gomboc, A. Karnavat and M. Rosenberg ,Intelligent tutoring for interpersonal and intercultural skills,Interservice/Industry Training, Simulationand Education Conference (I/ITSEC), 111, 2007.
[21] http://www.goodcharacter.com
[22] H. Vilhjalmsson and C. Merchant and P. Samtani, Social Puppets: Towards Modular Social Animation for Agents and Avatars, Lecture Notes in Computer Science, Springer Berlin / Heidelberg (Pub.), I(4564/2007), Online Communities and Social Computing, 192201, Friday, August 24, 2007.
[23] Instructional Design, *http://www.nwlink.com/ Donclark/hrd/learning*.
[24] J. L. McBrien and R. S. Brandt, The Language of Learning: A Guide to Education Terms, Association for Supervision and Curriculum Development, Alexandria, 17–18, 1997.
[25] J. L. McGrenere, Design: Educational Electronic Multi-Player Games ALiterature Review, Dept. Of computer Science, the University of British Columbia, June 1996.
[26] J. M. Thomas and M. Young, Becoming Scientists: Employing Adaptive Interactive Narrative to Guide Discovery Learning, AIED-07 Workshop on Narrative Learning Environments, lMarina Del Rey, California, USA, 2007.
[27] J. M. Randel, B.A. Morris, C.D. Wetzel, and B. V. Whitehill. The effectiveness of games for educational purposes: A review of recent research. Simulation and Gaming 23, 3, p. 261-276, 1992.
[28] J. P. Gee: Learning by design: Good Video games as Learning machines, ELearning, Vol. 2, Number 1,2005.
[29] J. P. Gee, Why are video games good for learning?, available at www.academiccolab.org/resources/documents/MacArthur.pdf
[30] J. Tan, C. Beers, R. Gupta, and G. Biswas, Computer Games as Intelligent Learning Environments: A River Ecosystem Adventure. Artificial Intelligence in Education, C.-K. Looi et al. (Eds.), IOS Press, 2005
[31] K. Maragos, M. Grigoriadou, Towards the design of intelligent educational gaming systems, AIED workshop5, 12th International Conference on Artificial Intelligence in education, Amsterdam, 2005.
[32] M. A. Gmez-Martn, P. P. Gmez-Martn, and P. A. Gonzlez-Calero, Game-Driven Intelligent Tutoring Systems, M. Rauterberg (Ed.): ICEC 2004, LNCS 3166, pp. 108113, 2004.
[33] M. Fasli, M. Michalakopoulos, Supporting Active Learning through Game-Like Exercises, icalt, pp. 730-734,. Fifth IEEE International Conference on Advanced Learning Technologies (ICALT05), 2005
[34] M. Klawe, When Does the Use of Computer Games and Other Interactive Multimedia Software Help Students Learn Mathematics?, Department of Computer Science, the University of British Columbia, 1998.
[35] M. O. Riedl and R. M.Young, From Linear Story Generation to Branching Story Graphs, IEEE Comput. Graph. Appl., 263, 2331, IEEE Computer Society Press(Pub.), Los Alamitos, CA, USA, 2006.
[36] M. Riedl and A. Stern, Believable Agents and Intelligent Story Adaptation for Interactive Storytelling, 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment, Darmstadt, DE, 2006.
[37] P. Bell and M. C. Linn, Scientific arguments as learning artifacts: Designing for learning from the web with KIE, *International Journal of Science Education*,**22**(8), 797–817, 2000.
[38] R. C. Anderson and J. W. Pichert , Recall of previously unrecallable information following a shift in perspective,*Journal of Verbal Learning and Verbal Behavior*, **17**,1–12, 1978.
[39] R. Hodhod and D. Kudenko , Interactive Narrative and Intelligent Tutoring for Ill Defined Domains, In proceedings of a workshop held during ITS-2008: ITSs for Ill-Structured Domains Focusing on Assessment and Feedback. The 9th international Conference on Intelligent Tutoring Systems, Montreal, Canada, June 23-27, 2008.
[40] R. Prada, I. Machado and A. Paiva. Teatrix: A Virual environment for story Creation, Intelligent Tutoring Systems, Springer (Pub.), G. Gauthier and C. frasson and K. van Lehn (edts.), 2000.
[41] S. Egenfeldt-Neilson, Beyond Edutainment: Exploring the educational potential of computer games, Submitted to the IT University of Copenhagen as partial fulfilment of the requirements for the PhD degree, February 2005.
[42] S. McQuiggan and J. Rowe and S. Lee mand j. Lester, Story-Based Learning: The Impact of Narrative on Learning Experiences and Outcomes, Proceedings of the Ninth International Conference on Intelligent Tutoring Systems(ITS-2008),Montreal, Canada, 2008.
[43] V. Bayon, J. R. Wilson, D. Stanton, and A. Boltman. Mixed reality storytelling environments. JVirtual Reality Journal, Springer London (Pub.), textbf71, 2003.
[44] W. G. Harless. An Interactive Videodisc Drama: The Case of Frank Hall, Journal of Computer-Based Instruction. http://www.idrama.com/Media

Appendix

This appendix shows an example of a complete run. The system introduction and the teaching moments are written in normal font. The student's actions are in bold. The planner actions are in italic. My illustration to some actions will be in capital letters
THIS IS A BRIEF INTRODUCTION ABOUT THE GAME WORLD.
This is your world, please have a look around! You have four places to go to: the house, the shop, the library and the school. There are four characters from whom you can make friends. You have a list of actions to choose your actions from. You are free to act whenever you wish You will be involved in some situations during your play time, TRY TO BE YOURSELF!

Now, it is time to choose your friends.....

Gina is a nice girl, she is sincere. Gina does not accept taking things without permission but she can lie Do you like Gina and wants to be her friend?
yes
You agreed
Ziad is a beloved boy, he is a good and sincere boy but sometimes he can cheat. Do you like Ziad and wants to be his friend?
yes
You agreed
Rana is a beautiful girl, she does not accept lying or taking things without permission. Rana is not sincere to her friends. Do you like Rana and wants to be her friend?
no
You disagreed
Ehab is a funny guy, he is popular. Ehab does not lie but he can take things not his. Do you like Ehab and wants to be his friend?
yes
You agreed
Please choose an action or press done for the system's turn

THE STUDENT'S CHOOSES TO PRESS THE SYSTEM'S TURN BUTTON
gina decides to go to the store
gina is at store now.
Now choose another action or press done for the system's turn
THE SYSTEM CHOOSES THE ACTION FRIEND, AND CHARACTER RANA.
You want to be rana's friend.
rana likes you and wants to be your friend too.
rana is calling you to come and meet them at the store. Do you like to go?
yes
You agreed to join rana.
Now choose another action or press done for the system's turn

THE STUDENT'S CHOOSES TO PRESS THE SYSTEM'S TURN BUTTON
THE PRECONDITIONS OF THE FOLLOWING TEACHING MOMENT HAVE BEEN SATISFIED, SO THE TEACHING MOMENT STARTS.
gina is looking around in the store, they saw the kind of chocolate they like. gina has no enough money to buy the chocolate gina has an idea!! they can take the chocolate if you helped them to distract the shop keeper attention Do you think you should help your friend to take the chocolate bar?
yes
You agreed
You helped gina to take the chocolate. rana does not like what you did rana thinks this is not good. Do you agree?
no
You disagreed
rana says that it is really dangerous to take stuff without permission, you will be a law breaker, do you agree??
yes
You agreed
rana asks if you agree that people who take things which are not theirs deserve to be punished
yes
You agreed
rana says that they are glad that you agreed that taking people stuff without permission stealing and it harms them Do you want to tell gina to return back the chocolate
yes
You agreed
rana admires what you did, the shopkeeper caught gina and will punish him for what they did.
Please choose an action to perform or press done for the system's turn
THE STUDENT'S CHOOSES TO PRESS THE SYSTEM'S TURN BUTTON
gina is telling about the big teddy bear they saw.
rana is telling about the big teddy bear they saw.
Your friends says you are too late and have to hurry to the school.
Your friends say that the football match is about to start.
You are at the school
Now choose an action or press done for the system's turn
THE STUDENT'S CHOOSES TO PRESS THE SYSTEM'S TURN BUTTON
gina decides to go to the school
gina is at school now.
Now choose another action or press done for the system's turn
rana is going to school. to attend the school assembly's training
rana is at school now.
Now choose an action or press done for the system's turn

THE STUDENT'S CHOOSES TO PRESS THE SYSTEM'S TURN BUTTON
THE PRECONDITIONS OF THE FOLLOWING TEACHING MOMENT HAVE BEEN SATISFIED, SO THE TEACHING MOMENT STARTS.
gina is telling you that their mum asked them to save money to buy the school cloth which costs 25 pounds. gina saves the money but they bought a concert ticket for 20 pounds and told their mum she saved only 5 pounds. gina wants you to confirm what they said if anyone asked. Do you agree to do this?
yes
You agreed
gina is glad because you agreed to lie for their sake. rana thinks this is a big mistake and if your parents know you will be in a big trouble? Do you still insist on helping gina?
yes You disagreed
rana is glad and said that you did the right thing. rana agrees with you in that friendship does not mean to accept to lie for your friend's sake! gina is feeling shy of themselves and left the place.
Please choose an action to perform or press done for the system's

turn

THE STUDENT'S CHOOSES TO PRESS THE SYSTEM'S TURN BUTTON
gina is at school. They are moving to their class now rana is at school. They are moving to their class now
Please choose an action to perform or press done for the system's turn

THE STUDENT'S CHOOSES TO PRESS THE SYSTEM'S TURN BUTTON
THE PRECONDITIONS OF THE FOLLOWING TEACHING MOMENT HAVE BEEN SATISFIED, SO THE TEACHING MOMENT STARTS.
Math is your difficult subject, Your teacher will be announcing last exam's grades today. You know that even after adding your half term scores, your grade will be 'C' at the end. The teacher announces the grades and you got 'A'. You are very excited about this, WOW, looks like you will have a present from your parents. The teacher now announces your colleague's grade, who is really good at Math. The announced grade is 'C'. Your colleague is surprised and very upset. You went home and thought of what had happened. Suddenly, you realized that it should be that the teacher has swapped the results unintentionally between you and your colleague You called your friends to talk about what happened gina is saying that this a good opportunity, you should not tell anyone about this mistake. Do you agree with gina?
yes
You agreed not to tell the truth
rana thinks this is cheat, and does not like your behavior rana thinks this is unfair for your colleague Do you think rana is right?
no
You disagreed
rana is very surprised because you don't want to tell the truth rana asks: what if this happens to you and someone else took your good grade Will you sill think the same way?
no
You disagreed
So you think rana is right. Are you going to tell your teacher about the confusion happened?
yes
You agreed
You decided to tell your teacher the truth.
You teacher is very proud of you, and your colleague is appreciating what you have done. Your teacher told your parents and they are also very proud of you, as a result of your honesty They will allow you to travel in the summer holiday rana is happy for you and is telling you that good behavior is much more important than grades It relieves the conscious and make you beloved by everyone.
Please choose an action to perform or press done for the system's turn

rana is at school. They are moving to their class now.
gina is at school. They are moving to their class now.
gina is asking you to move to the class now.
Now choose an action or press done for the system's turn

THE STUDENT'S CHOOSES TO PRESS THE SYSTEM'S TURN BUTTON
rana is at school. They are moving to their class now
Now choose an action or press done for the system's turn

THE STUDENT'S CHOOSES TO PRESS THE SYSTEM'S TURN BUTTON
THE PRECONDITIONS OF THE FOLLOWING TEACHING MOMENT HAVE BEEN SATISFIED, SO THE TEACHING MOMENT STARTS.
rana wants to go and get their snack, they asked you to keep an eye on their comic book gina seems to like rana's comic book, they asked you to give them the book Do you agree to give gina the comic book?
yes
You agreed
gina took the book. rana comes back and asks for their comic book. rana is very upset and said you were not keen enough on their stuff. Do you want to go and ask gina to return the book now?
no
You disagreed
rana is really getting more and more upset. rana asks you to buy them another comic book, Do you agree?
yes
You agreed
rana is really glad now because you agree to get them another comic book. rana hopes that you are going to be keener on other's things from now on.
You did a good job in this level. Please press the save button to save the whole experience.

Towards the automatic invention of simple mixed reality games

Robin Baumgarten, Maria Nika, Jeremy Gow and **Simon Colton**[1]

Abstract. The invention of mixed reality games that combine virtual and physical play offers a rich and challenging application area for AI techniques. We look at the possibility of using descriptive machine learning to automatically invent simple mixed reality games. Specifically, we demonstrate that the HR learning system can generate coherent domain knowledge from the noisy play data gathered from a number of simple physical games. We describe how this could be used to support mixed reality game invention, and discuss the prospects for further work in this area.

1 Introduction

Using AI techniques for game design is not nearly as well researched as using AI for avatars and for non-player characters, even though there is clearly potential to enhance the creativity of game designers. We look here at the possibility of using a descriptive learning approach to automatically invent simple mixed reality games.

Descriptive learning allows interesting concepts and properties of a domain to be discovered from observations, without being restricted to any particular learning goal. Applied to games, it has the potential to automatically discover game-specific domain knowledge (rules, strategies etc.) from observed play. This knowledge could help artificial agents fill a number of roles, without the necessity for providing game knowledge to the agent ahead of play. These include:

Game Player So-called *general game playing* agents can play unseen games without being told game rules or strategies [8].

Game Mediator Agents could mediate play between humans, e.g. taking on the role of a referee or coach.

Game Inventor Domain knowledge could be used as a basis for constructing new rule sets.

The advantage of using descriptive rather than predictive machine learning (see section 2) is that there is no specific goal, and we can simultaneously find hypotheses describing the environment and the particular game being played, which allows a greater understanding to be developed, e.g. to support game invention.

In addition, descriptive learning systems can start with background concepts but no data, and — through the use of third party systems such as constraint solvers, model generators and computer algebra systems — can invent concepts and flesh them out with examples [1]. Such abilities would allow agents to operate in a wider range of applications, e.g. social environments where humans are creating, playing and developing their own games. We envision agents that can join in such social play as a player, mediator or inventor.

[1] The Computational Creativity Group, Department of Computing, Imperial College London, UK. Contact email: sgc@doc.ic.ac.uk

Mixed reality games present a suitable domain for this approach, because the computer is already a natural part of the game, and would not have to be artificially introduced into play. However, they also present a challenging domain, partly because the physical data can be complex and noisy, and partly because of the typical complexity of the game mechanics

We demonstrate here that the HR system [2] is capable of extracting sensible domain knowledge from physical play data, obtained from location tracking of two players engaged in relatively simple physical games, such as Tag (sections 3 and 4). Applying descriptive learning to physical data is an essential first step for the invention of mixed reality games. We then argue (in section 5) that this knowledge can be used to invent new games, as well as discussing other directions for this work.

2 Background

2.1 Descriptive learning & HR

In a *descriptive* machine learning setting, an agent attempts to discover a theory that describes a data set. The theory can consist of example objects, concepts which categorise examples, conjectures which make claims about concepts, and explanations which support conjectures. This exploratory behaviour lacks a specific goal, and can be contrasted with predictive learning where the goal is to solve a specific categorisation problem. Logic-based descriptive learning systems include HR [2], CLAUDIEN [13] and WARMR [7].

HR is a theory formation system which generates a theory starting from an initial collection of example objects, in addition to a set of initial concepts and a set of axioms which relate the concepts, usually expressed in first-order logic. New concepts are constructed from the existing set using production rules, employing heuristic search based on various measures of interestingness [5] to control exploration of the concept space. HR has 17 production rules which each form a new concept by various syntactic manipulations and combinations of existing concept definitions. The production rules that we used in the application here were:

Compose Conjoins the literals of two input concepts.
Exists Abstracts ground values to existential variables.
Match Unifies distinct variables within a single input concept.
Negate Negates literals within a definition.
Size Counts the size of the success set of a clause.
Split Instantiates variables in a definition.

Conjectures are formed by HR during the concept search, by observing patterns in the sets of known examples that the concepts apply to. For instance, if HR noticed that the example set of a newly-formed concept was exactly the same as that of a previously defined concept, it would make an *equivalence* conjecture stating that the two definitions are logically equivalent. Conjectures can be proved from known axioms and theorems either internally or using a third-party automated theorem prover: this can add theorems to the theory or, if the proof is based on a very simple subset of background knowledge, it can be used to remove trivial conjectures from the theory.

Note that in domains where the data may be noisy, HR is able to make *near-equivalences*, i.e., equivalence conjectures where the truth of the conjecture is only partially supported by the data. The user is able to set a parameter for the minimum fidelity of conjectures (usually in the range 60-80%). For instance, if the user set the value to be 75% and HR reported the conjecture that $A \leftrightarrow B$, then this means that, of all examples which satisfy either property A or property B, at least 75% of them satisfy both properties. The user is also able to specify that the calculation of the fidelity is carried out on only the positive examples of the concepts. This tends to avoid the reporting of near-equivalences between concepts for which the examples are mainly negative, for instance the false conjecture in number theory that the concept of square numbers is equivalent to the concept of prime numbers. While there is no overlap in the positive examples of these concepts, the sparsity of examples on the number line mean that this conjecture has 65% fidelity over the numbers 1 to 100.

As mentioned above, HR's search is driven by heuristic measures of interestingness. These measures can also be used to filter and sort the concepts and conjectures in HR's output. The two measures we use here are *applicability* and *fidelity* of conjectures. Applicability is defined as the number of examples that a conjecture relates. Hence, conjecture about even prime numbers score very low for applicability, as they only describe the number 2. Fidelity is measured as the proportion of examples that support a conjecture, for instance the conjecture that prime numbers are odd, while false, scores highly for fidelity, because it is nearly true — with only one exception.

HR has been applied to a variety of domains, most notably mathematical domains where it has been used to make some interesting discoveries [6]. Of particular relevance to mixed reality games is the extension of HR to work with noisy data in order to learn about the rules of a dice game from vision data [14].

2.2 Mixed reality games

Mixed reality games combine physical and virtual elements in gameplay, and research interest in them has been growing over many years as supporting technologies become more sophisticated and readily available. Early examples included ARQuake [15], an augmented reality version of the first-person shooter Quake, and Mixed Reality Pong [10], where a virtual ball is project onto a tabletop and any physical object can be used as a bat.

Research has expanded to cover mobile mixed reality games (e.g. Phone Tennis [9]) and serious games like Virus Life [12], in which players 'clean' a room to defend territory against a spreading virtual virus that simulates hospital infections. Commercial mixed reality games have now begun to be released, such as Eye of Judgement for Playstation 3. Modern consumer hardware, like the Wii, is better able to support mixed reality games with movement sensors and cameras. Other supporting technologies, such as interactive displays, are gradually becoming more commonplace, e.g. Microsoft Surface [11]. Hence there is great potential for the popularity of mixed-reality games to grow over the next few years.

3 Mining conjectures from physical data

We took a three stage approach to generating domain knowledge from observations of physical game playing:

Data gathering Play data was collected from multiple rounds of several games (section 3.1).

Data encoding Logic based descriptive learning systems, like HR, require input in the form of logical statements. For each game, the play data was encoded as a set of ground first order predicates. These predicates were hand-crafted to describe the physical domain (e.g. relative positions in physical space), but are not game-specific (section 3.2).

Descriptive learning We used HR to form a theory about the data in the given encoding. HR's theory investigation tools to help us identify the most interesting conjectures which described the actions of the players in the game (section 3.3).

The approach is independent of the games studied here, and could be generalised to other game domains — providing suitable data encodings can be designed.

3.1 Data gathering

The Ubisense location tracking system [16] was installed in a medium sized room (approx. 10m by 6m). Each player carried a tracking 'tag' with a single button which they could use to provide additional play data (see Figure 1). To increase the accuracy of the location tracking, the players walked rather than ran, and players remained in sight of the location sensors. Because of these artificial constraints, the games were more simulated than played, although they were still engaging physical activities for the players involved. A more sophisticated approach (e.g. with better tracking technology) might remove these artificial constraints.

We chose three simple physical games, plus one structured physical activity:

Tag One player attempts to catch the other, and when caught they swap roles. Both players constantly clicked their button to indicate they were still more than one metre apart. A 'tag' was indicated by the players temporarily ceasing the clicking.

Easter Egg Hunt A third tag was placed in the room, and the players 'searched' for it: for practical reasons it was actually in sight of the sensors. A player would click the button to indicate they had found the tag, bringing the game to an end.

Human Pong As a simulation of a mixed reality Pong [10], we used a third tagged person as a ball, with the two players acting as bats. The aim of the game is to keep batting the ball back to your opponent. The players used their tag buttons to indicate they were batting the ball, with the 'ball' person confirming this with their own tag button.

Walls Three players moved around the room, and whenever two were near a wall, the third person would click their tag button. This was a structured activity, rather than a game, but served as a good training domain for HR.

The Ubisense installation consisted of a network of four sensors connected to our existing standard network infrastructure, three palm-sized tracking tags (see Figure 1) and a PC running tracking server software to collect data from the sensors. The installation of the system requires careful distribution and calibration of the sensors, especially in complex indoor layouts.

Figure 1. A Ubisense tag. Players used the tag button to record game events.

The sensors record the time and angle of arrival of UWB radio pulses from specific tags, enabling the system to compute each tag's 3D position up to 20 times a second with up to 15cm accuracy. Ubisense does not require optical line-of-sight, but the radio signals are attenuated by water so the human body can cast a solid radio shadow. Thus, detecting and tracking humans effectively requires multiple well-positioned sensors.

Data from physical games is noisy due to the difficulties of accurate location tracking and the possibility that players may violate rules. For our setup, the level of tracking noise depends on the obstruction of the transmitter units from the four receivers. While the system has an average resolution of 15cm, occasional accuracy fluctuations affect the system. The largest inaccuracy we recorded was a movement of about 2 meters within one second. To reduce the impact of these spikes, we applied a central moving average to the data, weighted by the temporal proximity to the current data point.

3.2 Data encoding

For each tag, the location tracking system records a series of points in a real-valued 3D coordinate system with timings and orientation, along with timings of the tag button presses. For our analysis, we ignored the height from the ground as well as the orientation of the sensor.

In order to have HR learn about the games, we discretised this play data and encoded it as sets of predicates from which the system could make conjectures about common patterns. The axioms only represent information about relative locations, which has two advantages over an absolute approach: a) it is independent from the dimensions of the environment, and b) we do not have to worry about the numerical representation of locations in first order logic.

Each game session is encoded using the following predicates:

- `event(E)`: A point in time when a player presses a tag button.
- `player(P)`: Name of a player.
- `wall(W)`: Name of a wall.
- `event_of_player(E,P)`: Identifies the player who caused an event.
- `near_time(E1,E2)`: These two events were less than 2 seconds apart.
- `near_player(E,P1,P2)`: At the event the two players were less than 1.5 metres apart.
- `near_wall(E,P,W)`: At the event the player was within 1.5 metres of the wall.
- `happens_before(E1,E2)`: The first event happened before the second.

3.3 Descriptive learning

We used data from the Walls activity to determine the correct combination of production rules, measures of interestingness and search parameters to maximise HR's chances of finding conjectures of interest. We then used the same setup for the other three physical games. This adds some credence to our claim that our approach can mine interesting conjectures from physical play data for a range of different games.

The data recorded by the location tracking system is not perfect, so we configured HR to form near-equivalence conjectures with a fidelity threshold of 80% correctness. We established this threshold by running a couple of test runs of the tracking system: it was a good balance between reducing noise and preventing HR from excluding less common events. We employed HR's *compose*, *negate*, *exists*, *size* and *split* production rules. The exact choice of rules determines the concepts that HR will generate, and this is a typical initial selection. However, other configurations of the 17 rules are possible which might generate richer concepts at the expense of a larger search space. For example, we could have used the *match* rule that equates two previously distinct objects (e.g. a concept about two objects becomes a concept about one), or the *forall* rule that establishes a relation between an object and all other objects (e.g. something that is larger than everything else). Further work could explore the effectiveness of different rule sets in this domain.

We ran HR for 2000 theory formation steps, each of which results in either a concept or a conjecture being formed, and we examined the resulting conjectures. In particular, we first sorted the conjectures in terms of an equally weighted sum of their applicability and fidelity. We then cross-referenced the conjectures, so that we could identify conjectures which related particular concepts, for instance concepts which include the `event_of_player(E,P)` predicate in their definition. While we still had to look through a number of conjectures which were not interesting, we found that we were able to fairly easily identify some conjectures which captured aspects of the physical games.

4 Illustrative Results

In Figure 2, we present visualisations of the tracking data for an individual round of the four games. The noise in the player's path data is apparent in the fact that the lines are not smooth. Note that some of the larger features of the lines are also due to tracking inaccuracies rather than player movement. HR creates a large number of conjectures, depending on the number of theory formation steps employed. However, after sorting them using the weighted sum described above, the results we present below were usually found near the top of the list, mixed with less interesting (i.e. more obvious) results. However, in a few cases, a more prolonged search was required.

Noise in the test data set caused conjectures to be generated that did not reflect the (intended) rules of the game or were artifacts of too little training data. This could be mitigated by increasing the sample size and the accuracy of the tracking system, e.g. one could make sure the sensors are not blocked by objects such as furniture.

In the following sections the two players are denoted a and b. For bound variables, we use p and q to denote players, e and f to denote events and u and w to denote walls.

4.1 Tag

In this game, a difficulty for HR is that the player is only caught ('tagged') once at the end of the round. This may be seen as an error in the data by HR when it formulates approximate conjectures. For example, it conjectures that all events were caused by the hunting player. Nevertheless, some interesting conjectures were found:

- When the event is caused by the hunted (player b), there are two players near each other. As there are only two players in the game, this means that the hunted has been caught:

$$\forall e.(event\_of\_player(e,b) \leftrightarrow \exists p,q.near\_player(e,p,q))$$

- When two events happen nearly at the same time, one of them is caused by the hunted (b). This means the hunted has been tagged:

$$\forall e.(\exists f.near\_time(e,f) \leftrightarrow event\_of\_player(e,b))$$

4.2 Easter Egg Hunt

As with Tag, the game structure was difficult for HR to work with, as it is only over once the player reaches the egg. This event only happens once, while the hunt takes longer. Thus there are a lot of negative examples and only very few positives, which result in near-equivalence conjectures that state that the egg is (almost) never found. Once we ignore these however, useful conjectures can be found:

- Whenever the event is caused by the egg, it is near a player (that holds because an event can only be caused by one player):

$$\forall e,p.(event\_of\_player(e,egg) \rightarrow \\ (event\_of\_player(e,p) \leftrightarrow \exists q.near\_player(e,q,p)))$$

- Whenever the egg is detected by player b there is also another event at nearly the same time. This indicates that a player pressed the button because the egg was found:

$$\forall e.(event\_of\_player(e,b) \rightarrow \exists f.near\_time(e,f))$$

4.3 Human Pong

A difficulty here was the exceptionally high rate of trivially true conjectures. For example, for all events e, there exists an event f such that e happens before f. While this is not true for the last event, it is valid for all other events and thus matches most of the experimental data. Note that the ball is actually a player in this experiment, which is reflected in the formalisation. We found the following conjectures in HR's output:

- Whenever there is an event, nobody is near player a if and only if somebody (i.e. the ball) is near b. In other words, whenever a button is pressed the ball is near one of the players:

$$\forall e.(\nexists p.near\_player(e,p,a) \leftrightarrow \exists q.near\_player(e,q,b))$$

- Successive events are not both caused by the ball. This is due to both the ball and the player close to it pressing their buttons at the same time:

$$\forall e,f.(happens\_before(e,f) \rightarrow \\ \neg(event\_of\_player(e,ball) \wedge event\_of\_player(f,ball)))$$

- When two events happen at nearly the same time, the first event will be caused by the ball. This is an effect of the discretisation:

$$\forall e,f.(happens\_before(e,f) \rightarrow \\ (event\_of\_player(e,ball) \leftrightarrow near\_time(e,f)))$$

- When two events happen at nearly the same time, either a or b is close to another player (which must be the ball, according to the first conjecture):

$$\forall e,f.(near\_time(e,f) \rightarrow \\ \nexists p.near\_player(f,p,a) \leftrightarrow \exists q.near\_player(f,q,b))$$

- Whenever a button is pressed, two players are close to each other (recall that one of the 'players' represents the ball):

$$\forall e.\exists p,q.near\_player(e,p,q)$$

4.4 Walls

HR found two conjectures that are very close to the 'rules' of the Walls activity. Firstly, one player clicks (i.e. there is no second event at nearly the same time) iff exactly two people are at the wall:

$$\forall e.(|\{p : \exists w.near\_wall(e,p,w)\}| = 2 \leftrightarrow \nexists f.near\_time(e,f))$$

Secondly: multiple players click iff three players are at the wall:

$$\forall e.(|\{p : \exists w.near\_wall(e,p,w)\}| = 3 \leftrightarrow \nexists f.near\_time(e,f))$$

Other conjectures HR found (with similar applicability and matching examples) describe side-effects of the above rules, or coincidences in the data. For example:

- At all events, there was never exactly one person at a wall:

$$\forall e.|\{(p,w) : near\_wall(e,p,w)\}| \neq 1$$

- Whenever a player presses a button, he is only standing at one wall or at no wall (u denotes a wall):

$$\forall e.|\{p : event\_of\_player(e,p) \wedge near\_wall(e,p,w)\}| = \\ |\{(b,u) : near\_wall(e,b,u) \wedge event\_of\_player(e,b)\}|$$

- Whenever more than one player presses the button, each player stands near exactly one wall:

$$\forall e.|\{(p,w) : near\_wall(e,p,w) \wedge event\_of\_player(e,p)\}| = 1 \\ \leftrightarrow \exists p.near\_time(e,p)$$

The latter two are coincidences of the play data, as it is also possible for players to stand in corners.

5 Future work

5.1 Inventing mixed reality games

The rules recognized by HR for these games can also be used to create games for players. By combining the output from HR when applied to the analysis of different games and using this as input data for a new HR session, we speculate that the system can be used to create new games. This could be achieved by forming new theories of games using the existing conjectures and concepts as background knowledge. The created game rules can then be used to guide movements of human players, with the goal of the human players being to guess the intentions of the computer, reversing the role of creator and learner. In particular, we envisage the following approach to the invention of guessing games:

- HR is used to produce theories about various mixed reality games, given data about the movement and actions of players (as above).
- From these theories, we extract two types of conjectures which are supported (at least partially) by the data. Firstly, conjectures which are true in multiple games. These are likely to be axioms of the physical environment, e.g. that a person cannot be close to three or more walls at the same time. Secondly, conjectures which are true only of an individual game. These are likely to contain concepts which can be used as ingredients in novel games, e.g., being near a wall, or clapping twice, etc.
- A new HR session is started, with the same background concepts as in the previous sessions, but without data for any of the concepts. In this mode, HR requires a mechanism for generating data to illustrate new concepts. (E.g. in [3] number theory concepts are given as background information without data and the Maple computer algebra system is used to generate data for new concepts.) In our context, we could use a constraint solver (as in [1]): whenever HR invents a concept, the constraint solver will be employed to generate data for it. The conjectures extracted from individual games will hopefully provide interesting ingredients for novel games.
- We can use our axioms of the physical environment to preprogam the solver with appropriate automatically generated constraints (see [1]) about the physical world, to prevent it from inventing physically impossible concepts.
- HR will produce a general theory of mixed-reality games, which will include various concepts which are physically possible. As in [4], we will enable HR to extract from the theory a set of mutually-possible concepts. The conjunction of these concepts will express a pattern of movements/actions for players in a mixed-reality game which is large enough to embed a pattern which is neither too obvious nor too convoluted. We envisage much experimentation in order to determine a suitable balance.
- Given the concept to embed in a series of movements and actions, we will employ a constraint solver to generate such a series which upholds both the physical axioms and the properties expressed in the concept. These will be given concretely as a set of timings for movements and actions for a set of players. The purpose of the game will be for the players to attempt to determine the underlying pattern that they are expressing, i.e., a guessing game.

Obviously, there is much work to do to in order to achieve such invention of mixed-reality guessing games. However, we believe that such an approach to inventing guessing games is certainly possible, and we plan to implement the methods required to achieve it.

5.2 Improvements to descriptive learning

A common challenge faced when applying HR to a new domain is the large number of uninteresting conjectures made along with the more relevant results. As mentioned above, we encountered a similar problems in this work. For example, HR correctly picks up uninteresting physical constraints that are independent of the game being played, e.g. a player cannot be near three walls at the same time. A typical solution that we could try is to filter trivial conjectures using a theorem prover: if a conjecture can be easily proved from a basic domain knowledge base (e.g. about players and walls) then it is removed, as per the application to number theory described in [3]. Alternatively, as described above, we could compare the conjectures generated from multiple different games, and assume that any conjectures appearing in multiple games actually describe axioms of the physical world (hence are not particularly interesting) rather than aspects of the game being played.

It is unclear how results from our approach could be more formally evaluated, other than simply reflecting on how well they describe the games. One approach might be to compare generated knowledge with first order logic versions of the intended game rules and known player heuristics. A theorem prover could be used to establish implication or equivalence between subsets of human- and machine-generated domain knowledge. This raises more general questions about how the system itself could distinguish between rules, player heuristics and coincidences.

5.3 Other applications

Apart from invention, mixed-reality games could benefit in other ways from knowledge about rules and theories concerning the behaviour of human players. Such knowledge could be used to guide computer players or computer-mediated play between humans. Learning from physical game data is also of interest itself, and descriptive learning in this domain has potentially interesting applications, e.g. coaching in sports education and training. The presented approach could be applied to a wider spectrum of physical games.

Having demonstrated early results in the physical domain, we are hopeful that our approach will have applications in other game domains. We are currently working on applying descriptive learning to combinatorial board games, where generated domain knowledge can be used to improve the performance of General Game Playing agents which can play unseen games without needing to be told domain-specific strategies [8]. Another potential application is video games, to facilitate intelligent game adaptation based on automatic analysis of player behaviour.

6 Conclusion

We have shown that it is possible to learn domain knowledge about physical games through a combination of a location tracking system, a discretisation algorithm and a descriptive machine learning system. HR was able to describe the physical activities after the data had been filtered with a discretisation process, which abstracted the data into a relative and environment-independent form. The setup of HR to handle noisy data is not trivial — for Pong and Easter Egg Hunt especially we encountered a large number of conjectures that were not interesting with respect to the games themselves. We discussed how to alleviate this by modifying the data representation and by introducing additional filtering mechanisms that use background information about the environment to remove unwanted conjectures.

We also discussed a way to use these uninteresting conjectures as axioms of the physical world in which the games are played.

While the work presented here is somewhat preliminary, and the automatic invention of mixed reality guessing games will require much additional work, we hope to have demonstrated the principal that raw data from a physical games can be turned into descriptions of the environment and the games being played. Looking at the nature of the raw data in Figure 2, we believe it is an achievement — albeit modest — to have succeeded in the first stage. In the second stage, we hope to demonstrate the potential for descriptive machine learning systems to innovate in game design — firstly with simple physical games, and eventually with fully featured mixed reality games.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their comments. This work is partly funded by EPSRC grant TS/G002835.

REFERENCES

[1] John Charnley, Simon Colton, and Ian Miguel, 'Automatic generation of implied constraints', in *ECAI 2006, 17th European Conference on Artificial Intelligence*, eds., G. Brewka, S. Coradeschi, A. Perini, and P. Traverso, pp. 73–77. IOS Press, (2006).

[2] Simon Colton, *Automated Theory Formation in Pure Mathematics*, Springer-Verlag, 2002.

[3] Simon Colton, 'Automated conjecture making in number theory using HR, Otter and Maple', *J. Symbolic Computation*, **39**(5), 593–615, (2005).

[4] Simon Colton, 'Automatic invention of fitness functions with application to scene generation', in *Applications of Evolutionary Computing, EvoWorkshops 2008*, ed., M.Giacobini et al., volume 4974 of *LNCS*, pp. 381–391. Springer, (2008).

[5] Simon Colton, Alan Bundy, and Toby Walsh, 'On the notion of interestingness in automated mathematical discovery', *Int. J. Human-Computer Studies*, **53**(3), 351–375, (2000).

[6] Simon Colton and Stephen Muggleton, 'Mathematical applications of Inductive Logic Programming', *Machine Learning*, **64**(1–3), 25–64, (2006).

[7] Luc Dehaspe and Hannu Toivonen, 'Discovery of frequent DATALOG patterns', *Data Min. Knowl. Discov.*, **3**(1), 7–36, (1999).

[8] Michael R. Genesereth, Nathaniel Love, and Barney Pell, 'General game playing: Overview of the AAAI competition', *AI Magazine*, **26**(2), 62–72, (2005).

[9] Anders Henrysson, Mark Billinghurst, and Mark Ollila, 'Face to face collaborative AR on mobile phones', in *ISMAR '05: Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 80–89, Washington, DC, USA, (2005). IEEE.

[10] K. Kallio. Mixed reality pong, 2001. Web page, accessed Feb 2009: `http://www.mlab.uiah.fi/~kkallio/mr-pong/`.

[11] Microsoft Corporation. Microsoft surface, 2009. Web page, accessed Feb 2009: `http://www.microsoft.com/surface/`.

[12] Maria Nika, *Virus Life: An infection spreading game with location tracking*, Master's thesis, Department of Computing, Imperial College London, 2008.

[13] Luc De Raedt and Luc Dehaspe, 'Clausal discovery', *Machine Learning*, **26**(2–3), 99–146, (1997).

[14] Paulo Santos, Simon Colton, and Derek R. Magee, 'Predictive and descriptive approaches to learning game rules from vision data', in *Advances in Artificial Intelligence: IBERAMIA-SBIA 2006, 2nd Int. Joint Conf., 10th Ibero-American Conf. on AI, 18th Brazilian AI Symposium, Brazil*, eds., J.Simão Sichman, H. Coelho, and S.O. Rezende, volume 4140 of *LNCS*, pp. 349–359. Springer, (2006).

[15] Bruce H. Thomas, Ben Close, John Donoghue, John Squires, Phillip De Bondi, and Wayne Piekarski, 'First person indoor/outdoor augmented reality application: ARQuake', *Personal & Ubiquitous Computing*, **6**(2), 75–86, (2002).

[16] Ubisense. Ubisense system overview. Available from Ubisense website `http://www.ubisense.net/` (accessed Feb 2009).

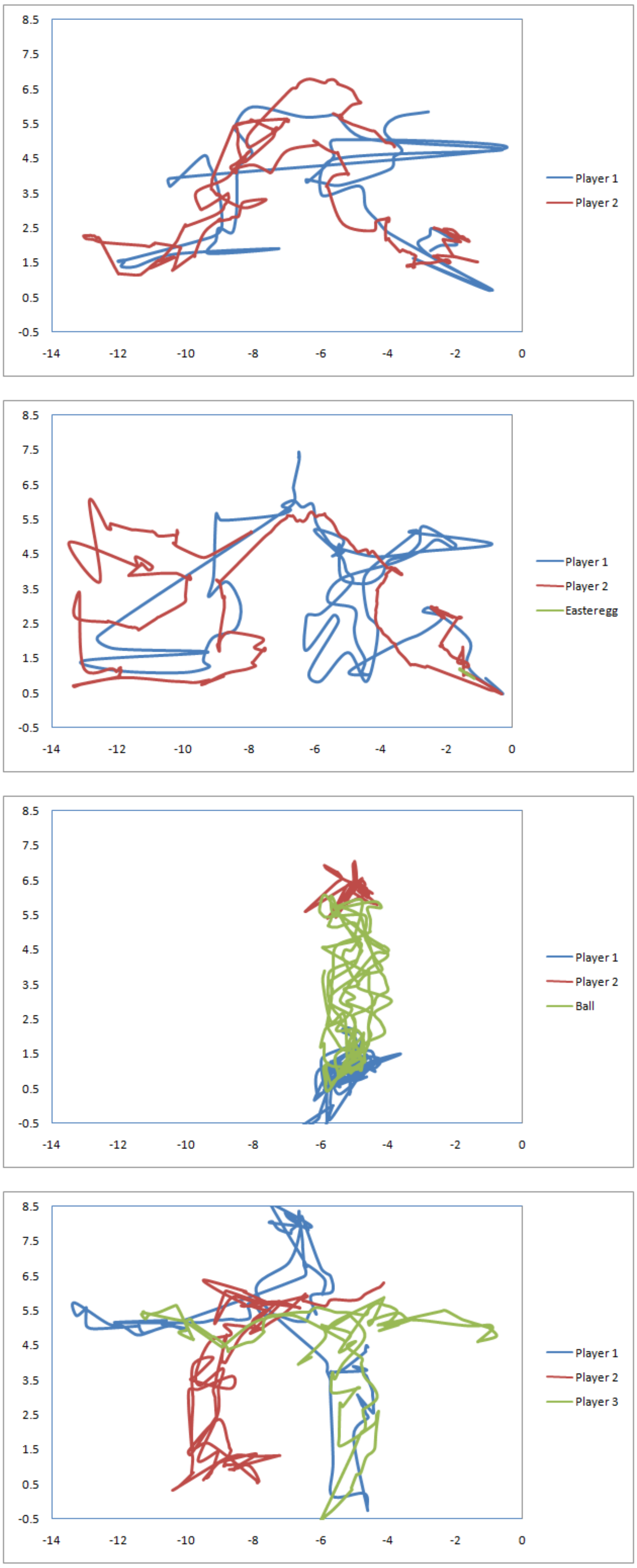


Figure 2. Tracking data showing player paths during a round of: (a) Tag (b) Easter Egg Hunt [with the egg located in the bottom right hand corner] (c) Human Pong, and (d) Wall.

Actual and Imagined Movement in BCI Gaming

Bram van de Laar and **Danny Oude Bos** and **Boris Reuderink** and **Dirk Heylen** [1]

Abstract. Most research on Brain-Computer Interfaces (BCI) focuses on developing ways of expression for disabled people who are not able to communicate through other means. Recently it has been shown that BCI can also be used in games to give users a richer experience and new ways to interact with a computer or game console. This paper describes research conducted to find out what the differences are between using actual and imagined movement as modalities in a BCI game. Results show that there are significant differences in user experience and that actual movement is a more robust way of communicating through a BCI.

1 INTRODUCTION

In the field of BCI, brain activity is recorded and automatically interpreted to be applied in various applications. Measuring brain activity is already well known in medicine using the electroencephalogram (EEG). EEG is a proven method, which has a number of advantages over other methods: it is non-invasive, has a high temporal resolution, does not require a laboratory setting, is relatively cheap, and it is even possible to create wireless EEG head-sets.

BCI systems need to make decisions based on very short segments of EEG data to make it useful for different applications such as wheelchairs, robots, and personal computers. In the case of software applications, BCI can be used as an additional modality of control, for evaluation of the user or the application, or to build adaptive user interfaces [16].

Games are usually the first applications to adopt new paradigms, driven by the gamers continuing search for novelty and challenges [17]. Apart from them being a suitable platform to bring this new interaction modality to the general population, games also provide a safe and motivational environment for patients during training or rehabilitation [5, 14]. Research has shown that using BCI instead of the conventional mouse and keyboard can add to the user experience by making a game more challenging, richer, and more immersive [1].

Before BCI can be adopted by the general population there are still a number of issues that need to be addressed: artifacts in the recorded brain data (signals that do not stem from the brain), inter and intra-subject variability, inter and intra-session variability, long training periods, low transfer rates (of commands), and BCI illiteracy [20]. Apart from that, more attention from the Human Computer Interaction community is required on how this new input modality influences the user experience, and how the interaction can be improved [12].

While most research into using movement for BCI has focused on imagined movement, some clinical research shows that actual movement in fact elicitates a more pronounced and therefore better discernable signal in the motor cortex [15].

Actual movement can also be used with other interfaces than a BCI. Interfaces such as a motion tracking system, for example, which is probably more reliable at this moment. One big potential advantage of a BCI however is that the measured EEG signals at the brains are always preceding actual muscle activity at the limbs. This advantage is amplified by the onset of a potential in preparation of a movement, the so called Bereitschaftspotential (Readiness Potential) [10]. Krauledat et al. showed that this lateralized readiness potential can be used to classify actual movement even before the movement itself is carried out [11]. This could give a gamer an advantage over other interfaces especially in fast paced, highly reactive games.

2 RELATED WORK

A few BCI games based on imagined or actual movement do already exist. Pineda et al. designed a first-person shooter game in which the user could move using the keyboard, and turn by imagined movement [19]. Players learned to control the BCI by experimenting; no instructions were given beforehand. Other examples include the virtual environments of Leeb et al. [13], the board game of Kayagil et al. [8], and the game BrainBasher [1] that we used in this study.

Both actual and imagined movement can be used for BCIs. Obviously, actual movement is a more natural and intuitive way for users to communicate and express themselves. All these games which involve movement tasks are based on a neurological process known as Event-Related Desynchronization (ERD) [18]. ERD is detectable as a decrease in power in the β-frequency band on corresponding motor cortices. Before use the BCI has to be adapted to person-specific examples of the ERD using machine learning techniques.

Actual movement is characterized by a more pronounced and reliable signal in the motor cortex [15]. This more pronounced signal is a very welcome advantage in the world of BCI where every extra percent of accuracy is appreciated.

When looking at the success of the Nintendo Wii, it becomes clear that actual movement is well enjoyed by gamers [2]. Moreover, imagined movement in adulthood is not as trivial as actual movement is. Although for example professional sportsmen and musicians use imaginary movement for training an actual motor skill it still is not as trivial to do as actual movement [7]. Though one can think of various applications in which imagined movement is used, these are almost always associated with skills which require a lot of training. Actual movement might therefore be a more natural and easier way of interacting with a BCI.

[1] Human Media Interaction, Faculty EEMCS, University of Twente, P.O. Box 217, 7500 AE, Enschede, The Netherlands, email: laar@ewi.utwente.nl, d.oudebos@ewi.utwente.nl, reuderin@ewi.utwente.nl, heylen@ewi.utwente.nl

[2] "Nintendo wiining the console war", December 2008. http://www.igizmo.co.uk/articles/news/744-gaming-nintendo-wiining-console-war

3 METHODS

The main question in this study is whether there are differences between imagined and actual movement in a BCI gaming environment. Some of the differences that will be looked into are the gaming experience for the user and the detectability of the signal from the EEG. We also looked at the generalizability of these BCI modalities for different user groups based on demographical characteristics.

3.1 Experiment Setup

To answer these questions an experiment has been carried out in which users communicate with the BCI game BrainBasher [1] using both kinds of movement. First, users fill in a form regarding demographics including handedness as well as characteristics that could influence their ability to focus on the task (like alcohol and caffeine consumption habits). This data is used to check for group differences during analysis. Our experiment consists of two parts: Actual movement and imagined movement. The order of performing actual and imagined movement is randomly assigned for each subject. Each part consists of two sessions.

For the system to be able to recognize the user's actions, a training session is required to create a user-specific classifier. For this classifier a cross-validated error rate is calculated which gives an indication of how well the system is able to detect the actions from a specific user. This is followed by a game session, after which the subject fills in a user experience questionnaire. This questionnaire has been designed based on the Game Experience Questionnaire (GEQ) developed at the Game Experience Lab in Eindhoven [6]. With this information the user experience for actual and imagined movement can be compared. Between all sessions are breaks in which the user can relax for a minute or two.

The setup is situated in a normal office environment, in contrast to a shielded room. This setting was chosen deliberately as it is a more representative setting for home use. Besides this, the EEG system used has active electrodes which pick up a lot less noise than passive electrodes would. During the experiments themselves, only the researcher and the subject will be in the room. This way distractions will be kept to a minimum, while still being able to provide help when needed.

The experiment is set up as a randomized cross-over experiment to eliminate sequence and learning effects induced by the succession of both tasks. After all experiments are done we compare the results of the actual movement sessions versus the results of the imagined movement game sessions. The new questionnaire has also been evaluated so it can be used for assessments of other BCI games and modalities.

3.2 BrainBasher

The BCI game used for this research is *BrainBasher*[1]. The goal of this game is to perform specific brain actions as quickly as possible. For each correct and detected action you score a point. Game control is achieved by two mental tasks: left hand movement versus right hand movement. For the actual movement task both hands are laid on the desk in front of the user. When the appropriate stimulus appears they have to perform a simple tapping movement with their whole hand. When performing the imagined movement task users are instructed to imagine the same movement, without actually using any of their hand muscles.

Before the user can play however, they will have to undergo a training session in which stimuli (in the form of symbols denoting

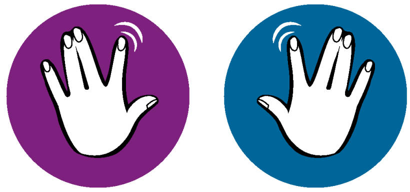


Figure 1. The symbols for left and right hand movement.

the user actions, see Figure 1) and breaks are alternated. During the stimulus the subject performs the indicated action: movement of the left or right hand. The user is instructed to stay relaxed and not to move, excluding the break periods, to prevent artifacts in the EEG. This is of course with the exception of the hand movement in the case of the actual movement sessions. In our system, the training consists of two short sessions, taking ten minutes in total. The EEG data from both training sessions are concatenated and used for training the classifier of the BCI system.

During the game session the user is instructed to take care that they carry out precisely the same movement (be it actual or imagined) as in the preceding training session. The difference is that they have to react as fast as they can to each new symbol popping up by performing the action right away. Bashing a symbol is accomplished when the classifier recognizes the action, according to a confidence level of at least 60%. Every *bash* results in one point added to their total score. The goal of the game is to bash as many symbols in the allocated three minutes, to achieve a maximum score. (Figure 2)

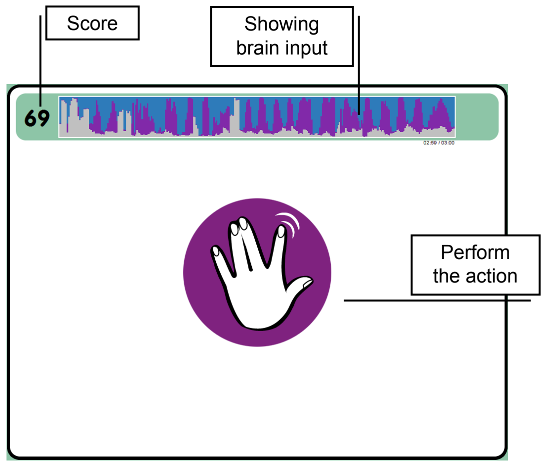


Figure 2. A game session.

A schematic view of the total system is shown in Figure 3. The user interacts with the system by executing brain actions, and also by keyboard to traverse the menu. Brain activity is acquired with a BioSemi EEG setup using 32 electrodes, sampled at 256Hz. For training the system, examples of the ERD for both the left hand and right hand are used to derive a linear classifier to be used during the online game session. The EEG data is processed as follows. First the raw data is re-referenced to the common average reference (CAR) to remove external sources of noise. After re-referencing a bandpassfilter isolates the frequency range in which the ERD occurs. Then we train spatial filters with the Common Spatial Patterns (CSP) algorithm [9] to extract activity on the motor cortices. These spatial filters are used to extract the band power in the most discriminative sources. Linear Discriminant Analysis (LDA) is applied to make a final prediction based on the band power features. After training the BCI generates four new predictions every second, based on the real-time EEG data. These predictions are used to play the game.

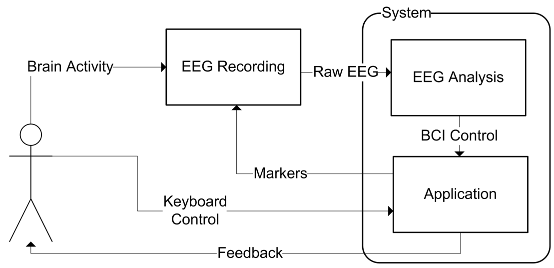


Figure 3. BrainBasher System View

3.3 Questionnaire design

To evaluate the user experience a questionnaire based on the GEQ [6] is developed. Although the GEQ consists of a lot of useful questions for evaluating various games, its main purpose is evaluating complex and immersive 3D virtual games. Therefore the questionnaire has been adapted to evaluate the user experience especially in BCI games. Questions that are not applicable, (e.g. the questions about the storyline, the complexity and the flow of the game) were left out. On the other hand we added questions, specifically on the amount of control the user experiences. The amount of control is a trivial aspect when using mouse and keyboard. These are reliable ways of communicating with the computer compared to a BCI system which is not so reliable. Also items about user concentration and alertness were added. This is an important aspect because users will have to concentrate to use a BCI game and possibly get tired more quickly than normal.

The questionnaire consists of statements to which users can respond on a 5 point Likert-scale ranging from 'completely disagree' to 'completely agree'. Some examples of items in the questionnaire might be: "I liked playing the game.", "I felt the computer recognized my actions." or "I'm exhausted."

To analyse the results of the questionnaire, we will use Cronbach's Alpha [4]. Alpha is a measure of internal consistency. It is (to a certain extent) a measure of how reliably a scale constructed out of the selected items will measure one concept. This does not necessarily mean that you are measuring the concept you intended to measure, therefore further (qualitative) validation is needed. Alpha is only an estimator of reliability: it measures to what extent the different items are correlating and are consistent, taking subject and environment variance into account. In this research the Standardized Alpha will be used because we want to sum standard scores to construct scales from Likert scale items. A commonly accepted threshold for Alpha of 0.7 is the goal for every scale. [3]

4 RESULTS

First we describe the demographics of the test subject pool, then we analyse the questionnaire used for evaluation. Using the results from the questionnaire we can look at the differences in user experience between actual and imagined movement.

Participants Twenty healthy persons participated as test subjects in this study. The average age across the group was 26.8 (standard deviation: 12.3, minimum: 13 and maximum: 58). Of the twenty participants 10 (50%) were male and 10 (50%) were female. Test subjects were randomly assigned to either group A or group B. Group A would do imagined movement as their first task and actual movement second, group B would do exactly the opposite. Each group had ten (50%) participants. 19 (95%) participants were Dutch, 1 (5%) participant German. Apart from standard demographics we also asked participants their handedness, because this characteristic might be of influence: 5 (25%) participants were dominantly left-handed 15 (75%) were right-handed. 14 (70%) received an education higher than average. Computer usage and game experience varied a lot among participants: 8 (40%) participants used a PC for more than six hours a day, 5 (25%) used a computer on a less than daily basis. The same variance goes for game experience: 2 (10%) played games two hours a day, 8 (40%) on a weekly basis, 6 (30%) on a monthly basis and 4 (20%) never played a video game.

Questionnaire construction All participants filled in the questionnaire after both tasks without missing any questions. The responses on the same items for both movement tasks were stored in the same respective variables for scale analyses and in separate variables to analyse the differences in user experience between both tasks. Scale reliability analysis was carried out in order to evaluate if the newly developed questionnaire would be useful as a reliable tool to assess user experience in BCI games. The total user experience questionnaire consisted of 42 items over 8 scales. Each item consists of responses to a statement on the user's experience on a 5 point Likert-scale.

Some items were recoded to avoid an expected negative correlation. Correcting the scales for items that did not constitute to the scales consistency, e.g. deleting items with a low or negative Inter-Item Correlation, Standardised Alpha's ranged from 0.620 tot 0.865 and all scales consisted of at least three items.

To evaluate the usefulness and dimensionality of the resulting scales, a factor analysis was done on all scales separately. The first dimensions in the factor analyses of every scale explained more than 56% of the variance in the data, except for the Negative Experiences scale. Scree plots [2] also indicated strong unidimensionality across all scales except for the Negative Experiences scale, which turned out to be a two dimensional scale. The corrected questionnaire consisted of 32 items divided over 8 scales. An overview of all corrected scales with their respective Alpha's and variance explained by the first dimension in factor analysis can be found in Table 1. The variance explained by the first factor measures to what extent a scale is measuring only one underlying attribute or construct.

Differences in user experience The final corrected scales were used to compare the user experience for users performing both kinds of movements. A direct comparison by means of paired *t*-tests was done. The results of these test can be seen in Table 2. The first column is the difference of the means of both scales, the second column

Construction of Scales			
	No. of items	Alpha	Var. explained
Alertness	3	0.783	70.4%
Challenge	5	0.777	56.4%
Control	3	0.783	69.9%
Goals	3	0.754	67.7%
Fatigue	3	0.759	67.6%
Immersion	3	0.620	57.0%
Negative Experiences	5	0.638	41.9%
Positive Experiences	7	0.865	55.8%

Table 1. Constructed Scales including alpha and variance explained by 1th principal component

is the total standard deviation, the third the *t*-score and the last column is the two-tailed significance of the difference. The data show that the differences in the user experience for the Alertness as well as the Challenge scales are significant. Actual movement scored significantly higher on the Alertness scale ($t(19)$=-2.42, p=0.03) which could be attributed to mental tiring process of performing imagined movement. The same trend is also shown in the Fatigue scale, while there is no significant difference between actual and imagined movement (p=0.12). One possible explanation for this can be found in the correlation between the Fatigue and Alertness scale. These show a strong negative correlation in actual movement (r=-0.707, $p<0.001$). Challenge also significantly differs between both kinds of movement ($t(19)$=2.17, p=0.04). User experience data therefore indicates that performing imagined movement is more of a challenge than actual movement is.

Differences of Imagined vs. Actual Movement				
	Diff of avg	StDev	t	Sig (2-tail)
Alertness	-.65	1.20	-2.42	.03
Challenge	.40	.83	2.17	.04
Control	-.30	1.34	-1.00	.33
Goals	-.18	.50	-1.63	.12
Fatigue	.40	1.11	1.62	.12
Immersion	-.15	.60	-1.12	.28
Negative Experiences	.00	.59	.00	1.00
Positive Experiences	-.24	.89	-1.22	.24

Table 2. Paired *t*-Tests Scales, comparing imagined and actual movement

Performance Using the error rate calculated by the classifier from the EEG data we can compare the performance achieved on different subjects. For each subject two error rates are available, one for actual and one for imagined movement. The average rate for actual movement is 38.67%, while the average error rate for imagined movement is 42.28%. A Wilcoxon signed-rank test showed that actual movement error rates are significantly lower ($W_{+}(20)$ = 48, p = 0.0328). Looking at performance across different groups there are no significant differences between men and women in actual ($t(19)$=0.584, p=0.570) or imagined ($t(19)$=0.205, p=0.840) movement. Comparing left handed with right handed test subjects also didn't show any significant differences in actual or imagined movement ($t(19)$=-0.876, p=0.403 and $t(19)$=0.99, p=0.923 respectively).

5 CONCLUSIONS AND DISCUSSION

Results from this study showed that differences in user experience and in performance between actual and imagined movement in BCI gaming do exist. Actual movement produces a more reliable signal while the user stays more alert. On the other hand, imagined movement is more challenging.

To be able to assess the differences in user experience between actual and imagined movement, we developed a questionnaire for evaluating BCI games. While this questionnaire was found to be a numerically grounded tool to be used in this setting, further research for validation is needed.

User experience data from this questionnaire showed two significant differences. Users found more challenge in performing imagined movement. This might be due to a higher error rate, which makes sense; looking at the average error rate, it is harder to perform imagined movement. If we assume imagined movement is a skill that can be learned this might be an advantage for using imagined movement. Gamers are always looking for challenges and limitations that they can overcome by practice [17].

On the other hand, for a few test subjects, the BCI system could not correctly recognize any movement. This corresponds to an error rate of 50%, in which case simple random 'guessing' would be as good as classification. Participants who achieved a high error rate also were not able to score any game points (other than maybe by chance). This is an issue that frustrates the user and is something that has to be resolved for wider acceptance of BCI gaming. This problem of not being able to be *understood* by a BCI is referred to as BCI illiteracy [20].

Alertness is the other scale in which a difference was found. This alertness has to do with the state of mind of the user after they played the game. The fact that they felt less alert after performing imagined movement is explainable. Imaginary movement requires more concentration and is a less natural action to perform. Doing something you do everyday does not tire you as much as doing something completely new. This was also reflected in the Fatigue scale, which scored slightly higher for imagined movement.

The generalizability over various demographic groups was good and there were no significant differences in performance. While there have been some anecdotal findings that women would be better in communicating through a BCI, results show no significant differences between men and women. Data also did not show any differences between left and right handed people. While the gathered data does not provide a clear view on how age is related to performance in the game, one might hypothesize that imagined movement is a skill of young children who mimic movements of others. A child sees someone performing a certain movement that can be of advantage to the child, for example grabbing something, they then try to perform it themselves. This probably is a skill that fades over time when a person gets older. While at a higher age humans are still able to mimic movements, it takes more time to learn them. This is possibly a ground for older people not performing to well at imagined movement. This was also reported by test subjects to the experimenter. They don't know how or what they should imagine.

Future work could include research into the different ways of imagining movement. As McFarland et al. [15] already explain: when given the instruction to imagine a movement, most people will try to sense the movement. Other kinds of imagination (e.g. visualizing the movement) might activate different cortical areas. Some users might even prefer to visualize a movement if they find it more natural or less tiring. Evaluating the performance and user experience of these different tasks are a valuable addition.

The developed questionnaire seems to be a instrument that can aid us in evaluating differences in user experience between different modalities for BCI, but it might also be of interest for evaluation of BCI games other than BrainBasher. Then further research on the validity and generalizability of the questionnaire is needed.

Although the game works in an online manner and the classification algorithm is fast enough to be computed realtime, there always is an inherent delay in feedback. This is due to the fact that the classification algorithm needs a measurement of EEG data of a few seconds. Currently this measurement is two seconds. The consequence is that users get feedback of what they did with a two second delay. This delay sometimes leads to confusion and a lower positive affect towards the system. Future research might include shortening the response time of the underlying system of the game and finding out what this does for acceptance of and positive affection towards the BCI.

Because of the similarities in brain activity between actual and imagined movement and the somewhat lacking of intuitivity for imagined movement one might suggest using actual movement as a training for using imagined movement. The user of the BCI can get accustomed to using movements for communications and at the same time trying to imagine the movement. With the acquired data from the actual movement, the imagined movement could be classified.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the BrainGain Smart Mix Programme of the Dutch Ministry of Economic Affairs and the Dutch Ministry of Education, Culture, and Science. We would also like to thank all the people who were willing to submit precious time to being a test subject for our experiments.

REFERENCES

[1] D. Oude Bos and B. Reuderink, 'Brainbasher: a BCI game', in *Extended Abstracts of the International Conference on Fun and Games 2008, Eindhoven, Netherlands*, eds., P. Markopoulos, J. Hoonhout, I. Soute, and J. Read, pp. 36–39, Eindhoven, (October 2008). Eindhoven University of Technology.

[2] R. B. Catell and S. Vogelmann, 'A comprehensive trial of the scree and Kg criteria for determining the number of factors', *Multivariate Behavioral Research*, **12**(3), 289–325, (1977).

[3] J.M. Cortina, 'What is coefficient Alpha? an examination of theory and applications', *Journal of Applied Psychology*, **78**(1), 98–104, (1993).

[4] L.J. Cronbach, 'Coefficient alpha and the internal structure of tests', *Psychometrika*, **16**(3), 297–334, (1951).

[5] B. Graimann, B. Allison, and A. Gräser, 'New Applications for Non-invasive Brain-Computer Interfaces and the Need for Engaging Training Environments', *BRAINPLAY 07 BCI and Games Workshop at ACE*, (2007).

[6] W. IJsselsteijn, Y. de Kort, K. Poels, A. Jurgelionis, and F. Bellotti, 'Characterising and Measuring User Experiences in Digital Games', *International Conference on Advances in Computer Entertainment Technology*, (2007).

[7] M. Jeannerod, 'The representing brain neural correlates of motor intention and imagery', *Behavioural Brain Sciences*, **17**, 187–245, (1994).

[8] T. Kayagil, O. Bai, P. Lin, S. Furlani, S. Vorbach, and M. Hallett, 'Binary EEG Control for Two-Dimensional Cursor Movement: An Online Approach', *IEEE/ICME International Conference on Complex Medical Engineering*, 1542–1545, (May 2007).

[9] Z. Koles, 'The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG', *Electroencephalography and Clinical Neurophysiology*, **79**(6), 440–447, (December 1991).

[10] H. Kornhuber and L. Deecke, 'Hirnpotentialänderungen bei willkürbewegungen und passiven bewegungen des menschen: Bereitschaftspotential und reafferente potentiale', *Pflügers Archiv European Journal of Physiology*, **284**(1), 1–17, (1965).

[11] M. Krauledat, G. Dornhege, B. Blankertz, F. Losch, G. Curio, and K.-R. Müller, 'Improving speed and accuracy of brain-computer interfaces using readiness potential features', *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, **26**, 4511–4515, (2004).

[12] A. Lecuyer, F. Lotte, R.B. Reilly, R. Leeb, M. Hirose, and M. Slater, 'Brain-computer interfaces, virtual reality, and videogames', *Computer*, **41**(10), 66–72, (October 2008).

[13] R. Leeb, C. Keinrath, D. Friedman, C. Guger, C. Neuper, M. Garau, A. Antley, A. Steed, M. Slater, and G. Pfurtscheller, 'Walking from thoughts: Not the muscles are crucial but the brain waves!', in *Proceedings of the 8th Annual International Workshop on Presence*, pp. 25–32, (2005).

[14] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, and G. Pfurtscheller, 'Brain-Computer Communication: Motivation, Aim, and Impact of Exploring a Virtual Apartment', *IEEE Transactions on Neur Sys. and Rehab. Eng.,*, **15**(4), 473–482, (2007).

[15] D.J. McFarland, L.A. Miner, T.M. Vaughan, and J.R. Wolpaw, 'Mu and Beta Rhythm Topographies During Motor Imagery and Actual Movements', *Brain Topography*, **12**(3), 177–186, (2000).

[16] A. Nijholt, D. Tan, B. Allison, J. Milan, and B. Graimann, 'Brain-computer interfaces for HCI and games', in *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pp. 3925–3928, New York, NY, USA, (2008). ACM.

[17] A. Nijholt, D. Tan, G. Pfurtscheller, C. Brunner, J.R. Millán, B. Allison, B. Graimann, F. Popescu, B. Blankertz, and K.R. Müller, 'Brain-Computer Interfacing for Intelligent Systems', *IEEE Intelligent Systems*, **23**(3), 72–79, (2008).

[18] G. Pfurtscheller, 'Functional brain imaging based on ERD/ERS', *Vision Research*, **41**(10-11), 1257 – 1260, (2001).

[19] J. Pineda, D. Silverman, A. Vankov, and J. Hestenes, 'Learning to control brain rhythms: making a brain-computer interface possible', *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **11**(2), 181–184, (june 2003).

[20] C. Sanelli, M. Braun, M. Tangermann, and K. Mller, 'Estimating noise and dimensionality in BCI data sets: Towards BCI illiteracy comprehension', *Proceedings of the 4th International Brain-Computer Interface Workshop and Training Course 2008*, (2008).

MetroPed: A Tool for Supporting Crowds of Pedestrian AI's in Urban Environments

Christopher Peters [1,2†] and Carol O' Sullivan [1]

Abstract.
MetroPed is a tool for supporting AI behaviours in real-time virtual city environments. Given information about the basic geometry of the environment being modelled, it allows a designer to annotate and mark-up elements of relevance to AI characters, providing support for appropriate autonomous AI behaviour in a manner consistent with the visual representation of the environment. We describe the tool and mark-up process, providing insight into the importance of AI tools and principled approaches when attempting to create large numbers of AI controlled characters for use in real-time applications situated in urban environments. We also describe how we have been using the tool in novel ways to research viewer perception of crowd behaviour.

1 INTRODUCTION

The availability of quality tools is increasingly being recognised as an important part of the repertoire of game development companies, and has even been suggested [9] as a significant factor in securing publishing deals for a number of AAA titles, for example Crytek's *Far Cry* [3].

Here, we describe *MetroPed*, a proprietary tool that is being used to support the creation of an inhabited real-time virtual city as part of our project, called *Metropolis*. MetroPed provides the designer with a specialised interface affording real-time feedback. The primary purpose is to ease the definition of schematic AI information for supporting behavioural and cognitive animation of pedestrian crowds. A parallel representation of the environment is constructed in addition to that used for rendering, with the specific purpose of informing the AI decision making processes. Importantly, the tool also supports the definition of crowd scenarios for viewer perception studies. While similar tools appear to be widely used in the games industry for supporting character behaviour (Autodesk's *Kynapse* for example, see also [6] and [5]), they have not generally been used to support experimentation, as is the case here.

There are a number of reasons as to why we have chosen to create a specialised, proprietary AI tool over using more generic tools not specific to the task, such as text-based interfaces or 3D application plug-ins.

1. The designer has a specialised GUI to work with, enabling visualisation of concepts that may be tedious to modify using a text-based interface. For example, character placement is not usually a difficult task. However, it becomes extremely time consuming if one must position large quantities of characters in the absence of specialised, high-level, intuitive controls.
2. Unlike the approach of extending a third-party graphics package, for example, creating a plug-in for Autodesk's *3D Studio Max*, MetroPed is specialised for the task: It constrains the activity space so that a novice designer need learn only those aspects and features of relevance to AI. Since the program is self-contained, developers are not dependent on changes made to third-party graphics programs as they are updated, and have full control over the tool.
3. When marking-up the environment, the designer is presented with a level of abstraction from the underlying engine. This means that he/she does not have to be expert in engine programming in order to be able to define scenarios and add behaviours.
4. The tool can ensure that the designer is constrained to entering valid, consistent information that will not crash the engine, and can provide additional help on engine-specific variables and parameters.

This paper provides an overview of the tool and process that is used to define and export a simple scenario in Section 3, and then focuses in more detail on two distinct roles: embedding semantic behaviour-related information into the scene (see Section 4) and specifying agent scenario information (see Section 5). In Section 6 we provide an important example of how the tool is being used, not only to simulate pedestrian behaviour, but also to help conduct perception studies for establishing factors of importance in natural looking crowd behaviour. First, we consider related concepts and work.

2 BACKGROUND

A number of methods exist allowing characters to interact with their virtual environment. A key factor of differentiation between these methods concerns where knowledge is stored in the system. One approach is to endow knowledge separately to individual characters, an extreme example of which would create autonomous agents that have their own artificial perceptions, reasoning, memories, etc with respect to the environment. For many reasons this is a technically challenging approach.

Another method is to place knowledge into the environment itself, to create a shared or partially-shared database accessible to characters. For example, rather than attempt to give the agent the ability to identify a car, we could provide a label in the car object identifying it as such. This approach can be taken further, so that objects provide semantic information about themselves e.g. a teapot identifies a grasp point, or even gain control of agents using them, e.g. an elevator object may position an agent in front of the door so that it can enter the elevator. According to this *smart object* methodology

[1] Graphics, Vision and Visualisation Group, Trinity College Dublin, Ireland
[2] Department of Computing and the Digital Environment, Coventry University, United Kingdom.
[2] † Research conducted while at Trinity College Dublin

[11], graphical objects are tagged with behavioural information and may inform, guide or even control characters. Such an approach is applicable also to crowd simulation in urban environments. For example, navigation aids, placed inside the environment description, may be added by the designer during the construction process. These have been referred to as *annotations* [1]. The resulting environment description contains not only rendering data, but also geometric, semantic and spatial partitioning information for informing pedestrian behaviour [4],[14], thus transferring a degree of the behavioural intelligence into the environment. In [7], for example, skeletal splines are defined that are aligned with walkways. These splines, called *ribbons*, provide explicit information for groups to use, such as the two major directions of travel on the walkway.

In addition to environment annotation and mark-up, interfaces for managing the definition of crowd scenarios have also been investigated. *Crowdbrush* [15] provides an intuitive way for designers to add crowds of characters into an environment using tools analogous to those found in standard 2D painting packages. It allows designers to paint crowds and apply attributes and characteristics using a range of different tools in real-time, obtaining immediate feedback about the results.

3 OVERVIEW

The core task of MetroPed is to define the AI environment in much the same way that a commercial graphics package would be used to define the rendering environment. As such, although MetroPed has basic rendering and simulation capabilities, these are the core responsibility of the destination engine. Although MetroPed is being used with the Metropolis visualisation engine, it could also be used to define environments for other engines.

3.1 Outputs

The output of MetroPed consists of a series of XML files containing the final annotations and scenarios that have been defined by the designer. These files are loaded into a separate rendering engine in addition to the scene meshes, textures and other display data. The main engine uses the data in the files to position pedestrians and direct their behaviours in real-time.

3.2 Modes of Operation

There are three primary modes of operation: *Create*, *Simulate* and *Experiment*.

1. Create: This mode allows the designer to mark-up a basic mesh by defining zones, paths, goals and other points of interest (described in more detail in Section 4.1). It also allows the designer to place pedestrians in the scene and specify starting scenarios, as described in Section 5.
2. Simulate: This mode allows the designer to run the pedestrian simulation and visualise the results. This allows the designer to ensure that any new alterations to the mark-up are functioning as expected with the pedestrian simulation algorithms. It also allows for prototyping of the new simulation algorithms.
3. Experiment: This mode allows a grid to be defined, into which agent formations may be placed, either as predefined in a 3D Studio Max file or according to automatic rules. This mode has been used to create experimental scenes with which to support our research on viewer perception of crowd formations [12][2] and is described in more detail in Section 6.

In Sections 4 and 5, we focus on operations conducted during the Create and Simulate modes of operation: that is, defining zones, nodes, paths and other features on the mesh, and specifying the placement of pedestrians within the environment.

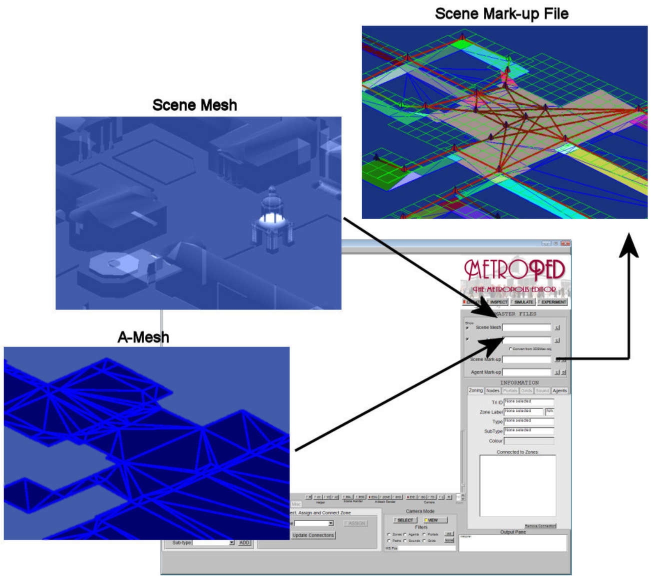


Figure 1. The primary purpose of MetroPed is to annotate an environment with information to guide the AI for both simulation and user perception studies. This process involves loading in a blank Annotated Mesh, onto which annotations are then conducted. Optionally, the scene mesh can also be loaded in, to provide a visual reference for the designer. The output of the annotation process is stored in a *scene mark-up file*.

4 SCENE ANNOTATION

Scene annotation refers to the process of embedding information into a scene to support crowd behaviours. For example, rather than endow each pedestrian with some form of sophisticated mechanism to discern footpaths from roads, it is more practical to specify footpaths and roads in the environment and then allow the agent to 'sense' or otherwise use this information for decision-making. It is worth noting that in the final simulation, no aspects of the annotated scene file will ever be rendered or directly seen by the viewer. In contrast, agents will only process the annotated scene file - they have no knowledge of the display geometry.

The first step in our system involves the creation of an *annotation mesh* as described next. The mesh is loaded into MetroPed, optionally with the scene rendering representation, and is then annotated by the designer. The result is saved out as a *Scene Markup File*, as depicted in Figure 1.

4.1 The Annotation Mesh

The *Annotation Mesh*, or A-Mesh, is a piece of basic geometry representing the ground plane. The A-Mesh is a type of navigation mesh [13], representing walkable areas of the city environment (see Figure 2). It acts as a base, onto which the designer annotates and adds semantic information, as will be described in Section 4.2. In order for an A-Mesh to be processed properly within MetroPed and the final simulation engine, the mesh must be *well-formed*.

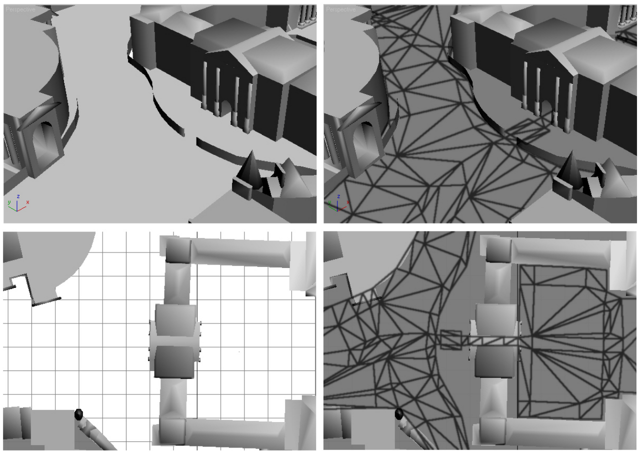


Figure 2. Depiction of the Annotation Mesh creation phase. In *3D Studio Max*, the city mesh is loaded in (left, top and bottom). Using this as a reference, the basic annotation mesh is created as a ground plane to represent the topology of streets and paths (right, top and bottom). It is later loaded into MetroPed, where the annotation process takes place.

4.2 Process

Once the A-Mesh has been created for a specific environment, the annotation process begins by loading it into MetroPed. Optionally, the scene mesh may also be loaded in: this mesh is a visual representation of the scene used solely for the purposes of display and rendering. MetroPed does not conduct any operations on this mesh, nor does the behaviour system interact with it. For an urban environment, this mesh will usually correspond to the buildings and other objects may ease the task for the user of visually placing behavioural objects in the environment.

In contrast, the A-Mesh represents a *blank behavioural slate* onto which behavioural information is added. It contains the basic geometry, such as vertices, edges and faces, that describe the ground features of the city - footpaths, roads, junctions, crossings and so on. It must be carefully created in a 3D graphics program before import into MetroPed. We used 3D Studio Max for this purpose. MetroPed allows the user to add behavioural data to this mesh and save it as a *Scene Mark-up file*.

4.3 Zones

In terms of the A-Mesh, a zone represents a collection of faces sharing some common spatial aspects (see Figure 3). Each zone is associated with it's own simple grid structure that is used for nearest neighbour calculations during simulation. We have enumerated two primary zone types of relevance to the A-Mesh, for supporting pedestrian behaviours in the city environment: pedestrian zones and road zones. Of the pedestrian zone types, there are four basic subtypes: *corridors*, *junctions*, *crossings* and *open* zones.

1. Corridor zones consist of uninterrupted stretches of pedestrian corridor. These zones are similar to passages where movement tends to be bidirectional.
2. Junction zones represent the intersection of two or more corridor zones.
3. Open zones represent areas of the city that do not fall into the corridor category and represent wider areas of space.
4. Crossing zones are similar to corridors, but span road zones.

As zones are being defined in MetroPed, connectivity information is automatically generated to track the connected zones and also provide feedback to the designer if invalid combinations of zones are being placed. A junction must be used to connect different zone types, for example, to connect two corridor zones directly together, or a crossing zone to a corridor zone.

Essentially, walking areas in the city environment are modelled as a connection of a set of corridor, junction, open and crossing zones. In practice, most of the city consists of interconnected corridor and junction zones, since these are more common than open zones and crossings. Although the categorisation is straight forward, it is a powerful way for partitioning the city and enables a simplification of navigation problems. Navigation is considered at two-levels. At a low-level, navigation is within-zone: that is, the consideration of how pedestrians can move about within a single zone (Section 4.4). At a higher level, zones are related to nodes in a navigation graph allowing for global pathfinding to take place across the whole city (Section 4.5).

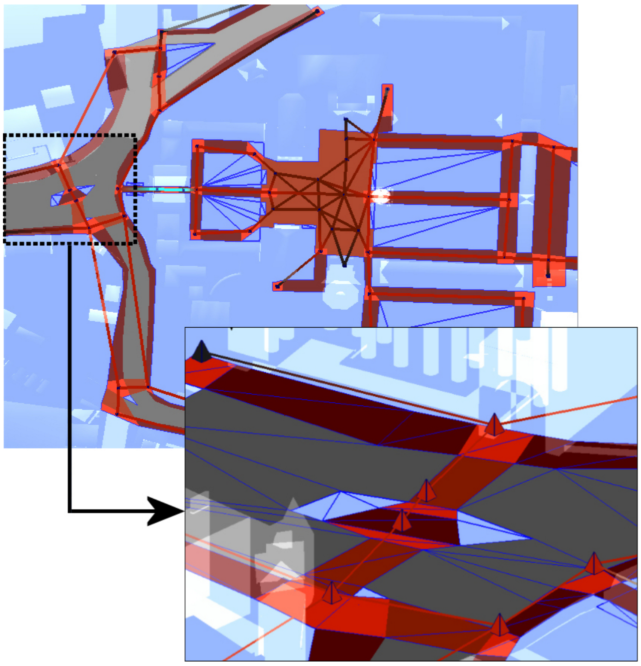


Figure 3. Depiction (top) of environment zoning and global paths, with nodes shown in red, with zoomed-in view shown (bottom). Zones depicted in red are pedestrian zones: corridors, junctions and crossings.

4.4 Local Paths

The environment that we model is based on a real European city and therefore the street layout is neither regular nor trivial. Because of this, a single pedestrian zone, such as a corridor, may actually correspond to an area of terrain over which it is difficult to navigate. Therefore, local paths are used to provide the necessary information in order to allow pedestrians to navigate within a zone, from one end to the opposite, without straying outside its borders.

When a zone has been defined and its neighbouring zones have been established, all edges in the corresponding A-Mesh geometry for the zone are classified and tagged with connectivity information. Zone edges are classified into three types (see Figure 4):

1. Border edges are defined as edges where the zone on the opposite side of the edge is not the same as the current zone. Border edges are important for ensuring *containment*, i.e. that pedestrians remain within the confines of the zone in which they are walking. Imposition of such a rule may be dependent on the type of zone on the other side of the border edge in question: for example, it would be more important to enforce the containment rule if there is no zone on the other side of the edge (specifying a non-walkable area) or a road zone, than if there is a walkable grass zone. These behaviours are programmed into the pedestrians, but require the edge information defined here in order to function.
2. Entry/exit edges are special border edges where the zone type on the opposite side of the edge is a pedestrian junction zone. The key purpose of local paths is to allow a pedestrian to navigate from the entry edge to the exit edge of the zone.
3. Spine edges are interior edges that run along the length of the zone, perpendicular to its main direction of travel. These edges are used during the automatic local path generation process to specify local path nodes.

A local path node is generated on the center position of each spine edge and entry/exit edge. These path nodes are connected together to form path connections. The result is a local path that can be used to guide pedestrians through the zone.

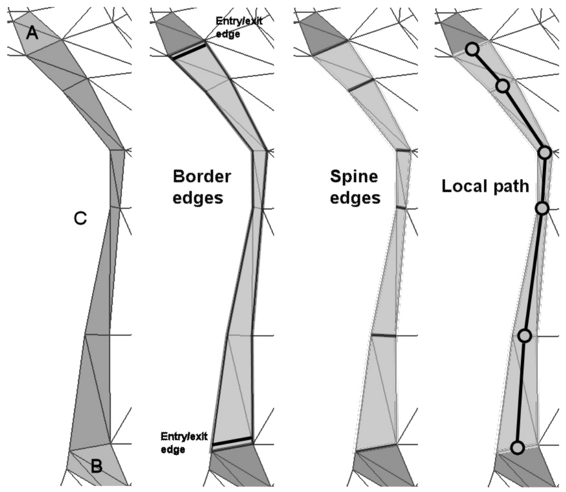


Figure 4. Local paths are specified for corridor zones. Here, A and B are pedestrian junction zones and C is a pedestrian corridor zone. The border edges and entry/exit edges (second from the left), spine edges (third from the left) and local path nodes and connections (rightmost) are shown for zone C.

4.5 Global Paths

Global paths are comprised of connected global nodes. There are three basic global node types in the current system, although new types can be added very easily as extra functionality is required.

1. Junction nodes correspond to junction zones and are the most basic type of global path node, representing the intersection of multiple connections.
2. Generator nodes describe areas where pedestrians can be added (or spawned) on the map or removed. For example, these nodes can be placed at building entrance points to simulate the movement of pedestrians into and out of buildings.
3. Tunnel nodes instruct the rendering engine not to display pedestrians that are walking on connections between two consecutive tunnel nodes. These can be used when pedestrians are walking through a tunnel and it is not necessary to display them to the viewer.

All of the global nodes are connected together into a global graph. This graph is queried by pedestrians to provide global navigation information based on A* search.

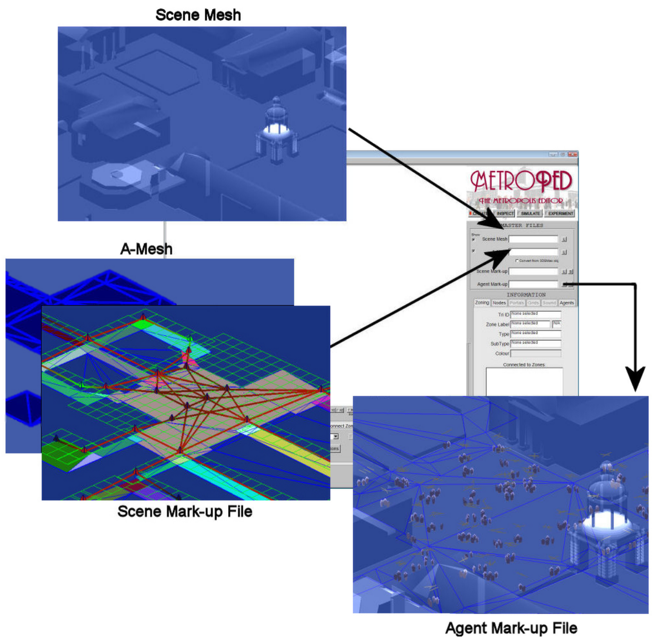


Figure 5. MetroPed can also be used to specify agent scenarios, defining initial placement parameters. Although this process does not strictly require an annotation mesh, in practice, agent placement should take place on the fully annotated A-Mesh in order to allow the system to enforce constraints, such as ensuring that pedestrians cannot be placed on road zones.

5 AGENT PLACEMENT AND SCENARIO SPECIFICATION

MetroPed may also be used the place agents and groups into the scene in order to specify crowd scenarios. The results can be saved to an *Agent Mark-up file* (see Figure 5).

5.1 Agent Behaviour

The are three primary operations involving agents. First of all, the selection of agent templates (Section 5.2), that define the basic properties and characteristics to be applied to the agents. Secondly, the placement (Section 5) of agents within the scene according to the environment placement constraints selected by the user. Finally, the simulation of the agents within the environment, according to both the environment annotation and the characteristics defined for the agents.

5.2 Agent Definition

A simulation containing agents that all looked and behaved in the manner would undoubtedly appear unrealistic. *Agent templates* allow agents to be attributed with a degree of individuality in terms of appearance and behaviour, based on a pool of predefined character types. An agent template consists of three key components:

1. Locomotion properties: these define the maximum and average walk speed of the agent.
2. Personality properties: these properties, based on the *OCEAN* model of personality [16], may be used to alter behaviour in a variety of ways according to whatever underlying pedestrian simulation model is being used. Following the OCEAN model, there are individual settings for *Openness*, *Conscientiousness*, *Extraversion*, *Agreeableness* and *Neuroticism*.
3. Character description: The character description links the behavioural aspects of the agent to purely graphical aspects by attaching some simple semantic data. A mesh is tagged with an accompanying description of the age, gender, physique, attire and a short textual description. The purpose is to ensure that the behavioural traits attributed to an agent are *appropriate* to the graphical appearance e.g. the maximum speed of an elderly-looking character should be less that of a character that looks young and fit. These details are also useful for scripting e.g. when created groups of characters automatically, specifying that groups containing two people consist of a female accompanied by male of appropriate age and matching attire.

It is important to note here that none of these components are necessarily attached to any particular simulation method - the same template properties may be used and interpreted differently by different underlying crowd simulation techniques, depending on the situation.

5.2.1 Agent Data Files

There are two input data files associated with defining agent properties in MetroPed. These are the *character description file* and the *agent templates file*. As previously mentioned (Section 5.2), the character descriptions file contains a list of filename references to a mesh with accompanying semantic categories.

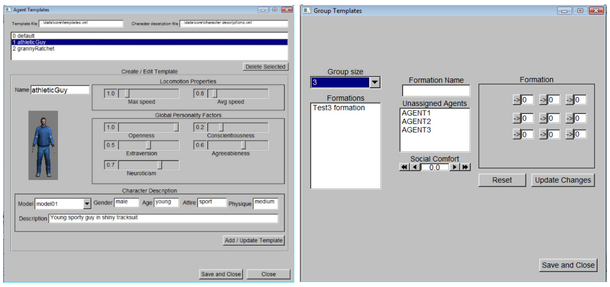


Figure 6. The agent template window allows character behaviour to be defined at a high level and to be linked with graphical appearance. Each agent in the system is associated with a single template.

5.2.2 Agent Templates

An agent template links locomotion and personality traits to a character description. Each agent in the system is attributed with a single agent template that defines not only high-level pedestrian behaviour parameters, but also the appearance of the character. The main reasoning behind agent templates is that the behaviour of the character should be consistent with its appearance. For example, a young character wearing a tracksuit should have a faster average walking speed than an older-looking character.

The agent templates window (Figure 6) is used to browse, edit and save agent templates. Each template consists of a unique template name, and locomotion and personality properties. A template is associated with a specific Character Description, which links the template to a graphical representation of the character and also contains a semantic description of the representation, such as the gender, age, attire and physique of the character depicted. These descriptors have been made as generic as possible in order to ensure independence from the underlying steering and decision-making system.

6 RESEARCH APPLICATIONS

MetroPed runs as a separate program from the Metropolis visualisation engine, the primary display engine, in order to reduce complexity and simplify the code-base with which the AI programmer must work. The resulting mark-up and scenarios are easily exported from the tool in a generic file format. This has allowed us to also use it, not only to aid in the definition and simulation of crowd behaviour, but also to support the creation of scenes for conducting user perception studies.

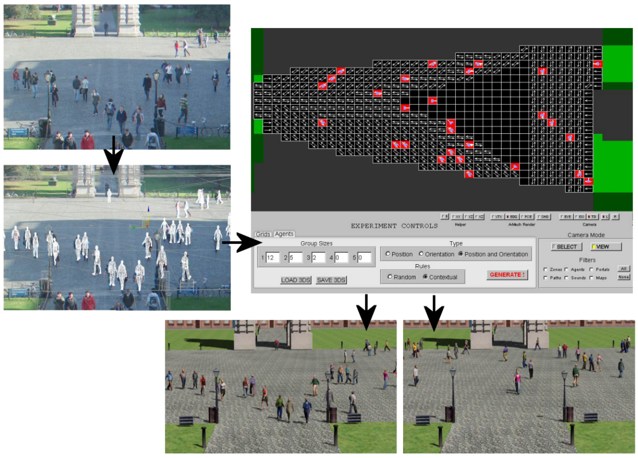


Figure 7. Process adopted for conducting viewer perception studies of crowds. (Left) Photographs of real scenes are annotated in order to create virtual replicas (bottom left). MetroPed (top right) alters details of the real scenes in order to produce altered versions (bottom right).

Our evaluation methodology consists of four phases (see Figure 7). During the *data collection phase*, a number of photographs are taken of one or more locations of interest. These photographs are manually annotated during the *annotation phase*, in order to highlight features of interest. For example, when investigating the formations and behaviours of groupings of two, three and four individuals, each group was designated by a colour-coded ellipse including all members of the group, highlighting their walking direction. This enables a quicker visual analysis to be conducted in the scene for recording general characteristics that are of importance later, during the *reconstruction phase*. In this phase, a virtual replica of the scene is constructed based on the annotated original. This is achieved by aligning a 3D model of the virtual scene with the real scene by manually tweaking the virtual camera parameters and then positioning virtual characters in the same locations as their real counterparts. This provides a relatively close approximation to the original scene composition. It should be noted that not all the final visual characteristics in the virtual scene will exactly match the original: clothes colour and other minor details may change. During the final *modification phase*, some of the virtual replicas are automatically altered. The altered crowd scenes are then presented, along with the replicas, to participants during evaluation studies to discern the impact of changes made to the scene.

MetroPed is central tool in this process: the ground area where the scene is located is loaded into the tool and the ground is segmented into zones and grids which are labelled with terrain and obstacle information. Once a virtual replica of a crowd scene has been created, containing the positions and orientations of each pedestrian for a single frame, it is also imported into MetroPed. Automatic algorithms may then be selected by the designer in order to alter aspects of the crowd based on the ground information previously added. For example, in [12], we enumerated a number of basic rules for modifying pedestrian orientation. The orientations of all individuals in the scene modified according to one of the following basic orientation rules: *Random* rule, where each individual is assigned an orientation chosen at random from one of eight cardinal directions. *Uniform* rule, where a single orientation is chosen at random from one of eight directions and all characters are aligned to match it. *Even* rule, where individuals are chosen at random and assigned an orientation from one of the eight cardinal directions so as to fit into an overall even distribution. We have conducted similar studies in relation to the positioning of individuals and groups. The use of MetroPed has enabled us to use many more scenes than would otherwise have been possible taking a purely manual approach. Nonetheless, further automation of time consuming tasks remains a key goal for future development, in order to reduce human burden when assembling experiments. Figure 8 provides a final rendering of an example scene imported into the Metropolis visualisation engine.

Figure 8. A screenshot of pedestrian behaviour in the Metropolis visualisation engine.

7 CONCLUSION

We have presented MetroPed, a tool for supporting pedestrian behaviour of AI characters in virtual urban environments. It allows a designer to mark-up the environment with information for use by the AI and supports the creation of crowd scenarios for visualisation in the main Metropolis engine. MetroPed stands out from other tools of its type as it has been designed to aid in the creation of scenarios for use in experiments: it has been indispensable in reducing the burden associated with the definition of scenarios in perception studies we have conducted [12][2].

An important area of future work is to improve the automation capabilities of the system. This will focus on reducing the burden on the designer and ease the manual mark-up process within the current tool, which is dedicated to annotating pre-existing cities. A further area of research is towards a fully automated annotation system for procedurally generated cities [10][8]: in this domain, populating the resultant generated cities with plausible pedestrian crowds continues to be a key challenge.

ACKNOWLEDGEMENTS

The authors wish to thank all the members of the Metropolis team, in particular Simon Dobbyn and Cathy Ennis, for their collaborations during the project. We also wish to thank the reviewers for their comments which helped to improve the quality of the final manuscript. This work has been funded by Science Foundation Ireland.

REFERENCES

[1] P. Doyle and B. Hayes-Roth, 'Agents in annotated worlds', in *AGENTS '98: Proceedings of the second international conference on Autonomous agents*, pp. 173–180, New York, NY, USA, (1998). ACM.

[2] C. Ennis, C. Peters, and C. O'Sullivan, 'Perceptual evaluation of position and orientation context rules for pedestrian formations', in *APGV '08: Proceedings of the 5th symposium on Applied perception in graphics and visualization*, pp. 75–82, New York, NY, USA, (2008). ACM.

[3] Far Cry. CryTek studios. http://www.crytek.com, 2004.

[4] N. Farenc, R. Boulic, and D. Thalmann, 'An informed environment dedicated to the simulation of virtual humans in urban context', in *Proceedings of EUROGRAPHICS99*, pp. 309–318, (1999).

[5] A. Fuller, 'Urban modeling in games', in *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, pp. 148–166, New York, NY, USA, (2007). ACM.

[6] J. Hayes and K. Thompson. The creation of saints row's open world cityscape: Stilwater. GDC Presentation, 2007.

[7] T. Hostetler, *Controlling steering behavior for small groups of pedestrians in virtual urban environments*, Ph.D. dissertation, 2002. Supervisor: J. Kearney.

[8] G. Kelly and H. McCabe, 'Interactive city generation methods', in *SIGGRAPH '07: ACM SIGGRAPH 2007 posters*, p. 100, New York, NY, USA, (2007). ACM.

[9] I. Millington, *Artificial Intelligence for Games*, chapter Tools and Content Creation, 769 – 788, The Morgan Kaufmann Series in Interactive 3d Technology, Morgan Kaufmann, June 2006.

[10] Y.I. H. Parish and P. Müller, 'Procedural modeling of cities', in *Proceedings of ACM SIGGRAPH 2001*, ed., Eugene Fiume, pp. 301–308, New York, NY, USA, (2001). ACM Press.

[11] C. Peters, S. Dobbyn, B. Macnamee, and C. O'Sullivan, 'Smart objects for attentive agents', *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, (2003).

[12] C. Peters, C. Ennis, R. McDonnell, and C. O'Sullivan, 'Crowds in context: Evaluating the perceptual plausibility of pedestrian orientations', in *Short Papers Proceedings of Eurographics 2008*, Crete, Greece, (2008).

[13] G. Snook, *Game Programming Gems I*, chapter Simplified 3D Movement and Pathfinding using Navigation Meshes, Charles River Media, 2000.

[14] G. Thomas and S. Donikian, 'Virtual humans animation in informed urban environments', in *CA '00: Proceedings of the Computer Animation*, p. 112, Washington, DC, USA, (2000). IEEE Computer Society.

[15] B. Ulicny, P. De Heras Ciechomski, and D. Thalmann, 'Crowdbrush: Interactive Authoring of Real-time Crowd Scenes', in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2004*, pp. 243–252, (2004).

[16] J.S. Wiggins, *The Five-Factor Model of Personality: Theoretical Perspectives*, The Guilford Press, New York, 1996.

Multi-agent Systems and Sandbox Games

Sergio Ocio [1] and **Jose Antonio Lopez Brugos** [2]

Abstract.
In recent years, several games have presented non-linear gameplay systems; they are also known as sandbox games. Players are offered big, open, full of life worlds where they have a high degree of freedom to choose what they want to do to progress through the game. Multi-agent systems can help providing game designers with the means to achieve their creative visions and build more complex environments.

1 INTRODUCTION

First videogames were shown in 1970's. Since then, the growth of the industry has been exponential. For instance, videogame and console sales reached 18.85 billion dollars in 2007, far outpacing growth of film and music industries [1]. Sales also outstripped those of DVDs in 2008 [2]. Complexity has increased in a similar manner. From games like *SpaceWar!* or *Pong*, to the massive universes presented in recent games like *Fallout 3* or *Spore*, computers' evolution has provided creative teams with new tools to achieve their visions.

1.1 Motivation

Artificial Intelligence techniques used in a game are one of key factors that contribute to make it actually fun to play, while conveying a sense of an alternative, yet plausible, world where players want to spend their time in, which is the ultimate objective of every development team.

Building an AI system that is suitable for a videogame is a very complex and dependant on the nature of the game process [6]: for example, the AI of an strategy game is completely different to that in an FPS (First Person Shooter).

There is a growing divergence between what usually is the subject of study in educational scopes (such as machine learning or evolution), and what it seems the videogame industry is interested on (as film-like techniques, where a world, a story, is recreated, trying to make the player take an active role on it). A great example of this type of game is *Bioshock*, where players are presented with an alternate world, set up under the sea, where the attempt to achieve a perfect society, formed by the elite of humanity, destroyed itself. From the very first moment, when the main character suffers a plane crash, leading him to discover Rapture (the underwater city), the game is trying to convey a sense of grandeur and immersion: every step leads to discover the whole plot, as in a big Hollywood production (rather than a videogame).

Building such systems is a complex process, as they depend on hundreds of variables, and are also in a continuous relationship with players (who judge the effectiveness and quality of them every second). Ultimately, they pursue two defined objectives: achieve a solid AI and provide a fun challenge. Many times the most important thing is not to get an invincible enemy, but to balance "intelligence" and player's satisfaction adequately. It is, then, clear that the kind of AI techniques used in games are completely different to those used in other fields of science.

Here comes the dichotomy: do we really want to build an unpredictable AI, which can surprise players with unexpected actions, or are we looking for a system that just follows a predetermined script (although players do not lose the feeling of facing a real intelligence)? Some research is being conducted regarding this matter, so a Visibly Intelligent Behaviour [3] (i.e. one that not only solves problems effectively, but appears to be intelligent: it is perceived as a human-like intelligence) is achieved. So, the final goal is to try to avoid cases where evolution of agents [4][5] produces non-realistic results, even though they could be getting correct results (in terms of problem solving).

1.2 Sandbox games

Sandbox games have become one of the most successful type of games in recent years. They cannot be classified into any of the traditional genres, like, for instance, strategy, adventure, shooter, sports or driving games, and most of the times they are a conglomerate of different gameplay experiences. They can also be described as systems that allow players to do whatever they want, without having to follow a strict linear story.

Grand Theft Auto is the best example of what a sandbox driving game can offer. It presents a city where players can do almost everything they want: drive cars, cycles, ride trains, fly helicopters... The game is divided in different missions that do not necessarily follow a strict order, and represent a way to progress through the main plot. So, one of the keys to the success of the franchise is the freedom it offers, which has its foundations on the recreation of a whole city (or region), which is full of traffic, presents day/night cycles, is populated by many agents (police, pedestrians), has its stores...

However, this *life* is just an illusion: traffic is limited, police acts mostly triggered by scripts or a small subset of player behaviours, pedestrians only appear in a short number, etc. But not all of this is negative, as the game offers what it has been designed for. Nonetheless, it seems that sometimes that is not enough.

As games become more and more complex, AI systems beneath them must be updated to deal with a wider range of situations and offer human-like responses to the problems they are trying to solve. The creation of a game is an expensive process that can take several years of hard work. Thus, this paper presents a high-level view of what a multi-agent based city should be, decomposing it in its basic elements, which could be further studied in the future.

[1] University of Oviedo, Spain, email: sergiocio@gmail.com
[2] University of Oviedo, Spain, email: brugos@uniovi.es

2 INTELLIGENT AGENTS

Software agents have their origin in Software Engineering, Artificial Intelligence and Human-Computer Interaction fields. They have some roots in "actors", a technology presented in late 70's. They are, basically, objects with an internal state and the ability to communicate and interact with their environment. We can quote several definitions offered by AI researchers:

- An intelligent agent is a computer system that replaces a person or process to carry out an activity or complete a requirement. This entity is capable of making decisions, which are similar to those described by human intentions. An intelligent agent can operate within the limits of a general necessity or precisely represented within the limits of a certain area of information [7].
- From a general point of view, and considering its behaviour, an agent is something that perceives its environment through sensors and acts on it through effectors. Regarding rationality, an agent can maximise its efficiency based on the evidence offered by a sequence of integrated knowledge and perceptions it might possess. An intelligent agent is autonomous because its actions and preferences depend on its experience, rather than on external knowledge built in the environment created by its designer [8].
- An agent is a piece of software that is able to perform flexible and autonomous actions in certain environments. In that case, it is considered that flexibility has reactivity, proactivity and sociability, where [9]:
  - A reactive system maintains a continuous relationship with its environment and responds to its changes.
  - A proactive system focuses its behaviour on achieving certain goals. Therefore, it is not only controlled by events, but it can also take the initiative.
  - Sociability refers to the fact that as a multiagent environment, there are certain goals that cannot be achieved without the existence of inter-agent cooperation.

3 THE CITY AS A MULTI-AGENT SYSTEM

In order to convert a sandbox city into a multi-agent system, it is important to know what this type of systems are. According to Demazeau [10], they can be defined as an organised group of agents that interact in a common environment. So they are made up of four key elements:

- Organisation
- Agents
- Environment
- Interactions

3.1 Organisation

As the resulting system will be huge, an action hierarchy [11] will be used to try to organise the way things work.

Decisions are dispersed along the chain, going from long-term, high-level decisions, to immediate low-level ones. So, higher levels control the ones below them, while these inform their superiors about the state of the world. This will allow both logic organisation and performance to be improved. The proposed levels are:

- The city itself, which represents the highest level and can work as an agent perceiving whatever is happening inside it as an input.
- A control layer will be the second level, as a fundamental means of communication between the city and players.
- A third level composed by actual game's NPCs (non-player characters), as traffic, pedestrians, cops.
- The lowest level everything that can be seen as dispensable is found. However, this elements, shall they appear in the game, would increase the sensation of reality. Examples of this type of elements could be animals (cats, rats, birds...), public transportations (metro system)...

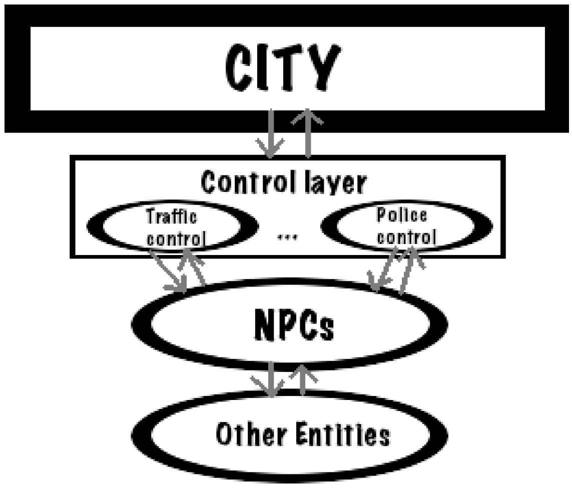


Figure 1. Structure of the city. Lower levels transmit their state to higher ones, while these emit orders.

3.2 Agents and environment

3.2.1 The City

The city is the brain of the system. The biggest difference with traditional multi-agents systems is that it is the environment and, at the same time, an agent, and as such it must be able to respond to different stimuli; these inputs would be bonded to different necessity conditions (city's health). Its reactions must, also, be adequate to players' game style.

This way the city would be ruled by two fundamental variables: equilibrium maintenance and adaptation to players.

Maintenance The city must maintain an optimal pace of operation at all times and avoid undesirable situations. For example, it should avoid severe traffic problems and maintain law enforcement in the city. This is achieved by creating the action plan for the level immediately below it in the hierarchy of the AI, i.e. the police.

While drawing up plans, the city must know how to cope with different situations, such as keeping an optimum number of policemen in the world to control any problem adequately, or modify its traffic lights patterns to solve traffic jams.

Adaptation to players Equilibrium maintenance in the city could be made even perfectly, avoiding any attempt to break it by players, but this could lead to a lack of interest for the game. Therefore, it is necessary to find a threshold for keeping the welfare state at acceptable levels, without disturbing players.

Moreover, not every player faces a videogame the same way, so it is interesting that the city takes into consideration players' styles to try and develop a strategy that maximises fun for players. For example, if a player breaks the virtual law he will be chased by AI-controlled policemen, and the system must decide whether to use an aggressive strategy, with greater numbers of units and/or more effective ones, or use an easier approach for the player, in case this has been classified as being reluctant to face these encounters.

3.2.2 *Control layer*

In order to be able to receive information about its state, and to maintain optimal stability values, the city must be able to communicate with the NPCs layer adequately. As several types of NPCs can be present, this layer would contain a set of different specialised controllers, like a traffic controller, a pedestrian controller, police controller...

Focusing on the former, police is managed at a high level by its controller, which prepares a general plan of action, such as "chase the suspect" for each individual; however, at a low level, police itself must be able to cope with problems using their own reasoning.

Each unit would be a police officer who can communicate with the whole city's police department: they know about general state of the city, because it has been passed to them, and perceive further information through their senses. They must also be able to travel between two locations in the city in an optimal way, as well as recognise what problem they are facing in every situation.

3.2.3 *NPCs*

They are undoubtedly the soul of the city, and those that make it feel alive. Moreover, they represent the AI which players would have most of the contact with. Each vehicle is controlled by an agent, which is, in fact, a pedestrian driving its car. Drivers must know their destination, which path to follow, comply with traffic rules (as traffic lights and pedestrian crossings), avoid accidents, try to optimize their trip (overtaking, changing lanes, searching for alternative ways...). In addition, each agent shall have its own personality, which would affect his driving style (aggressive or conservative, how much risk to take...), how they would behave against other agents...

Agent personality A system shows artificial personalities when it is possible to recognise that its characters are expressing personality archetypes. According to Ellinger [12]

> "A personality archetype is a set of clear, recognisable and consistent behaviours users can describe with a single word"

In order to achieve this, it is necessary, first of all, to define which archetypes shall be implemented. Each of them must be easily recognisable and different from each other, so it is not advisable for a videogame to present a extremely high number of them. Random behaviours must be minimised, as it is difficult to tell different archetypes apart when they are present. On top of this, an agent (character) must not change its archetype, unless the game's context calls for such mutation explicitly.

Strategy games usually feature archetypes easily differentiable, where each type of unit can present its archetype. For instance, a harvester, which just mines for resources and can be considered just a worker with no military force, or a soldier specialised in long-range attacks, that would try to avoid close combat with the enemy, which would make him more vulnerable. In *Black & White*, the archetypes shown try to imitate the behaviour of their owners (players), introducing basically two types of creatures: Good and Evil; although, in this case, the game will try to blend both archetypes to create a third one, depending on how inclined towards one of the sides the Creature is.

In a driving sandbox game, personalities will be most noticeable on agent driving styles. Each agent would have its set of personality traits, which would decide the way it drives. An example of personality characteristics could be:

- How likely the agent will drive on pavements.
- How likely the agent will ram other cars.
- How likely the agent will decide to make u-turns.
- How actively the agent will look for shortcuts.
- Whether the agent will prefer to drive at a fast or a slow speed.

Combining these values, different personality archetypes could be produced. This would allow designers to create a richer experience easily, whilst the code remains unchanged.

3.2.4 *Other entities*

This latest set of entities includes everything that makes the environment feel alive, but which absence would not affect the overall impression of the world.

For example, a dog could be implemented: if it is frightened by some noise, it would run in the opposite direction, without noticing the presence of a road, where an agent was driving his vehicle towards his destination; this agent shall add a new goal to its list, "obstacle avoidance", trying not to run over the animal, although this could end up causing an accident against another car.

This is, more possible situations could be added, improving gaming experience and the ability to surprise the player, but they would not add anything strictly required for the game to work.

3.3 Interactions

Once the agents, environment, and organisation are defined, a suitable communication system must be implemented. There are two different types of communication to be taken into consideration.

3.3.1 *Agent-environment communication*

Agents must be able to communicate with their environment, as this is one of the characteristics that define a multi-agent system. There are many options to do so:

- A first method is called Smart-Objects [13]; in this case, objects themselves pass their information to agents. This is similar to what is done in The Sims. This approach offers a clear separation between agent and object, as well as a knowledge decentralisation.
- Another method consist in taking interaction further, and develop a communication system that imitates nature [14]: agents will be able to interact with objects taking into account past experiences, that allow them to have some degree of knowledge about objects they had never seen before. SOTAI (Smart ObjecT-Agent Interaction) represents this method. Every object will be defined as a set of actions, each of them associate to a series of tokens. A token is a symbol that represents a stimulus. Agents will have several streams, or stimuli receptors. From this information, the system will build contexts, or cause-effect relations between these stimuli

and consequences. Contexts will group in strings, which represent knowledge nets and that make it possible to know the characteristics of an object from a set of input stimuli. Agents can use these nets to face new types of objects, that where unknown, more efficiently.

3.3.2 Communication among agents

Achieving a great communication between agents is really important, so they can work as sensors for the city. They can convey feelings, emotions, ask for help... and the city would be listening to this state (that is, in short, its own) and trying to act accordingly.

Communication among agents can be implemented as a messenger system [15], which can also be used for any other game system. A message is composed by an information type, emitter and recipient identifiers, as well as some additional information fields. A delay field can be added, so messages are not processed immediately after reception. A message would be similar to this:

```
// Message types
enum EMessageType
{
MT_MOVE,
MT_ATTACKED,
...
};

// Message
struct TMessage
{
EMessageType  type;
Identifier  senderID;
Identifier  receiverID;
float  delay;
...
};
```

Each game entity would implement a HandleMessage method, that would process received messages.

```
// Recipient
struct IMessageRecipient
{
virtual void HandleMessage(
const TMessage& msg )  = 0;
};

// Entity
class CEntity : public IMessageRecipient
{
...
public:
virtual void HandleMessage(
const TMessage& msg );
};
```

Using this kind of agent-to-agent communication, cooperation among agents can be achieved.

4 LOD AGENTS

Transforming the city into a multi-agent system, as described in the previous section, would require a huge agent count to be active all the time, which would produce a big overhead on performance.

AI, as any other subsystem of a game, would need to fit in the resource slot it has been assigned, so performance must become as priority as obtaining human-like behaviours.

LOD (Level of detail) is a concept mainly used in computer generated graphics. In a 3D environment (with perspective) objects will become smaller as the are taken further back on the scene. Thus, at a certain distance, using a worse quality model or texture will not affect the final result, and will save resources (as memory).

Extending this idea to AI, multi-agent systems, active agent count can be reduced, so they become less resource-intensive. Agents must be classified depending on their relevance (most of the times, their distance to players/cameras). Each of these groups represent a level of detail, would be ruled by its own logic and would, probably, we updated at a different rate, so those entities in lower levels are stepped less frequently [16] [17] [18].

4.1 Architecture

Once the different levels have been decided, a suitable architecture shall be chosen. This paper proposes a hierarchical model using agent controllers or *overlords* and the *mastermind*.

In this model, the highest LOD will contain fully functional agents. They will behave as regular agents and will be updated at the maximum available rate. Once the system decides an agent has become less important, it will be downgraded.

Agents in the second level of detail would lose their identity as individuals, and would be controlled by an *overlord*. This overlord is, in fact, a new type of agent, a controller one, which will act as a proxy between the environment and the agents it is managing, deciding what information is relevant for each of its minions and what actions shall affect the environment.

A *mastermind agent* forms the lowest LOD. It will act as a ruler for the *overlords* and decide when and how to update them.

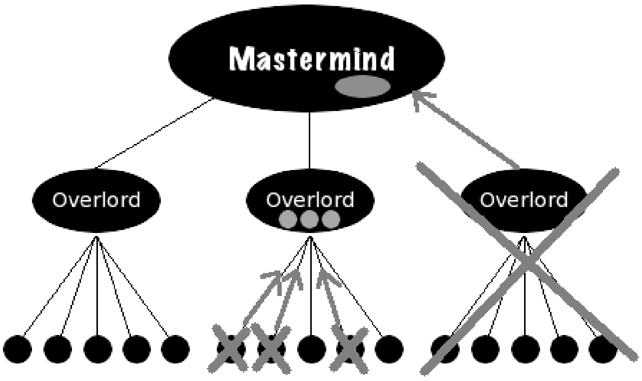


Figure 2. Overlord/mastermind architecture. Individual agents would become part of overlords as they are downgraded. The same would happen to overlords, which eventually would become controlled by the mastermind. This way, the number of active agents would be reduced.

So, contrary to what happens in graphics LODs, an agent would not be independent during all the states of its life, but would become part of a group at any time its behaviour is not directly noticeable by players. This way, the agent count would be dramatically decreased.

5 CONCLUSION

Videogames are a great challenge for Artificial Intelligence systems and represent a whole research field in themselves, trying to achieve a type of intelligence that is not looking for perfection, but to improve the game experience.

Videogames tend to present rich, full of life worlds, getting close to things that, so far, were only within the reach of films, although they are still at an early stage of development, and not always succeed in showing complex realistic environments.

Situations handled by an AIs are highly dependent on the type of game and its design requirements, and it is possible that a solution used in one videogame cannot be applicable in any other, which would lead to the development of specific techniques. Therefore, this is a field that offers a great potential in terms of researching and development of new algorithms.

The proposal of this work is to start a new research aimed at achieving more realistic sandbox games, building a city full of life, using, improving or creating specific algorithms or techniques built upon existing work relating multiagent systems.

Specifically, the research lines that could be followed are:

- Mastermind/overlord LOD architecture.
- Traffic control system, with intelligent agent-driven vehicles.
- Command hierarchy to manage conflicts.
- Conflict-solving planning.
- Personality and agent artificial emotions.
- Adapt AI behaviours to players style.
- Study a new methodology to implement agent-based AI systems.

REFERENCES

[1] Growth of gaming in 2007 far outpaces movies, music. http://arstechnica.com/news.ars/post/20080124-growth-of-gaming-in-2007-far-outpaces-movies-music.html

[2] Video game sales outstripped sales of DVDs in 2008, say analysts. http://www.telegraph.co.uk/scienceandtechnology/technology/technologyreviews/videogamereviewsandpreviews/4357957/ Video-game-sales-outstripped-sales-of-DVDs-in-2008-say-analysts.html

[3] Bryant, B. D. Evolving Visibly Intelligent Behavior for Embedded Game Agents. University of Texas at Austin. (2006).

[4] Miikkulainen, R., Bryant, B. D., Cornelius, R., Karpov, I. V., Stanley, K. O., and Yong, C. H., Computational Intelligence in Games. In Yen, G. Y. and Fogel, D. B. (editors), Computational Intelligence: Principles and Practice, pp. 155-191. IEEE Computational Intelligence Society. (2006).

[5] Stanley, K. O., Bryant, B. D., Miikkulainen, R., Evolving Neural Network Agents in the NERO Video Game. Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG05). (2005).

[6] Nareyek, A. AI in computer games. Queue archive. Volume 1, Issue 10 (February 2004), pp. 58-65. (2004).

[7] King, J.A. Intelligent Agents: Bringing Good Things To Life. AI Expert, pp. 17-19. (1995).

[8] Russell, S. y Norvig, P. Inteligencia Artificial: un Enfoque Moderno. Prentice-Hall. (1997).

[9] Wooldridge, M. Intelligent Agents: Introduction. 2nd European Agent Systems Summer School, EASSS2000. August 14th-18th. Saarbrucken, Germany. (2000).

[10] Demazeau, Y. Foundations of Multi Agent Systems. 2nd European Agent Systems Summer School, EASSS2000, August 14th-18th, Saabrucker, Germany. (2000).

[11] Reynolds, J., Tactical Team AI Using a Command Hierarchy. AI Programming Wisdom. Charles River Media, Inc. (2002).

[12] Ellinger, B., Artificial Personality: A Personal Approach to AI. AI Programming Wisdom 4. Course Technology. (2008).

[13] Abaci, T., Ciger, J., Thalmann, D., Action Semantics in Smart Objects. Proceedings of the Workshop towards Semantic Virtual Environments, pp. 121-126. (2005).

[14] Sequeira, P., Vala, M., Paiva, A., "What can I do with this?" Finding Possible Interactions Between Characters and Objects. Proceedings of the Sixth Intl. Joint Conf. on Autonomous Agents and Multiagent Systems. The International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS). (2007).

[15] Rabin, S., Enhancing a State Machine Language through Messaging. AI Programming Wisdom. Charles River Media, Inc. (2002).

[16] Kyo-Hyeon, P., Dong-Moon, K., Tae-Bok, Y., and Jee-Hyong, L. Real-Time Crowd Control using Fuzzy Steering Behavior. ISIS 2007. Proceedings of the 8th Symposium on Advanced Intelligent Systems. (2007).

[17] Brom, C., Lukavsk, J., er, O., Poch, T., and afrata, P. Affordances and level-of-detail AI for virtual humans. Proceedings of Game Set and Match 2, Delft. (2006).

[18] MacNamee, B., Dobbyn, S., Cunningham, P., and OSullivan, C.. Men Behaving Appropriately: Integrating the Role Passing Technique into the ALOHA System. Proceedings of the AISB02 symposium: Animating Expressive Characters for Social Interactions (short paper), pp. 59-62. (2002).

Implementation of Millenson's Model of Emotions in a Game Environment

William Blewitt[1], **Aladdin Ayesh**[1]

Abstract. Emotions, arguably, impact our perception and decision-making. Modelling emotions and emotional responses in non-player characters is an active field of investigation. Our work centres on the implementation of psychological models through fuzzy logic, as a way of informing agent decision-making in video games. This paper presents an emotion model proposed by Millenson, and the demonstration of this model in a game environment. We shall include an outline of the psychological background to our work, the specifics of the implementation, and some discussion of our conclusions.

1 INTRODUCTION

The development of interactive games, and video/computer games in particular, has provided an outlet to examine the subtle intricacies of human interaction and non-verbal communication. An important factor in these interactions is the emotional state.

The accurate modelling of the emotional state of any agent, and changes to that emotional state, has long been the subject of extensive research [15, 27, 29, 5, 2, 1, 14]. This work has varied in scope from the philosophical questions about why to make an emotional agent, to the sociological questions about what impact emotional agents have on human-computer interactions [6, 30]. Within computing sciences, such debate has tended to revolve around the practical questions regarding the mimicry of emotional responses and modelling of emotional state.

Within psychology, there are many schools of thought regarding the nature of emotions and how they are best modelled. We have performed an extensive review of these approaches, and further to this concerned ourselves with how they have been previously implemented in the field of artificial intelligence.

Recent publications in psychology have expressed a growing view that, rather than emotion being impedance to human creativity and decision-making, it is indeed an asset, if not a fundamental basis to the cognitive process [22, 21, 18, 19, 34, 23]. As this viewpoint begins to garner wider acceptance, it is expected that a renewed interest in emotion modelling and representation, from an academic standpoint, will follow, along with an exploration of emotion models that are not inherently linked to elementary control.

Our model is built upon the viewpoint that the emotion model itself should have the capacity to be wholly independent of the elementary control systems of the agent, instead being a construct to determine ongoing emotional state which is then called upon to inform decision-making.

A potentially key implementation of this concept lies in the decision-making capacities of non-player characters in game environments. These agents, being arguably the foremost means of direct human-agent interaction in the modern world, are naturally of significant interest to us. Bearing this in mind, our implementation in a game environment has been kept both simplistic, so as not to overshadow the theoretical implications, and intuitive, so as to ensure the work is clearly understood.

Within this paper we shall present some preface, discussing currently accepted approaches to modelling the emotional state of an agent, and how this work has fed into our own. We shall then proceed to outline the psychological emotion model we have selected, namely Millenson's theory of the behavioural school of emotions [4]. We shall present a method of applying fuzzy logic to the model itself, and the game implementation outlining the model's application through a simple, emotion-driven decision-making agent.

2 PSYCHOLOGICAL BACKGROUND

2.1 Preliminaries

Within the various fields of psychological research, two schools of thought appear to dominate the debate regarding the nature of emotions, and how they are best modelled [28]. From a philosophical perspective, the nature of their divergence and their theoretical differences are of great importance; from a computing sciences perspective, however, their differences lie in the nature of the models they propose.

The view of emotions as an evolutionary construct was initially proposed by Darwin in 1872 [8]. Over the past century, this has ultimately given rise to a school of thought which maintains that there are several fundamentally defined emotions, and that any given emotional state is a function of, or defined by, these emotions.

The exact number of 'fundamental' emotions widely varies. Plutchik first proposed his system of emotion classification in 1980 [24], containing eight fundamental emotions. In contrast, Ekman proposed a system consisting of six fundamental, or basic, emotions in 1982 [9]. Within this school of thought, the maximum number of basic emotions is generally thought to be fourteen [26].

Following on from the definition of basic emotions comes the definition of more complex emotions. Often these categories are divided using nomenclature indicating primary and secondary emotions as in the structure proposed by Parrott [20]. In general terms, however, it is the view of this school of thought that the sum of human emotional experience can be defined as a

[1] Centre for Computational Int., De Montfort Univ., LE1 9BH, UK. Email: {wblewitt, aayesh}@dmu.ac.uk.

function, or construct, of less than a dozen named emotions [12, 11, 10, 13].

An alternative to this view, first proposed by Wundt in 1904, suggested that emotions could be better defined in the context of experience than crisp linguistics [33]. Research based on this principle has given rise to many varied schools of thought following the same fundamental idea.

In Wundt's original model, emotional state was represented in terms of three facets of experience which he labelled 'pleasantness', 'approach' and 'arousal'. He asserted that any individual emotion would be better modelled in the context of relative magnitudes of these facets of the emotional experience than through inherently contextual verbal labels.

Subsequent to Wundt's original work, significant research has been performed regarding this idea of a 'dimensional' emotion model. In many cases it is common for the third axis to be ignored and, instead, for proponents of this view to model emotions in the context of 'valence', which might be seen as a clearer definition of 'pleasantness', and arousal.

More recently it has been suggested that these views are not necessarily mutually exclusive. Russell produced a circular model of emotions outlining the position of what he believed to be fundamental emotions in terms of relative values of what were effectively arousal and valence [25]. A geometric application of a Darwinian idea has been viewed as a reasonable progression [26].

It is upon this idea of a hybrid of the two major psychological schools of thought that we have focussed our attentions. In particular, on the works of Millenson in the context of his geometric model of emotions that combines specific, named emotional states with a three-axis geometric model.

2.2 Millenson Model

Millenson's model (1967) was presented as a standalone concept, rather than in conjunction with a particular theory of emotion as was common at the time. It is similar in construction to modelling of Watson's three-factor theory, and finds basis in the technique of conditioned emotional responding [31, 32, 28].

In his model, Millenson presents a three-axis representation of emotional state. Along each axis he places what he considers to be a basic facet of emotional experience. He abstracts three emotional factors which he believes represent the sum total of human emotion, the axes being defined, arguably, as emotions relating to anger, anxiety and elation, respectively.

Understanding that three emotions cannot truly represent emotional experience in explicit terms, he extends his behavioural analysis based upon two key ideas. Firstly, that some emotions are indistinguishable from each other, save in terms of their relative intensities. Secondly, that the emotions he presents are basic emotions, and other emotions are simply compounds of these [17, 16]; in this, his work carries noticeable analogue with that of Plutchik and Ekman.

The nine emotions Millenson lists as fundamental are divided into three groups, one group per axis, and can be summarised as follows:

Anger-related Axis
Annoyance
Anger
Rage
Anxiety-related Axis
Apprehension
Anxiety
Terror
Elation-related Axis
Pleasure
Elation
Ecstasy

From our perspective, Millenson essentially proposes that, based on his second condition, any given emotional state may be represented by a point within this three dimensional model or, as we are wont to refer to it, emotional statespace.

3 FUZZY MODELLING OF MILLENSON

3.1 Method Outline

We dub our system 'Fuzzy Millenson Emotion Model', or FMEM. The emotional statespace, while it can be visualised as a cubic region, can just as easily be described as a collection of three general factors, one for each axis, each containing three membership functions, one for each linguistically defined emotion. We follow from our previous work [4] in defining the mathematical representation of FMEM.

Let us consider these variables in terms of **A**, **B**, **C**.

$$\Re = \{\mathbf{A}, \mathbf{B}, \mathbf{C}\} \tag{1}$$

where

$$\mathbf{A} \subset \Re; \mathbf{B} \subset \Re; \mathbf{C} \subset \Re \tag{2}$$

Each of these variables contains one membership function for each linguistically defined emotion. We shall define these emotions in algebraic terms of consistent form. For example, a_1 is the first membership function within the **A** variable. This expands to:

$$\begin{aligned} a_i &\subset \mathbf{A} \text{ for } i = 1,2,3 \\ b_i &\subset \mathbf{B} \text{ for } i = 1,2,3 \\ c_i &\subset \mathbf{C} \text{ for } i = 1,2,3 \end{aligned} \tag{3}$$

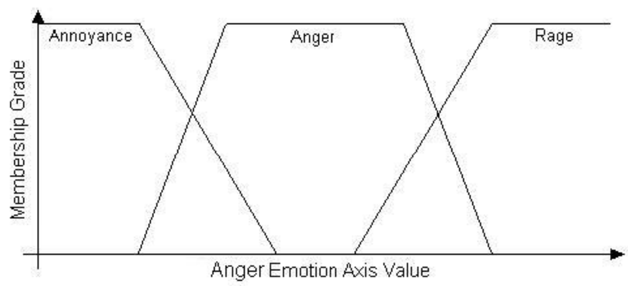


Figure 1. This figure shows the hypothetical form of an emotion variable in terms of individual fuzzy membership functions

An example of a potential makeup of the 'Anger'-related emotion axis is included as Figure 1. Any point within the emotional statespace is a column vector with values for all three axes, and a membership for each determining its applicability to the current manifestation of the statespace membership functions. We define this as an *emotional experience*. Hereafter, emotional experience shall be referred to as **e**, and defined

$$\mathbf{e} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4}$$

where x, y and z are coordinates along the 'Anger', 'Anxiety' and 'Elation'-related axes, respectively, and where

$$x \in \mathbf{A}; y \in \mathbf{B}; z \in \mathbf{C} \tag{5}$$

The membership grades obtained from a given e, are defined as follows.

$$\begin{aligned} \mu_{a_1} &= f_1(x) \\ \mu_{a_2} &= f_2(x) \\ \mu_{a_3} &= f_3(x) \\ \mu_{b_1} &= f_4(y) \\ \mu_{b_2} &= f_5(y) \\ \mu_{b_3} &= f_6(y) \\ \mu_{c_1} &= f_7(z) \\ \mu_{c_2} &= f_8(z) \\ \mu_{c_3} &= f_9(z) \end{aligned} \tag{6}$$

We should stress that the functions defining membership grades are not necessarily different across variables; the notation is written such as to provide the option if necessary.

Following on from Millenson's proposal that emotional state was a combination of these basic emotions, we may proceed to infer emotional state in terms of non-zero membership grades. Further to this, we should consider the factors defining how the emotional state **e** changes over time.

Our rationale for choosing to fuzzify the individual emotions rather than the emotional state is two-fold. Firstly, fuzzifying the emotions themselves gives us some measure of accounting for linguistic inconsistency in their definition as determined by intensity; in other words, fuzzifying the emotion Anger permits us to address the question 'how angry is angry?'. Secondly, by fuzzifying the emotions rather than the emotional state, we are able to obtain greater overlap between individual emotions across the same spectrum, making for a more varied and intricate fuzzy inferencing system and a more fluid change from one dominant emotion to the next.

For the purposes of clarity, the steps of the idealised model are illustrated below. Let us confine our axes to values between 0.0 and 100.0 Further to this, let us assign arbitrary values for **e** within these limits.

$$\mathbf{e} = \begin{bmatrix} 10.0 \\ 50.0 \\ 30.0 \end{bmatrix} \tag{7}$$

These values might produce the following membership grades.

$$\begin{aligned} \mu_{a_1} &= 0.4 \\ \mu_{a_2} &= 0.0 \\ \mu_{a_3} &= 0.0 \\ \mu_{b_1} &= 0.0 \\ \mu_{b_2} &= 1.0 \\ \mu_{b_3} &= 0.0 \\ \mu_{c_1} &= 0.8 \\ \mu_{c_2} &= 0.2 \\ \mu_{c_3} &= 0.0 \end{aligned} \tag{8}$$

This would give us the following non-trivial results from which to determine the subsequent behaviour of the system:

Annoyance: 0.4
Anxiety: 1.0
Pleasure: 0.8
Elation: 0.2

This is, of course, an idealised form of the model. For the purposes of our experimentation, the model is presented in a simplified form for ease of implementation on the chosen platform.

3.2 The Changing Emotional State

In the example we shall later present, we adopted a method of changing emotional state that created contextual relationships between environmental factors and specific axes of the model.

It is arguable as to whether this is preferable to a system that identifies specific events within the environment and attributes an emotional component relevant to all three axes to each event, since it necessitates the assumption that each axis is only affected by a specific environmental factor.

In an idealised system, each environmental factor would have an impact on all of the emotional axes, and those relationships would be exhaustively calculated such as to ensure that any event within the environment – which must, by definition, be measured in terms of its impact on the environment – is completely represented, even though not in specific terms addressing it.

For the purposes of this paper, however, we choose to link environmental factors to specific emotional axes. In the general case, we assign the environmental factors variables *L*, *M*, and *N*. This is true where

$$\begin{aligned} x &= f(L) \\ y &= f(M) \\ z &= f(N) \end{aligned} \tag{9}$$

which leads to

$$\begin{aligned} \mu_{a_i} &= f_i(f(L)) \\ \mu_{b_i} &= f_{(i+3)}(f(M)) \\ \mu_{c_i} &= f_{(i+6)}(f(N)) \end{aligned} \tag{10}$$

Thus with numerical values for the environmental variables, it is possible to define the agent's complete emotional state in general terms.

4 APPLYING FMEM TO GAMES

4.1 The Game Environment

For our game implementation, we opted to model an environment where an agent governed by the emotion model interacted with both a user-controlled element and an element governed by random path generation. To this end, using Game Maker™ 7.0, we generated a simple two-dimensional game. Figure 2 shows the starting layout of the game.

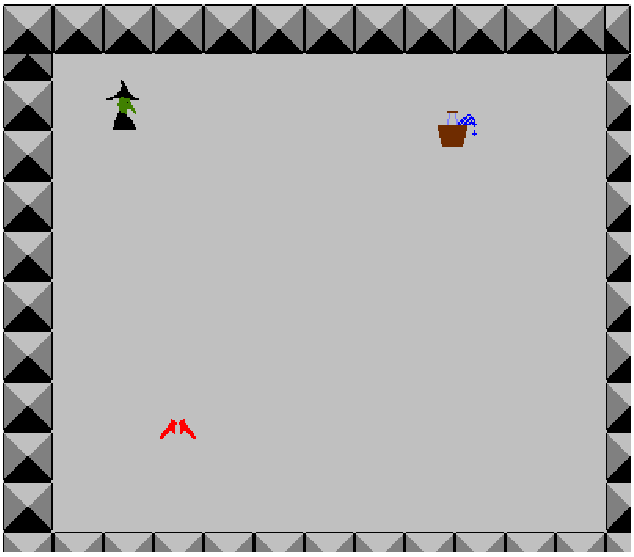


Figure 2. Screenshot of Starting Layout

We adapt the pedagogic case outlined by Ayesh [3] in the context of an agent, its food, and a predator.

- Bucket (Predator). The object in the upper right quadrant of Figure 2. This object is controlled by the player and moves in 25 pixel steps according to player input from the keyboard. The Bucket has an 80% chance to destroy the Elphaba object on contact, which is the player goal.
- Slippers (Food). The object in the lower left quadrant of Figure 2. This object is governed by a simple random pathing algorithm that causes it to spontaneously change direction, on average, every five movements of the Bucket. The Slippers have no objective.
- Elphaba (Agent). The object in the upper left quadrant of Figure 2. This object is governed by the emotion model (implementation specifics are outlined in the next section). The goal of this object is to come into contact with the Slippers, while avoiding contact with the Bucket. The speed and pathing of this object are managed entirely by the emotion model, with one exception; an override code ensures that the object will turn to a right angle leading it away from the Bucket if it impacts a wall.

4.2 Implementing FMEM

As has been indicated previously, the most straightforward implementation of the emotion model in a game environment requires us to connect environmental factors to each emotional axis. Now that our game outline is clarified, these factors can be outlined in context of the goals of the agent in question: Elphaba.

The first emotional axis we shall address is that concerning itself with elation-related emotions. Mindful that our agent's goal is to reach the Slippers, it makes sense for us to connect the proximity of the Slippers to the agent's sense of elation; the further away the Slippers, the lower the intensity of these positive emotions.

The second emotional axis we shall address is that which concerns anxiety-related emotions. Since contact with the Bucket will kill Elphaba, proximity to the Bucket should generate a heightened sense of these emotions; the closer the Bucket, the closer the agent comes to Terror.

The final emotional axis represents emotions relating to anger. That being the case, we opt to connect this emotion to an environmental factor extraneous to the activities of the other to objects: time. The longer it takes the agent to obtain the Slippers, the higher the value along this axis.

To simplify the implementation of our emotion model in this system, we choose to discretise our fuzzy membership functions and define our membership grades according to bands of values of x, y, and z such that multiple potential values of x share the same membership grade.

Figure 3 shows an outline of the implemented function of fuzzy membership for the pleasure-related axis. In this case, the value determining membership is distance from the Slippers; as such, the lower the value, the higher the membership of the more intense emotions on the axis. The solid line represents the emotion "Elation"; the dotted line represents the emotion "Ecstasy"; and the dashed line represents the emotion "Pleasure".

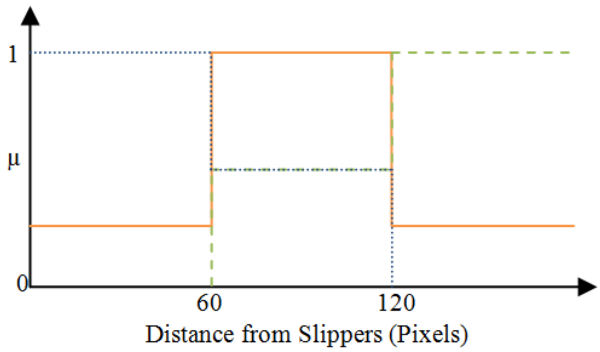


Figure 3. Fuzzy Membership Function (Elation-related Axis)

In implementing the rules for our fuzzy system, we used a method of scoring each rule for each iteration of the system, with the agent selecting the rule with the highest overall score. This is analogous to the process of a true fuzzy inferencing system, the key difference being the complexity of implementation due to inherent limitations within the development platform, and the crispness of the output.

For the purposes of clarity, our axes were defined thus: elation-related emotions, axis 1; anxiety-related emotions, axis 2;

anger-related emotions, axis 3. The three rules governing our agent's behaviour were as follows:

- Rule 1: If axis 1 is Ecstasy, and axis 3 is Anger, move towards Slippers
- Rule 2: If axis 1 is Pleasure, and axis 3 is Rage, move towards Slippers
- Rule 3: If axis 2 is Terror, and axis 1 is Pleasure, move away from Bucket

All behaviour of the Elphaba object was governed by these three rules, save the override which caused the object to change direction when impacting a wall.

5 TESTING AND RESULTS ANALYSIS

The program was run ten times from the same starting positions.

Figure 4 is a screenshot taken 22s into test 3, as the Elphaba object veers away from the Bucket, as the Rule 3 score begins to outweigh Rules 1 and 2. Figure 5 shows the Elphaba object successfully avoiding the Bucket and approaching the Slippers.

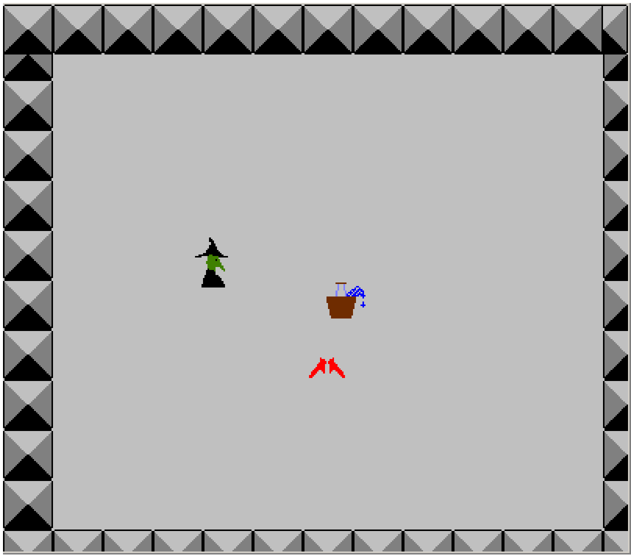


Figure 4. Screenshot of Test 3 as the Elphaba Object retreats

Six times, the Elphaba object was intercepted by the Bucket; the remaining four, the Elphaba object obtained the Slippers. The longest run of the program was test 6, where the Elphaba object was intercepted by the Bucket after 42s.

In this section we shall discuss notable behavioural patterns which were observed during testing.

Most notably, the Elphaba object reliably avoided the Bucket until Rule 2 overrode Rules 1 and 3, in all simulations where the Elphaba object was destroyed by the Bucket. This is due to the frustration effect that Rule 2 represents. Since play time is linear and has no ebb and flow, in the manner that proximity does, ultimately the rule governed by time will score higher than the others. In that situation, the Elphaba object could be perceived as being so frustrated as to gamble on the 20% chance that a collision with the Bucket will not destroy her. It should be noted that in test 6 this 'gamble' paid off and the Elphaba object passed through the Bucket and obtained the Slippers.

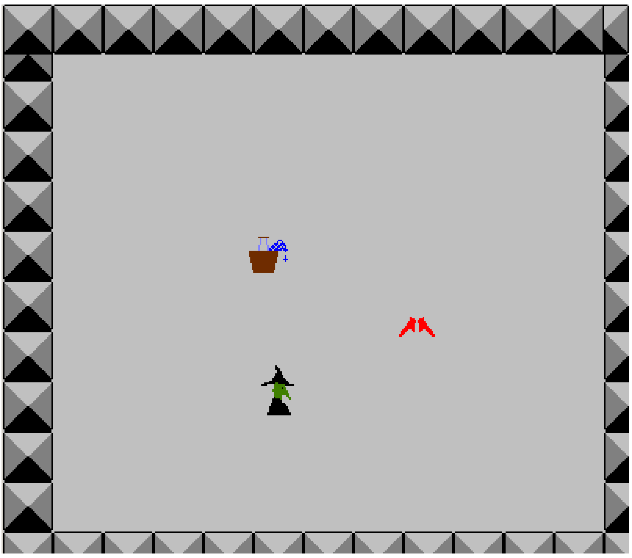


Figure 5. Screenshot showing the Elphaba object approaching the Slippers

Other repeated behaviours were highly predictable. When the Bucket was positioned next to the Slippers, the Elphaba object would hastily retreat, and then make another attempt. The same was observed when the Bucket was positioned between Elphaba and the Slippers.

An unexpected side-effect of the pathing override for wall collisions was that if the Elphaba object hastily retreated to a wall, it would follow the wall in whichever direction brought it closer to the Slippers, and make a beeline for the Slippers the instant Rule 1 outweighed Rule 3, circumnavigating the Bucket very effectively. This behaviour was observed in tests 2, 7 and 10.

6 CONCLUSIONS & FUTURE WORK

The paper has presented an emotion model for an AI-governed non-player character. The model has been shown to be grounded in psychological theory, and compatible with the general principles of a fuzzy inferencing system. The model has been implemented in a game format to govern the motion of an antagonist to the player (Elphaba).

The purpose of this research is to develop an emotion model that might serve as a guide to decision-making in agents that will experience ongoing interaction with users. This initial implementation has demonstrated that further investigation of this field has the potential to provide valuable insight. Time can now be devoted to expansion of the design of the system, and exploration of more advanced implementations that might better demonstrate its effectiveness.

The next stage in this research has two components. The first will be to explore the implications of two antagonistic agents whose movements are both governed by this emotion model. The second shall be a more traditional implementation of the model, with no simplifications, using a hardcoded fuzzy inferencing system and applying the model more extensively to an agent that would refer to it in multiple decisions, rather than simply pathing.

We shall also more extensively test the implementation of an emotion model to govern actions against alternative control methods, and in conjunction with alternative control methods, where the emotion model informs rather than dictates.

In addition, we shall continue to explore alternative psychological models of emotion that share the same basic groundings in psychological theory, and the potential implementation of higher order fuzzy logic systems [5], for comparison purposes. The increased level of fuzzification may permit a greater uncertainty of behaviour and, therefore, a closer analogue with emotionally erratic decision-making.

REFERENCES

[1] H. Ahn and R. W. Picard. Affective cognitive learning and decision making: The role of emotions. In The 18th European Meeting on Cybernetics and Systems Research, 2006.

[2] T. Archevapanich, B. Purahong, M. Klingajay, and P. Sooraksa. Facial visualization for robotic indicator by using fuzzy emotional system. In SICE-ICASE, 2006. International Joint Conference, pages 5634–5637, Oct. 2006.

[3] A. Ayesh. Perception and emotion-based reasoning: A connectionist approach. In Informatica, volume 27, pages 119–126, 2003.

[4] W. Blewitt, A. Ayesh, R. I. John, and S. Coupland, "A Millenson-Based Approach to Emotion Modelling" Conference on Human System Interaction (HSI'08), Poland, 2008.

[5] W. F. Blewitt and A. Ayesh, "Modeling the Emotional State of an Agent through Fuzzy Logic With Reference to the Geneva Emotion Wheel " European Simulation and Modelling (ESM'2008) Conference, Le Havre, France, 2008, pp. 279-283

[6] K. Boehner, R. DePaula, P. Dourish, and P. Sengers. How emotion is made and measured. International Journal of Human-Computer Studies, 65(4):275–291, April 2007.

[7] A. Camurri and A. Coglio. An architecture for emotional agents. Multimedia, IEEE, 5(4):24–33, Oct.-Dec. 1998.

[8] C. Darwin. The Expression of the Emotions in Man and Animals. Harper Collins/Oxford University Press, 1872/1998.

[9] W. V. Friesen, P. Ellsworth and P. Ekman. Emotion in the human face, chapter What emotion categories or dimensions can observers judge from facial behavior?, pages 39–55. Cambridge University Press, 1982.

[10] P. Ekman. Are there basic emotions? Psychological Review, 99:550–553, 1992.

[11] P. Ekman. An argument for basic emotions. Cognition and Emotion, 6:169–200, 1992.

[12] P. Ekman. Facial expression of emotion. American Psychologist, 48:384–392, 1993.

[13] P. Ekman. Genetic, Ethological and Evolutionary Perspectives on Human Development: Essays in Honor of Dr Daniel G. Freedman., chapter Expression or communication about emotion. American Psychological Association, 1997.

[14] N. Esau, E. Wetzel, L. Kleinjohann, and B. Kleinjohann. Real-time facial expression recognition using a fuzzy emotion model. Fuzzy Systems Conference, IEEE International, pages 1–6, 2007.

[15] C. Gershenson. Modelling emotions with multidimensional logic. In Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American, pages 42–46, 10-12 June 1999.

[16] J. R. Millenson. Principles of Behavioural Analysis. New York: Macmillan, 1967.

[17] J. R. Millenson. The Psychology of Emotion: Theories of Emotion Perspective, chapter 4, pages 35–36. John Wiley & Sons, 1967.

[18] R. Muramatsu and Y. Hanoch. Emotions as a mechanism for boundedly rational agents: The fast and frugal way. Journal of Economic Psychology, 26(2):201–221, April 2005.

[19] N. Naqvi, B. Shiv, and A. Bechara. The role of emotion in decision making: A cognitive neuroscience perspective. Current Directions in Psychological Science, 15(5):260–264, 2006.

[20] W. G. Parrott. Emotions in Social Psychology. Psychology Press, 2001.

[21] L. I. Perlovsky. Integrated emotions, cognition, and language. In Neural Networks, 2006. IJCNN '06. International Joint Conference on, pages 1570–1575, 16-21 July 2006.

[22] L. I. Perlovsky. Toward physics of the mind: Concepts, emotions, consciousness, and symbols. Physics of Life Reviews, 3(1):23–55, March 2006.

[23] E. Peters, D. Vastfjall, T. Garling, and P. Slovic. Affect and decision making: A hot topic. Journal of Behavioural Decision Making, 19(2):79–85, 2006.

[24] R. Plutchik. Emotion: Theory, research, and experience: Vol. 1. Theories of emotion, chapter A general psychoevolutionary theory of emotion, pages 3–33. New York: Academic, 1980.

[25] J.A. Russell. A circumplex model of affect. Journal of Personality and Social Psychology, 39:11611178, 1980

[26] K. R. Scherer. What are emotions? and how can they be measured? Social Science Information, 44(4):695729, 2005.

[27] T. Shibata, K. Inoue, and R. Irie. Emotional robot for intelligent system-artificial emotional creature project. In Robot and Human Communication, 1996., 5th IEEE International Workshop on, pages 466–471, 11-14 Nov. 1996.

[28] K. T. Strongman. The Psychology of Emotion. John Wiley & Sons, 1996.

[29] J.D. Velasquez. An emotion-based approach to robotics. In Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on, volume 1, pages 235–240, 17-21 Oct. 1999.

[30] G. Wang, Z. Wang, S. Teng, Y. Xie, and Y. Wang. Emotion model of interactive virtual humans on the basis of mdp. Frontiers of Electrical and Electronic Engineering in China, 2(2):156–160, 2006.

[31] J. B. Watson. Psychology. From the Standpoint of a Behaviourist. Philadelphia: Lippincott, 1929.

[32] J. B. Watson. Behaviorism. Chicago: University of Chicago Press, 1930.

[33] W. Wundt. Principles of physiological psychology. New York, NY: Macmillan., 1904.

[34] Marcel Zeelenberg, Rob M. A. Nelissen, Seger M. Breugelmans, and Rik Pieters. On emotion specificity in decision making: Why feeling is for doing. Judgement and Decision Making, 3(1):18–27, 2008.

Attacking the Knowledge Acquisition Bottleneck through Games-For-Modelling

S.J.B.A. (Stijn) Hoppenbrouwers[1] and P.J.F. (Peter) Lucas[1]

Abstract. Many model-based methods in AI require some sort of formal representation of knowledge as input. Acquisition of such formal models is either done manually, using a knowledge elicitation and modelling method, or automatically, applying knowledge discovery and machine learning techniques to available data. For the acquisition of highly structured, domain-specific knowledge, machine learning techniques still fall short, and knowledge elicitation and modelling is then the standard. However, obtaining formal models from informants who have few or no formal skills is a non-trivial aspect of knowledge acquisition, which can be viewed as an instance of the well-known "knowledge acquisition bottleneck". In addition, if there are social requirements on knowledge representations, e.g. constructive agreement on concepts and propositions, this poses a further challenge. Based on our work in conceptual modelling and method engineering, we propose to cast methods for knowledge modelling in the framework of games. The resulting games-for-modelling approach is illustrated by a number of examples from ongoing projects. Our chief long-term aim is to decrease the threshold for formal knowledge acquisition and modelling.

1 INTRODUCTION

In this paper we propose and illustrate an approach to knowledge acquisition and formalisation that does not primarily address the formal structures to be delivered, but rather the *process* of conceptualization and modelling, yielding formal models.

Formal knowledge models of some sort are essential in AI and related fields, not just as part of the theoretical foundations of the fields, but also for application: computation based on knowledge structures inherently demands some artefacts, which may vary from "lightweight formalisations" (e.g. diagrams or strictly structured text) to expressions with formal semantics. Focus in most cases is on the syntax and semantics of the formal artefacts, and on associated reasoning methods to apply them to problems. Obtaining such formal models is nowadays tackled in AI by using some knowledge acquisition (KA) and modelling method, such as CommonKADS, which suggests a step-wise approach, starting with informal, conceptual representations and methods and refining these until a formal model is obtained [1]. Although such methods were initially proposed as solutions to the *Knowledge-Acquisition Bottleneck* (KAB), experience shows that the KAB is as real as it was more than a decade ago when CommonKADS was proposed [2]. The enormous increase in the volume of knowledge discovery from data and machine learning research during the last decade, which was largely motivated by the appeal of *automatic* knowledge acquisition from data [3], is evidence that the KAB is as prominent nowadays as when it was first mentioned in the 1980s.

Within our current focus, the most urgent KAB aspects are the following:

- Knowledge is hard (and expensive) to make explicit, and even harder to formalise;
- The domain experts required for this job are usually not available for lengthy involvement in KA activities, nor do they possess the required modelling and formalisation skills;
- The expert knowledge engineers/modellers that could be hired to do the modelling job are few and expensive. Breaking the KAB by structurally employing expert modellers will only work in the most urgent of cases, covered by unusually large allocation of resources.

Many of the promises of AI concern a global user community of organisations and citizens that will never have access to expensive knowledge engineering experts. This will simply prevent many of the promises with respect to the wide availability of knowledge-based AI solutions from being fulfilled and, furthermore, it casts doubts about the future of the semantic web.

The practical problem of the KAB presents us with challenges that are urgent and interesting enough to warrant focused academic efforts for understanding and alleviating the problem. As acknowledged by Wagner [4], in fields like software engineering, information system engineering, and enterprise engineering, we are confronted with KAB-like problems on a large scale, and consequently solutions are actively sought.

In our research we are developing alternatives to existing knowledge acquisition and modelling methods. One idea we are exploring is to look at formal knowledge modelling activities as *games*, forcing ourselves to look at contextualised, operational modelling in which human factors are inevitably included. Because the way in which we employ games for formal knowledge modelling involves human-computer interactions (HCI), these games-for-modelling systems can best be tested using HCI-like evaluation methods, including existing methods specifically aimed at game evaluation. This combination, games-for-modelling and exploitation of HCI methods for evaluation, is, to the best of our knowledge, new to AI.

After providing a brief overview of related work, we will argue in favour of this approach, explain how methods can be viewed and designed as games, and provide some examples of such games, though admittedly only preliminary results on tested games are available as of yet. The paper is rounded off with conclusions of what has been achieved so far, and we offer a sketch of envisioned future work.

[1] Institute for Computing and Information Sciences, Radboud University Nijmegen, P.O. Box 9010, 6500 GL, Nijmegen, the Netherlands. Email: {stijnh,peterl}@sc.ru.nl.

2 RELATED WORK

We need to be clear about two distinct categories in KA: *automated* and *manual*. The first category uses knowledge discovery from data and machine learning techniques to derive models [3], the second depends on the construction of models by hand (aided by tools), by individuals or teams. Although there are certainly many situations where knowledge discovery from data and machine learning can be very useful, the fact must be faced that learning technology will not resolve the KAB for cases in which highly domain specific knowledge (ultimately kept in individuals' minds) has to be made explicit and formalised.

Early work on KA by Newell and Simon mainly focused on elicitation of verbal data collected from domain experts in the act of solving problems, called *think-aloud protocols* [5]. Useful knowledge was subsequently extracted from the protocols, using a technique called *protocol analysis*. Although the intention of protocol analysis was to obtain representations that could be manipulated by a computer, little attention was given to the actual semantics of the representations. The innovation by Newell and Simon was mainly to introduce techniques to AI which were originally developed in the area of psychology.

As mentioned above, in knowledge engineering, perhaps the foremost comprehensive method is CommonKADS [1], though many more exist. The essential idea is to work from informal, yet conceptually rich, models towards more formal models (using for example predicate logic), using a selection from a given set of *problem solving methods*. Problem solving methods can be best seen as generic methods that are aimed at solving particular tasks, such as diagnosis. A problem solving method can be instantiated for a particular domain, and the result is then a system that is able is to solve the task for a particular problem in the domain. Despite the large size and huge number of people years invested in CommonKADS projects, only a limited collection of problem-solving methods are being offered by the CommonKADS methodology. The researchers who were originally involved in the development of CommonKADS are no longer active in this area, and the methodology has never become the industry standard of knowledge acquisition and modelling. Knowledge modelling is nowadays also called ontology building [6].

Roughly similar approaches are also widely used in system development, e.g. RUP [7] (typically in combination with the UML [8]). They all make use of roughly defined, iterative phases in the modelling process, from exploration and informal sketching to formalization and implementation; also, they all, to a stronger or lesser degree, suggest or prescribe specific artefacts (descriptions, models) for particular phases and purposes, often involving strong structuring and/or specific modelling languages.

Useful as all this is (though the number and diversity of specific modelling languages is rampant), such deliverable-oriented textbook methods only provide very limited help for non-expert modellers in the actual execution of their modelling tasks; they still require considerable study and above all practice to be mastered. The availability of tools provides some help, but currently such tools are usually highly technical model editors that require in-depth technical knowledge, and do not actively assist in the act of conceptualisation and formalization of the models. In other words, they support *model-centric modelling* instead of *modeller-centric modelling*.

An additional problem with domain specific, manual KA is posed by the social context of domain specific knowledge modelling, which in many cases calls for intensive negotiation and validation of models by heterogeneous teams of stakeholders. In line with this, there is increasing interest in approaches for *collaborative modelling* [9,10]. Related issues are on the agenda in context of the Web 2.0 effort, and also in ontology engineering [11].

In [4,12,13], a conversation-based approach to knowledge modelling is suggested; in this vein, actual systems for modelling support have been created and studied, e.g. Wiki-based approaches [4] and a negotiation-based approach [10], the latter of which is most closely related to our own work, and in fact is an exception in that it does focus on the (conversational) process of modelling, or rather on supporting it. Yet, it still positions model editing (UML) as a central activity, assuming basic, diagram-oriented modelling skills to be available in the participants.

In view of the considerable challenge posed by the KAB, and focusing on manual KA, we find a dissatisfying lack of interest in issues that prevent real-life, operational modelling from becoming successful. In addition to focusing on representational issues (still the mainstream topic in literature on modelling), we believe that the situated act of modelling itself warrants study (starting from initial, sketchy conceptualization and moving on to actual formalization), including any relevant human factors involved. We may, for example, look at like usability/playability, learnability, even enjoyability, but also, of course, effectiveness and efficiency.

3 WHY GAMES?

Let us briefly explain what we mean by the *game metaphor*. People often refer to activities, tasks, or challenges (even complex, elaborate ones) as games, e.g. "the game of politics", "the game we play in this firm", "that sort of practice is not our game". At times, this referential metaphor is extended into actual identification or introduction of game aspects: competition, scores, declaration of winners/losers, rules, and so on. It seems justified, even fruitful, to use the game metaphor as well as actual game design as instruments in the study and development of tools for modelling support: games for modelling. This is in line with a well-established tradition of "Serious Gaming", prominently including management games [14]. We will now elaborate on our proposal to apply the game metaphor to thinking about modeller-oriented support systems for knowledge modelling.

In [15], a number of arguments are developed in favour of approaching the creation of operational methods/tools for modelling as game design. We briefly list the main arguments below:

Make formal modelling available to non-modellers As discussed, if low-threshold, domain-specific use AI is to really take off, large scale and low-threshold formal modelling *will* be required. An obvious but non-trivial way to proceed is to create software applications that make creation of required models as painless and efficient as possible: bring lightweight formal modelling to the masses through the virtual world emerging on the internet, and by shaping such applications as games.

Improve motivation of modellers

In the wake of Von Ahn [16], who managed to harness the creative energies of great numbers of on-line game players to perform "human computing", we believe it would be very helpful from both a methodological and a productivity point of view to make modelling more attractive (challenging, enjoyable), and thereby boost modelling in order to answer the needs and help bring AI to its full utilitarian potential. We believe games are a highly promising way of doing so.

Improve quality of modelling
More in line with common objectives in the field of knowledge modelling, a gaming setup may help improve the quality of the products of modelling, both *textual* (the models as such) and *contextual* (knowledge, understanding, agreement etc. across communities involved with models and modelling). Useful strategies for modelling can be built into the game design (e.g. shaped as sub-games, tasks, challenges) or be left to the participants (the players' strategies), as best fits the situation.

Tooling: virtual environments for collaborative modelling
The relation between digital tools/environments for modelling and digital games is obvious. Video games are highly advanced interactive systems. Completely virtual work environments may not be accepted on a large scale yet, but completely virtual multi-player games most certainly are. It is quite possible that the knowledge modelling tools and environments of the future feature serious game characteristics.

Apart from the above arguments that focus on the support of actual, operational modelling, there is one that concerns research and development methodology with respect to games for modelling:

Research and development approach: improving performance by improving game design
The game metaphor as well as the actual application of game design theoretical concepts will help *focus on the relevant research questions* concerning model oriented interaction systems and duly constrained modelling. Games can be tried and tested on various audiences, providing ample and well-structured data on interactions and results, and therefore offering an empirical hold on modelling processes that otherwise would be much harder to obtain in large volumes. This will enable modelling-oriented research using evaluative approaches from AI and HCI.

4 GAMES EMBODYING METHODS

For the link between methods (i.e. systems for modelling support) and games-for-modelling, we turn to Game Design Theory. Järvinen [15] provides clear concepts for analyzing and designing games that help greatly in performing game design (and therefore also aid method and tool engineering in a gaming context). Below we list generic game elements according to Järvinen, and add the equivalent thereof for the construction of methods.

1. **Components**: objects that the player is able to manipulate and possess in the course of the game. In methods this corresponds to any objects manipulated in the modelling process, typically brief fragments of natural language text (even individual terms) and elements of diagrams, including instantiations of modelling concepts. These are in fact the items now manipulated by means of editors; however, we expect that the explicit incorporation of more fine-grained *intermediary deliverables* in the process (related to taking smaller steps in conceptualization and formalization) will add to the number of different game components.

2. **Rule set**: rules produce each individual possibility and constraint that a game has to offer for its players, including *set goals* and *procedures*. In methods, such rules constrain the liberty of action of the modeller; one could say the rules *constitute* the method, plus situational goals set for a particular modelling job. In section 6, we will briefly return to this in view of a study into rule setting for modelling sessions.

3. **Environment**: the stage for game play. For example: a board, a field, or a virtual environment in a digital game. In operational methods, this can range from a meeting room to a whiteboard to a digital editor; in a completely digital (virtual) setting, interactive and possibly collaborative tools will be involved. Editor-like environments may be used, but beyond these, series of assignments may also be executed in less technical settings resembling virtual game boards or even 3D worlds.

4. **Game mechanics**: describe possible means with which the player can interact with game elements as she is trying to influence game states in order to complete a goal. For example: throwing in basketball, hitting in tennis; in more verbal games (and more relevant to our sort of gaming), proposing, asking, rejecting, and so on. The link with interaction mechanisms in operational modelling is obvious, but do note that game mechanics are not at all part of traditional (textbook) methods. In a conversation-oriented approach, the mechanisms associated with "verbal games" apply quite directly.

5. **Theme**: game theme is the subject matter that is used in contextualizing the rule set and its game elements to other meanings than those which the game system as an information system requires. For example: real-estate market in Monopoly, or a fictional context, or a historical event. For methods, setting themes is quite unusual so far (except for the actual, real modelling context as such). An inspiring yet rather radical idea for a theme would be, for example, *performing magic*, since this metaphorically corresponds nicely to applied knowledge modelling: "in order to get something (some service, information, prediction, and so on) you have to describe something precisely, conforming to procedures that the magic practice demands (ritual) and using the appropriate magical language."

6. **Information**: what the system and players need to know; the game state communicated. For example, a scoreboard, or a screen display, and/or component attributes such as value or number. In operational models, this can be the state of the model and procedural knowledge, but also feedback to the modellers (players) on the model (model checking, AI-based analyses) and the modelling process (status, progress, results, efficiency, etc.).

7. **Interface**: the tools to access game elements via game mechanics when direct (i.e. physical) access to game objects is impossible. For example, game pads, dance mats, mouse, steering wheels, etc. Though in operational modelling, rather standard games/systems interfacing is obvious, more innovative forms of interfacing may be worthwhile considering (e.g. 3D physical interfacing ("data gloves") or "surface computing").

8. **Player(s)**: the human factor in the game; their behaviour, mood, abilities and skills, relationship with games, game tastes. In modelling, this of course applies to modellers or other participants, and their competencies, interests, expertise, and preferences. Interestingly, player characteristics may be linked to specific roles, expertise, concerns, and preferences of participants (stakeholders) in the modelling process.

9. **Context**: the physical location of the game, the time, players' personal histories, and other informal, external aspects to the game system that possibly affect the experience of playing the game. In modelling, this refers to the situational aspects of a particular modelling task and session.

At least the following elements are minimally required to design a game (constituting a working *definition*): a) components complemented with rules governing their behaviour, b) an information structure to store the game states and component attributes and relations, c) at least one game mechanic to give players something to do, and d) a *goal* that the mechanics are designed to help completing, combined with *end or victory conditions*.

Goal setting is a key aspect of rule setting in games. In line with this, designing interactive games for modelling can be fruitfully driven by *goals of modelling*. This concerns both *utility goals* (i.e. what the model/modelling is useful for) and *modelling goals* (i.e. sub-goals pertaining to details of the modelling process as such).

The typical *utility* goals are, of course, knowledge creation, description, and formalisation, but additional goals include organisational and individual learning, consensus building; ultimately, they relate to typical strategic business goals involving investment and some sort of gain (commercial or otherwise). This aspect is too often ignored when academics get involved in actual application of KA.

For an overview of key *modelling* goals we refer to [17]. The chief goals are:

• **Creation goals**: which items (documents, objects, conceptualizations) are to be delivered when playing the game;
• **Grammar goals**: which language rules (syntax, vocabulary, possibly also semantics) does the player need to comply to;
• **Validation goals**: what sort of agreements, about which items, and between whom, is required in the game.

A number of sub-goals can be distinguished underneath the main goals, like argumentation goals, sense making goals, proof goals, abstraction goals, and so on.

Goals, sub-goals, and combinations of goals can be set for concrete modelling sessions or activities, involving one or more participants. *End goals* may be worked towards via *intermediary goals*. Strategies and techniques can be selected and deployed to achieve specific goals for concrete situations [17], in line with goals set but also with resources available and capacities and attitudes of participants. Once clear goals are set, and made more concrete by means of the definition of (a hierarchy of) combined assignments, challenges, etc., we can move towards actual game design. Importantly, the assignments given to the players need not overtly reflect the utility and modelling goals that the game designers have in mind. Any assignment that appropriately focuses, guides, and stimulates the player(s) will do. In fact, creative invention of (combinations of) appropriate assignments is key to successful game design.

This brings us to an issue that is possibly the one farthest removed from classical thinking about modelling: motivation. In the gaming world (both academic and industrial), much purposeful thought has gone into ways of making games captivating [18]. Indeed, game designers have now become so good at this that serious addiction is sometimes the result. If we allow ourselves to run with the devil for at least a few yards, we might learn something about how to make dull or hard tasks (including collaborative ones) more pleasantly challenging, more easily learnable and doable, and generally more effective. Perhaps modelling does not always have to be *great fun*, but we may at least succeed in making it *less boring* or *more positively challenging* for an audience not intrinsically motivated by the challenges of creating good formal models. The game design approach to the support of modelling thus creates opportunities for *designing motivation*.

5. GUIDING CONCEPTUALISATION

Contrary to what is often assumed, knowledge modelling is not just a matter of "translating informal into formal language" [19]. In addition, and perhaps more fundamentally, formalization requires rational, "clean" construction of representations according to utilitarian rather than associative principles. It entails rationally governed construction (engineering) of conceptual structures conforming to conceptual patterns dictated by some formalism. Such rational construction needs to take place before actual formal representations are produced, and possibly even independent from a specific formal syntax and semantics. Skilled formalists can perform such analysis and construction implicitly, and thus can produce formal representations (though perhaps sketchy ones) as an initial product. Laymen need a much more gentle, stepwise form of guiding and structuring. If procedures for achieving this can be successfully created, lightweight formalization can be achieved without confronting a player with any form of math, or even a semi-formal diagram. Models are then not elicited directly, but indirectly. After a guided conversation in which specific knowledge descriptions are elicited stepwise, conform rational principles (rules) governed by well-defined "goals for modelling", it should be possible to automatically derive formal representations based on strictly structured bits of natural language text, or simple visualizations, and the strictly governed relations between them.

From a method perspective this will force us to look at "pre-formal", *intermediary* products that may include information that does not belong in the end product, but which is used to *derive* the end product (formal model) by means of reasonably basic reasoning.

For example, as illustrated in the middle column of figure 1, a Business Process Model in the standard language BPMN (Business Process Modelling Notation [20]) typically shows an ordering of activities, e.g. activities D and E must be completed before activity F can be started. However, the reason why this is the case is that D and E respectively produce entities n and o that are needed in F (resulting in what is technically called an "AND-join"). This is illustrated by the text in the leftmost and rightmost columns of figure 1, in which these entities and dependencies are made explicit. However, such dependencies and entities are not made explicit in a regular BPMN diagram, even if they are crucial for creating a useful, "good" one. As a consequence, the entities and dependencies involved are usually left implicit and exist only in the head of the modeller –if you are lucky. Even if the objects in the process are made explicit, perhaps in another model, they are not explicitly used as a basis for deriving AND-joins.

This idea inspired our first design of a game-for-modelling (briefly discussed below), which is not to say it will be the basis for all such games. Exploration of possibilities has only just begun.

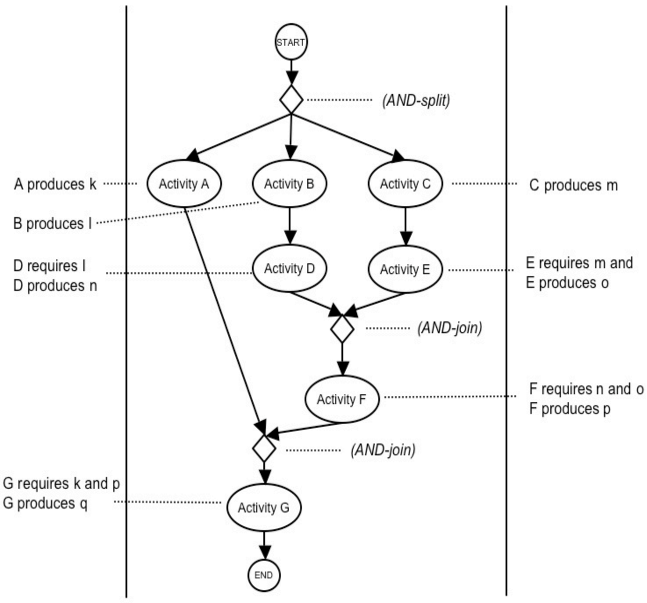


Figure 1: dependency information underlying a basic process model

6 SOME FIRST PROJECTS AND RESULTS

Various lines of work have been started in view of the "games for modelling" concept. The most fundamental line (in progress) concerns a methodology for the evaluation, in view of clear goals set, of modelling-activities as games. Rules are a crucial topic in this PhD project, directly linked to the game metaphor. The evaluation methodology is intended to serve as a key component in a design science cycle aiming at the development of principles and systems (i.e. games) for knowledge modelling support. The evaluation takes the shape of a transparent and traceable score system that is influenced by goals set for the game and weights assigned to them. This implies that not just the score system must be clearly (even formally) described, but also the game as such (i.e. all rules governing a particular interactive modelling session, *including* the goals set for it, in terms of concrete game results but also concerning collaboration, agreement, and efficiency).

As an initial part of the abovementioned project, we have performed an explorative study into "rule setting in modelling", also based on the game metaphor [21]. Using qualitative research techniques, we recorded, coded, and analysed a semi-realistic, 18-minute collaborative modelling session involving three modellers. Applying the game metaphor, we reverse engineered the session as a game, identifying precise goals and rules by which the game was played. Results allowed us to perform a preliminary comparison of game-like methodological concepts with related work on the quality of modelling [17] and collaborative modelling [10]. In addition, we became very much aware that in collaborative modelling, modellers discuss and introduce rules of their own, i.e. shape their game play together, as *part* of the game. This leads us to distinguish "rules set *for* the game" and "rules set *within* the game".

A second project that is well on the way concerns the design and implementation of a "Task Description Game", which is based on the idea presented in section 5. The game requires a player to describe some procedure (task; a favourite example is baking an apple pie) in terms of the items required to perform it and resulting from it, the steps taken, and various dependencies among steps and items used in/resulting from them. The player must fill in a number of form-like, interrelated "cards" (*item cards* and *step cards)*, and adhere to explicit rules governing the relationships between the cards and their content fields until she feels confident enough to risk a "try". After a try, a score is given at the hand of a set of scoring rules independent of the conceptual contents of the task description but rather guarding the rules constraining the description's components and their interdependencies. Penalties are dealt if rules have been violated, and another try may have to be prepared. Multiple tries are allowed, but more tries does mean a lower score.

After the game finishes, a simple algorithm helps derive a BPMN model from the structured information gathered in the game.

Various evolved board game-like versions of the game have been exploratively tested on a small scale. The initial design remained roughly intact, but a main lesson learned is that playability depends on many small details in rules, and in adequate communication of those rules to the player. For example, it has to be made *very* clear to the player of a game-for-modelling that it is necessary to play by the rules even if this is more difficult than free-format and perhaps more intuitive description of the task; also, that something is *gained* by doing this ("a computer understanding the description"). We are now in the process of applying basic principles from gaming-oriented HCI [22] to evaluate the finalised game more systematically. An implementation of a digital version of the game is on the way.

In this first attempt to design a game-for-modelling, we have not yet aimed at the game actually being *fun* (note that Järvinen's reasonably authoritative definition of "game" in section 4 in fact does not demand this), but merely for a *playable* game which should enable a layman to produce a simple formalisation "as a side effect". Once the game is sufficiently playable, we do intend to improve its design to increase the fun factor, using insights as discussed in [24]. However, we do not necessarily expect the game to become "very much fun"; we would settle for "mildly entertaining", as it also has a job to do.

Besides being a proof-of-concept for the idea of "formalization without formal language", the digital Task Description game should allow for empirical data gathering on a sufficiently large scale, as it can be exposed on-line to a large population of players.

More distant from knowledge modelling, but nevertheless strongly related, is the use of a game setup in testing a user friendly method for query formulation (the Interactive Query Language or IQL) intended to provide a more user friendly alternative for SQL [23]. In a game context, players use either IQL or SQL to answer questions by means of querying a fixed data set; results have led to preliminary proof that IQL speeds up the process of query formulation, and is easily learnable. Possibilities to extend the IQL approach to rule based modelling (in this case, Business Rules) are being considered.

Another project that is in its write-up phase concerns exploration of possibilities to introduce gaming aspects in an existing, operational industrial Business Engineering environment, with a strong AI interest (rule based modelling). Emphasis in this project is on focusing and motivating teams of business engineers (and others involved). The project is rendering considerable insights into design and evaluation of emotion and motivation, merging method engineering, HCI, and game psychology [24].

In addition, we are in the process of designing three more games, one aiming at value chain modelling, one concerning simulation (experiencing) and manipulation of enterprise architecture models, and one exploring possibilities to capture strategies for interactive formal proofing (natural deduction) within a game.

7 CONCLUSIONS AND FURTHER WORK

We discussed a Gaming approach to methods for knowledge elicitation and formalisation, in view of attempts to break, or at least widen, the Knowledge Acquisition Bottleneck. We discussed our perspective on the KAB, presented arguments for introducing the Game Metaphor, discussed the link between game design and operational method design, and provided an exemplary discussion of a conversation based, low threshold approach for guiding lightweight conceptualisation without using formal representations. Next we listed a number of projects within the frame of the Games For Modelling concept, thus providing some illustrations for it.

We realize that so far, no substantial published results have ensued. This is mostly because there simply has not been much time to do so: projects were all initiated less than a year ago. Still, we believe the concept as such is interesting enough to report on, and we hope it may fire up a discussion on the KAB and "methods embodied as games".

Further work was partly covered by the previous section, but in general concerns the continuation of our effort to design, test, and improve various sorts of games-for-modelling. This extends to generic, fundamental aspects like the development of adequate metrics for the quality of the games and the models they bring forth [17] and the development of design principles. In addition, we continue our exploration of theoretical contributions from AI, HCI, collaborative systems, and psychology/cognition (to name but a few) that might help us understand and further the creation of interactive systems for supporting formal modelling.

Acknowledgement We want to thank Denis Ssebuggwawo, Bart Schotten, Dennis Aarts, Jeroen Claassen, and Ilona Wilmont, to who's work we refer in section 6.

8 REFERENCES

[1] A. Th. Schreiber, J. Akkermans, A. Anjewierden, R. De Hoog, N. Shadbolt, W. Van De Velde & B. Wielinga: *Knowledge Engineering and Management: The CommonKADS Methodology*. Cambridge, MA, USA: MIT Press., 1999.

[2] F. Hayes-Roth, D.A. Waterman and D.B. Lenat. *Building Expert Systems*. PA: Addison-Wesley, 1983.

[3] M. Berthold and David J. Hand (Eds.): *Intelligent Data Analysis, An Introduction.* Springer-Verlag, Berlin, 1999.

[4] C. Wagner: Breaking the Knowledge Acquisition Bottleneck Through Conversational Knowledge Management. In: *Information Resources Management Journal*, 19(1), 70-83, January-March 2006.

[5] A. Newell and H.A. Simon: *Human Problem Solving* Prentice-Hall, Englewood Cliffs, NJ., 1972.

[6] M. Cristani, and R. Cuel: A Survey on Ontology Creation Methodologies. In: *International Journal on Semantic Web & Information Systems*, Vol. 1, Issue 2, p. 49 – 69, 2005.

[7] P. Kruchten: *The Rational Unified Process: An Introduction*. 2nd edition. Addison Wesley, 2000.

[8] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modelling Language User Guide.* New York: Addison Wesley, 1998.

[9] N. Kock and P. Rittgen, editors: Collaborative Business and Information Systems Design, special issue of the *Journal International Journal of e-Collaboration* (IJeC). Information Resources Management Association. Hershey, USA: IGI Global. 2009 (in print).

[10] P. Rittgen: Collaborative Modelling – A Design Science Approach, *Proceedings of the 42nd Hawaii International Conference on System Sciences* (HICSS-42), Waikoloa, Big Island, Hawaii, USA, January 5-8, 2009, CD-ROM, Los Alamitos, CA: IEEE Computer Society, 2009, 10 p.

[11] Diaz, A., Baldo, G. and Canals, G. Co-Protégé: Collaborative Ontology Building with Divergences. *Proceedings of the 17th International Conference on Database and Expert Systems Applications* p156 – 160. 2006. Washington, DC, USA IEEE Computer Society.

[12] S.J.B.A. Hoppenbrouwers, H.A. Proper, and Th.P. van der Weide. Formal Modelling as a Grounded Conversation. In: G. Goldkuhl, M. Lind, and S. Haraldson, editors, *Proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modelling (LAP'05)*, pages 139–155, Kiruna, Sweden, EU, June 2005. Linköpings Universitet and Hogskolan I Boras, Linköping, Sweden, EU.

[13] G.E. Veldhuijzen van Zanten, S.J.B.A. Hoppenbrouwers, and H.A. Proper. System Development as a Rational Communicative Process. In: *Journal of Systemics, Cybernetics and Informatics*, Nr: 4, Vol: 2, International Institute of Informatics and Systemics (IIIS), 2004.

[14] Elgood, C. (1993): *Handbook of Management Games*, 5th ed. Aldershot: Gower Publishing.

[15] S.J.B.A. Hoppenbrouwers, P. van Bommel, and A. Järvinen. Method Engineering as Game Design: an Emerging HCI Perspective on Methods and CASE Tools. In: *Proceedings of EMMSAD'08 (Exploring Modelling Methods for System Analysis and Design)*, held in conjunction with CAiSE'08. Montpellier, France, June 2008.

[16] L. Von Ahn: Games With a Purpose. In: *Computer* 39, 6 (Jun. 2006), 92-94.

[17] P. van Bommel, S.J.B.A. Hoppenbrouwers, H.A. Proper, and J. Roelofs. Concepts and Strategies for Quality of Modelling. In: T. Halpin, J. Krogstie, and E. Proper: *Innovations in Information Systems Modelling, Methods and Best Practices*; Chapter IX, p167-89. Advances in Database Research series, IGI Global Publishing, USA, 2008.

[18] K. Salen, and E. Zimmerman: *Rules of Play, Game Design Fundamentals*. Cambridge, MA: MIT Press.

[19] S.J.B.A. Hoppenbrouwers: Community-based ICT Development as a Multi-Player Game. In: *Proceedings of International Conference "What is an Organization? Materiality, Agency and Discourse"*, Montreal, May, 2008. Dept. of Organizational Communication, University of Montreal.

[20] Object Management Group (OMG): *BPMN 1.0, OMG Final Adopted Specification*, 2006.

[21] D. Ssebuggwawo, S.J.B.A Hoppenbrouwers, and H.A Proper: *Analyzing a Collaborative Modeling Game*, Submitted to the 21st International Conference on Advanced Information Systems Engineering (CAISE 2009).

[22] Heather Desurvire, Martin Caplan and Jozsef A. Toth: Using heuristics to evaluate the playability of games. In: CHI '04, extended abstracts on human factors in computing systems p15-12. New York: ACM.

[23] J. Claassens: *The Expressive Powers of an Interactive Query Language Compared to SQL.* Technical report, Radboud University Nijmegen, 2007

[24] I. Wilmont: *A Gaming Approach to Collaborative Modelling*. Master's Thesis, Radboud University Nijmegen, 2009.

‘Big Chief’: The Utilisation of Model Building for the Design of Science Education for Sustainable Development Games.

Timothy Barker[1]

Abstract. A model building paradigm was employed to inform the design of a game for Science Education for Sustainable Development (ESD). The intention is that such techniques should be readily achievable by university level students as they recontextualise game design to their locale. Hence, they not only produce a game which can be played in their community but they also begin to understand the complex processes underpinning the sustainable development of their own locale. Consequently, the game design process is elucidated together with one particular illustrative model. Finally, results for this model are presented before conclusions concerning the validity of this rationale are reached.

1 RATIONALE

There is a need to further engage both the public and the scientific community further in the issues concerning what has come to be known as Sustainable Development. It is true that today these issues have a degree of social currency yet aside from the sensationalism do people really understand the complexity inherent in a sustainable future? Furthermore, how does one communicate these complexities in simple terms in which individuals who may not have received the training in the necessary ways of thinking can understand with the minimum of effort? Further still, how does one contend with the many varieties of people who need to understand these issues in a global context? These are some of the questions for ESD.

2 OVERVIEW

In the context of this research project we are proposing that one way to communicate the diverse range and complexity of ideas to a large audience may be through the development of a game. There is a history of game usage in ESD and an academic journal dedicated to the theory of game design. More importantly though games have been shown to increase participants’ motivation to engage with educational materials [1]. That is, the association of games with fun and leisure time may be leveraged to help achieve some of the aims of ESD, notably higher acceptance of the concepts being explored.

However, we are proposing a two stage process in this game scenario. Firstly, having decided what the game should roughly be about students should adapt a (or construct an entirely new) model of the game playing process. This model should be based on local conditions wherever the game is hoped to be played. Thus, the idea is that the game itself will be more relevant to those who will be playing it thereby helping to increase its applicability to their lives and hence increasing acceptance. Further, if ideas for local interventions were to develop based upon the game then these also would be more suited to the locality.

Secondly, the same students who produce the model should use it to test the various interactions of variables in order to decide what the rules of the game should be. For example, which resources (e.g. people, fish, water, fertiliser) do the ‘pots’ represent, what kind of potential disasters (e.g. war or disease) are represented by the dice and how many beads (counters) are needed for each pot? All of these answers should become apparent when the model is developed.

With an idea in mind of the basic category of game (inspired by an African genre of game called Mancala) it is necessary to decide what your ‘pots’ and ‘beads’ represent. Further you should decide what is to be ‘painted’ on your dice. Questions you could ask yourself may be:

1. what factors influence how well the people do in the context, e.g. rainfall, good weather, animal migration paths, etc (these could be on your dice)?
2. what factors influence how poorly people fare in the context, e.g. disease, war, political situations, etc (these could be on your dice)?
3. what do people do? do they plant crops? do they walk for miles to a water hole? do they work? do they look after huge families?
4. what happens in the land around the people? do animals wonder freely and are hunted? does the river bed often dry up? which animals prey on others? which animals are useful to humans?
5. is there any industry in the area? are there huge farms nearby? are there any kinds of factories? do local people work at the factory? is industry (or people) polluting the area?
6. what do you know about the science of the area? is the lake saline? what is the soil composition? what is the most used form of energy? what is the ‘best’ form of energy?
7. what is healthcare like? is it a long walk to the hospital? is ‘western’ medicine used? is there a traditional healer nearby?

Box 1. Typical Student Model Design Questions

[1]DIYNGO (www.diyngo.org). Email: dr.timothybarker@gmail.com

3 THE MODEL

Box 1 shows typical questions for students to consider when designing their rules for the model. 'Pots' include 'people', 'crops', 'fertiliser', 'irrigation', 'eagle', 'water', 'fish', etc. Additionally 'people', 'crops', 'plants' and 'fish' each have two further 'pots' called 'r' and 'k'. The reason for these extra pots is that a formula known as the Verhulst equation is used to derive the cumulative population of these 'pots' over time. It basically states that a population grows by factor 'r' until a 'carrying capacity' (e.g. the amount of food or land, etc.) is reached. However, as the 'carrying capacity' is gradually being reached the 'growth factor' gradually diminishes. This is quite common sense as one could easily imagine that if there is little food to go around a population, for instance, then it is likely that they will grow less and of course the more they grow the less food there is. The result is a logistic curve, typically a sigmoid function (see Figure 1).

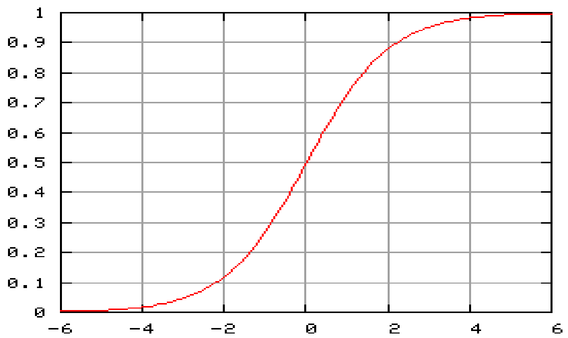


Figure 1. Logistic Curve[2]

Students were asked to develop the model in Excel. The first column in each Excel 'sheet' contains the heading 'year'. Thus as we descend the spreadsheet each subsequent row represents the next year. Hence we can see how the years, in our case 25, affect the growth or decline of 'beads' in the 'pots'. We use several sheets to represent several 'villages'. Theoretically, all of the 'pots' could be on one 'sheet' however, for the sake of legibility on the screen we utilise these various 'sheets'.

4 RESULTS

The model, once established, can be run any number of times as students explore the interplay of changing variables. For example, Figure 2 shows that a high amount of fertiliser entering the lake causes more pollution when the lake bank is not so healthy resulting in fewer fish for the eagles to feed on eventually leading to their extinction.

5. CONCLUSIONS

Model building can be a joyful experience particularly as one experiences the interplay of variable, or 'pots' in our case, which were unforeseen. This is, in some ways, the joy of the process of scientific discovery. On the other hand, when one is tweaking formulae and trying to mimic nature in all its glory through the divination of mathematical creativity this joyful process can, at times, become rather annoying. However, it is worth persevering during the most painful moments of creativity as, as ever, the end product can be well worth the pain.

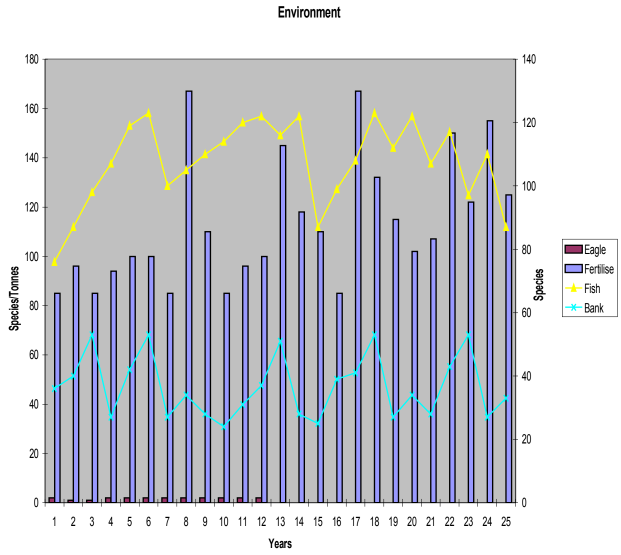


Figure 2. A Bad Year for the Eagle
(Fertiliser = 30, Land = 10 for all 3 chiefs)

Likewise, the problem with building a model is that the work is never done. However, we all know that models are *representations* of reality and, as such, are *approximations*. This is worth bearing in mind as a result is interpreted and subsequently acted upon. For this reason a model should be 'run' a great deal of times and results aggregated over these 'runs'. Further still, the well practiced Scientific method of consensus amongst peers could also be of use. More importantly the originator(s) of the model would do well to point out its weaknesses and limitations before, what may turn out to be, unfounded conclusions are acted upon.

In this light it is wise to point out that this model has many limitations. Indeed its correlation to the reality it purports to represent is, at best, of such a coarse scale that, in terms of scientific accuracy it may be dismissed by some out of hand. However, this model is an **educational tool**, it is a vehicle with which to commence discussions, to stimulate interest in the fundamental principles underlying Sustainable Development. That is, that social, economic, environmental and cultural (at least) systems are all interrelated and that misusing one can have untold effects upon the others. As such we believe that it, and more importantly the method, are useful aids for ESD.

REFERENCES

[1] Barker, T. & Morrison, E.(2008). How to Make Physics More Interesting or Learn Science and Save the World! *JETL*, University of Leicester (see www.timothybarker.com)

---

[2] http://en.wikipedia.org/wiki/Verhulst_equation (accessed 12/2/07)

Camera System for Interaction in Golf Game

Ki Hyun Kim[1], Chang Ok Yun[1], Hyun Woo Park[2], Woo Suk Joo[3], Dong Hoon Lee[3] and Tae Soo Yun[3]

Abstract. We propose the camera system for user interaction through flight data measurement of the golf ball by using line-scan camera and high-speed cameras in the golf game. And then line-scan camera is checked the fly or no of the golf ball and flight status of the golf ball takes image by using the high-speed camera. Therefore, we can confirm the progressive direction and spin of the golf ball through the image processing.

1 INTRODUCTION

As golf games become more popular, technology that recognises the operation of the real user has also advanced. The more commonly used system is the optical sensor, however optical sensors are expensive. In addition, measurement errors can easily result depending on the flight of the ball and judgement errors due to minute dust or humidity. In order to overcome these limitations, we propose the camera system for flight data measurement of the golf ball by using a line-scan camera and high-speed cameras in the golf game. Through the camera system, we can more accurately measure the progressive direction and spin of the golf ball.

2 CAMERA SYSTEM IN THE GOLF GAME

Our system consisted of one line-scan camera, an image capture board and two high-speed cameras in which several exposures are possible when one image is taken.

Figure 1. Camera System

In our system, we use the line-scan camera to check whether the golf ball has passed or not. By controlling the illumination we set up an environment that makes the flying golf ball well seen. After the captured image rapidly judges whether the ball has passed or not, it sends a trigger signal to the high-speed camera to take a photograph. The image received in the high-speed camera measures the flight data of the golf ball through image processing. This image is equivalent to an image captured by a high-speed camera of more than 1,000fps. After the location information of the golf ball is measured, the speed and a direction are calculated using the physical formula. This information is then applied to the golf game.

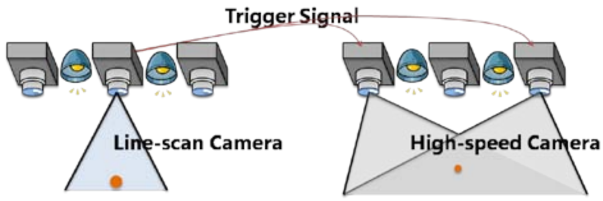


Figure 2. Principle of the Camera System

2.1 The Line Scan Camera

We use the line-scan camera for checking whether the golf ball passed or not. The line-scan camera that spec is resolution of 1024×200, gray image and frame rate of 25 KHz, is able to capture the golf ball although it is fast. Figure 3 is the image which the golf ball passes in the line scan camera. After the captured image rapidly judges the weather passed or not, it has to send the trigger signal to the high-speed CCD camera for it taking a photograph. Therefore, image processing of the line-scan camera's image has to be simple. So, if the bigger value of line-scan camera's image than 40 was 300, it determined that the golf ball passed and the trigger signal was transmitted to the high-speed CCD camera.

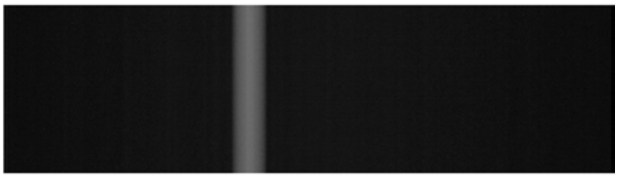


Figure 3. The golf ball passes in the line-scan camera

2.2 The high-speed camera

The high-speed CCD camera overlaps the image of many cuts on the image of the leaf, because it can acquire the image with the fast speed. The high-speed CCD camera acquires image when, it received trigger signal from the line-scan camera. At this time, the Sensor Exposure value is defined by the Exposure Numbers, the Exposure Duration, and the Exposure Interval. Exposure Numbers; the exposure number definition of the camera lens the Trigger signal once received. Exposure Duration; the defining

[1] Dept of Visual Contents, Graduate School of Design&IT, Dongseo University, Busan, Korea. Email: khkim@dit.dongseo.ac.kr, coyun@hanmail.net.
[2] Regional Innovation Center for Ubiquitous Appliance, Dongseo University, Busan, Korea. Email : phw1010@gdsu.dongseo.ac.kr.
[3] Dept of Digital Contents, Dongseo University, Busan, Korea. {savrang, dhl, tsyun}@dongseo.ac.kr.

the exposure time of a camera. Exposure Interval; duration of the next exposure of a camera. (Figure 4. reference)

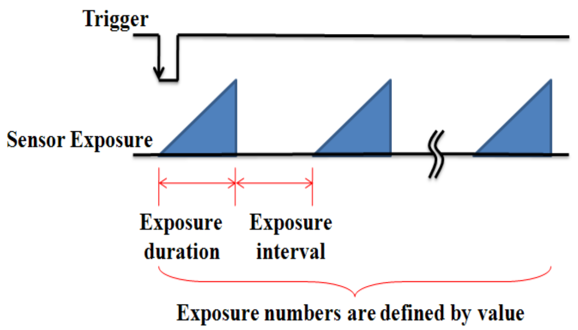
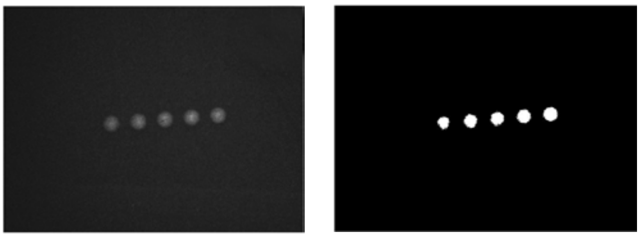


Figure 4. The photographing method of the high-speed camera

The obtained image one by one classifies five balls after performing the image processing. At this time, in the result image, because it is not successively enumerated from the right, by using the selection sort, the indexing of the golf ball enumerates as the order of a left from the right (Figure 5. reference).

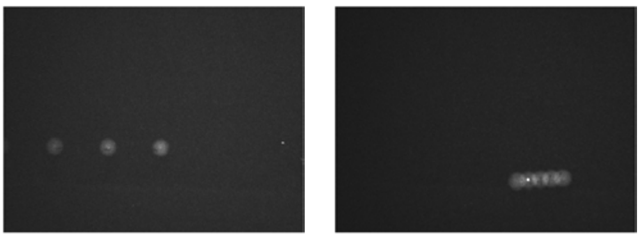


(a)Before image processing (b)After image processing

Figure 5. The captured image by the high-speed camera

2.3 The exception handling of captured image

It is faster than the Sensor Exposure set value of the high speed CCD camera or there is the slow case in the image acquisition according to the Sensor Exposure set value of the high speed CCD camera. The golf club and method of the golfer hitting a golf ball are mistaken according to a distance between the golf ball and a hall, a location. At this time, an affect is caused in the speed of the golf ball.



(a) The fast case (b) The slow case

Figure 6. The difference of The Sensor Exposure set value of the high-speed CCD camera.

In case the first, the shapes of five balls aren't photographed because the golf ball are faster than the Sensor Exposure set value of the high speed CCD camera (Figure 6(a) reference). Because this kind of case calculates by using the difference of the time and distance between the golf ball of first and the second (based on the right), it is able to make a speed and angle with calculation. In the second case, the golf ball are slower than the Sensor Exposure set value of the high speed CCD camera, the overlapped image of the golf ball is obtained (Figure 6(b) reference).

In such case, since knowing the set value of the Exposure Numbers, it can do in the size of one blob with assumption whether several golf balls are overlapped or not. Here, we can calculate Y Coordinate at the right and left of a blob and by using the Y-axis projection technique. We cannot be known about the progressive direction of the golf ball in the case of one blob. However, because a distance is known about the diagonal line element of the golf ball, we can calculate a speed of golf ball. At this time, the first value in the Y-axis projection sets as the canter of the golf ball at the Y-axis. By using the trigonometrical function and the physical formula of S=VT, the coordinate of a ball obtained through this kind of process makes a speed and angle with calculation.

3 CONCLUSIONS

We proposed the camera system for measuring flight data of the golf ball on the screen golf game system. We were also able to see the appearance of the flying ball through direct application of the physical simulator to our data. Our camera system can be applied to various sports games that use a ball, such as baseball.

Figure 7. Playing the Golf Game

ACKNOWLEDGMENT

This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

REFERENCES

[1] Kihyun Kim, Hyunwoo Park, Donghoon Lee, Taesoo Yun, “The Camera System for the Flight Data Measurement of the Golf Ball in the Screen-Golf”, Proceedings of MITA 2008, pp.423-426, 2008.

[2] Sang Hyuk Ahn, “Development of a simulator for Interactive 3D Golf Game”, Hallym University, Master’s thesis, 2006

[3] David M. Bourg, “Physics for Game Developers” : O’Reilly

Multi-touch Display System for AR Card Game

Sang Heon Han[1], Chang Ok Yun[2], Tae Soo Yun[3] and Dong Hoon Lee[3]

Abstract. We propose FTIR (Frustrated Total Internal Reflection) based on the multi-touch display system for playing the AR (Augmented Reality) card game. Existing card games request simple input in the form of a button to interact with a user and to show information that varies with the information or the position of the card. In order to overcome the limitation of these interactions, we present a tangible user interface by applying the multi-touch screen system which can input the pattern of the various forms. In this way we can offer more interaction and higher engagement for a user.

1 Multi-touch Display System

In this paper, our system has three modules as shown in Figure 1. The first module is invisible marker tracking, and the second module is FTIR based on multi-touch tracking. The third module is SAR (Spatial Augmented Reality) display.

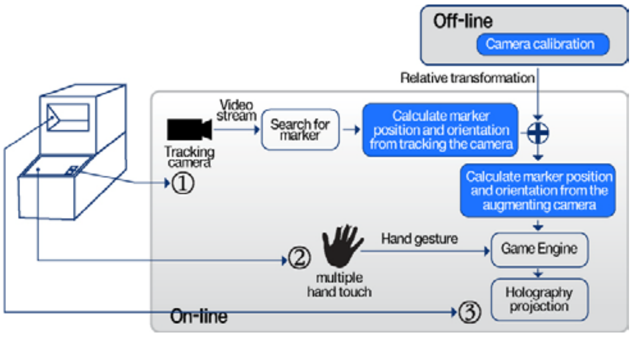


Figure 1. System Flow

Firstly, we present FTIR[1] based on the multi-touch screen that is made with the table top form for a tangible user interface (Figure 2). FTIR method is based on optical total internal reflection within an interactive surface. After the light of the infrared LED causes a reflection in the acryl inside, the infrared light escapes and is reflected at the finger's point of contact due to its higher refractive index. An infrared-sensitive camera at the back of the pane can clearly see these reflections. Secondly, we present the module for recognizing the card above a screen (Figure 3). Then after the infrared reflective sheet applying the specific pattern of a shape is plated to the card (Figure 4), by using the DI (Diffuse Illumination) method, the information of the card is recognized.

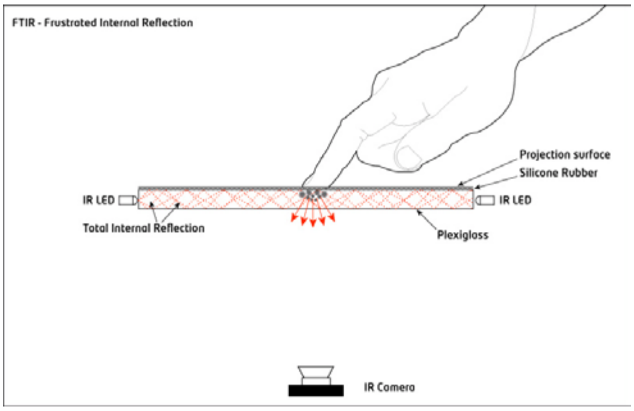


Figure 2. FTIR : Frustrated Total Internal Reflection

At this time, we utilize the infrared camera that we used inside the multi-touch system. With this method, there are almost no additional expenses for the infrared ray card recognition or the hardware configuration.

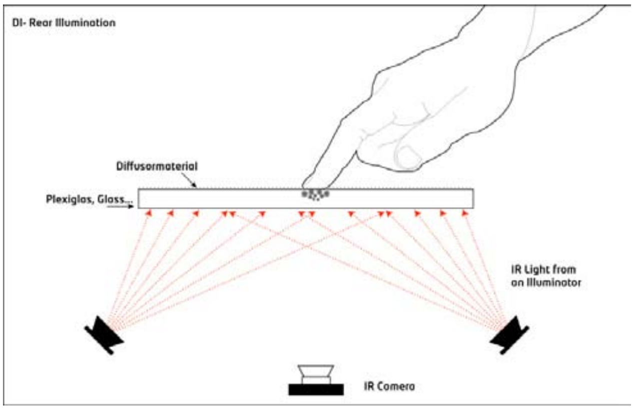


Figure 3. DI : Diffuse Illumination

Figure 4. Simple pattern for card recognition.

[1] Dept of Visual Contents, Graduate School, Dongseo University, Busan, Korea. Email : `alpha815@gmail.com`.
[2] Dept of Visual Contents, Graduate School of Design&IT, Dongseo University, Busan, Korea. Email: `coyun@hanmail.net`.
[3] Dept of Digital Contents, Dongseo University, Busan, Korea. `{tsyun, dhl}@dongseo.ac.kr`.

The Society for the Study of Artificial Intelligence and Simulation of Behaviour Convention 6th - 9th April 2009, Edinburgh, Scotland

Finally, in order to provide not only the expansion of an interface but also the visually high captivation for a user, we apply the SAR[2] method for a viewer looking at the stereoscopic image in the monoscopic image (Figure 5). Then in the multi-touch screen area, by using various interfaces, the image showing on the image output unit enhances the interaction between the contents and the viewer.

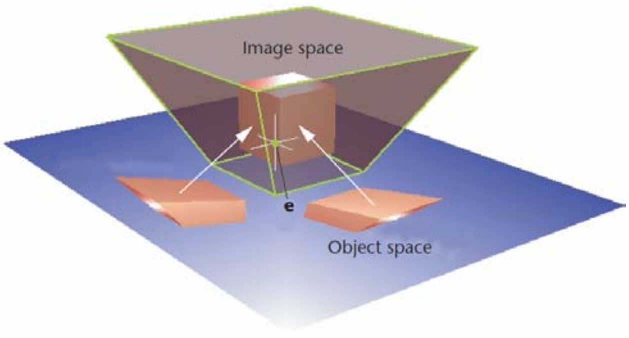


Figure 5. Virtual Showcase for SAR

2 Conclusion & Future Work

We proposed a multi-touch screen system for playing the AR card game. Our system can be applied to various interactive contents. In this paper, our system is limited due to 'large size' with the projector base. And we don't need a FTIR method; because of DI method is possible to detect both finger point and card. Therefore, we plan to develop the multi-touch display of the LCD base to further advance this system, and to alternate system for totally DI method.

Figure 6. Playing AR Card Game

ACKNOWLEDGMENT

This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

REFERENCES

[1] Jefferson Y. Han, "Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection" UIST'05, 2005, Seattle, Washington, USA, 2005.

[2] Bimber, O., Frohlich, B., Schmalstieg, D., and Encarnacao, L.M., "The virtual showcase", IEEE Computer Graphics & Applications, vol. 21, no.6, pp. 48-55, 2001.

Interactive Content System based on Spatial Augmented Reality and Multi Touch Screen

Kim Jung-hoon[1], Kim Ki-hyun[1], Yun Tae-soo[2] and Lee Dong-hoon[2]

Abstract. In this paper, we proposed interactive content system based on spatial augmented reality and multi touch screen. The system is designed to help people get information to easy way. People who want get information they just touch the screen, then can see the result through the spatial augmented reality display. As the result, People get information while they enjoy this system.

1 INTERACTIVE CONTENT SYSTEM

These days, we used many kinds of interface for input and output device. Generally, button and switch was the popular input device. However it changed to touch screen. One of the reason is that touch screen is easier than button and switch. When we use touch screen, we do not need broad space for a lot of buttons and switches. Touch screen will be changed for each situation. Output device is also changed like input device. We used braun tube for the monitor before the LCD is popularized. However, those are only 2D display. For the sense of the real, many techniques are developed like 3D monitor.

In this paper, we proposed interactive content system based on spatial augmented reality and multi touch screen. For this system, we application two type of technique.

- MULTI TOUCH SCREEN
  It is wrapped with rear-screen onto acrylic board for multi touch. For make multi touch screen, use 8mm acrylic board and 880nm Infra-Red LED (IR LED).

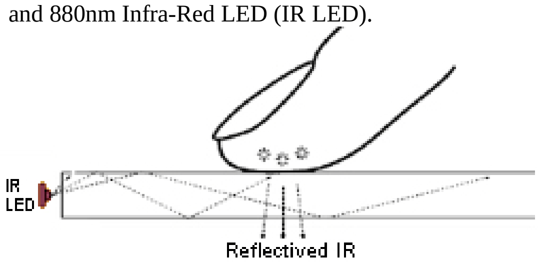


Figure 1. Movement of IR rays

IR rays are gone into the acrylic board and it diffused reflect. However, if we touch the acrylic board, then IR rays are pop-up outside. At this time, we can see the touched point through IR camera. The IR camera was removed IR-Cut filter inside a normal webcam, and use IR filter. Then the camera only can capture IR rays, without any visible rays. As the result, we should not care about displayed image from projector. When the camera checks the touched point, we calculate the captured point to windows point; like a mouse cursor. At the calculation, camera's coordinates are changed to windows coordinates.

- SPATIAL AUGMENTED REALITY
  We use the half-silvered mirror (Figure 2.b) that is one of the display ways for the spatial augmented reality (SAR). Spectators (Figure 2.e) can watch not only the diorama (Figure 2.a) through the half-silvered mirror but also the reflected image (Figure 2.c). The reflected image seems to be displayed mixed with the diorama, but it is a mirror reflected image of a monitor, which is built in Figure 2.c.

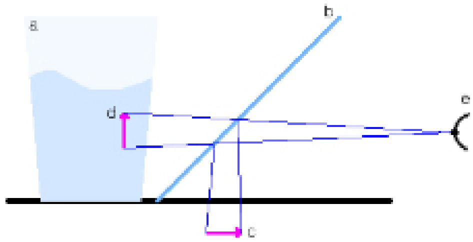


Figure 2. SAR based on half-silvered mirror

However, the reflected image is differently distorted by the leaning degree of the mirror. This undesirable situation can be fixed using the transformation matrix. As the relationship between the leaning degree of the mirror and the degree of distortion of the outcome can be calculated, it is possible to beforehand apply the converted image into transformation matrix. Then, the final image can be realized as expected.

$$R = \begin{bmatrix} 1\text{-}2a^2 & \text{-}2ab & \text{-}2ac & \text{-}2ad \\ \text{-}2ab & 1\text{-}2b^2 & \text{-}2bc & \text{-}2bd \\ \text{-}2ac & \text{-}2bc & 1\text{-}2c^2 & \text{-}2cd \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

The reflected world coordinates can be calculated by applying the transformation matrix (numerical formula 1) to the world coordinates. Through this method, therefore, the image distortion resulting from the leant mirror can be adjusted.

[1]Dept. Visual Contents, Graduate School of Dongseo University.
Email: {melc81, happyguy81}@gmail.com
[2]Dept. Digital Contents, Dongseo University.
Email: {tsyun, dhl}@dongseo.ac.kr

2 CONCLUSIONS

We proposed the interactive content system based on spatial augmented reality and multi touch screen. For multi touch screen, we made acrylic rear screen and detect touched point through IR LED and IR camera. In addition, we built SAR display space with half-silvered mirror and diorama. Half-silvered mirror were mixed virtual image with real diorama.

Figure 3. Single and Multi touch

Normally, we use only one finger for single touch. However at the special scene we use more than two fingers for multi touch, then we can see the special effect.

Figure 4. Diorama display and SAR display

Use the SAR display technique, as the result we got the two kind of background. One is the original diorama mountain, and the other is the image layered mountain like a perspective drawing. Our proposed system can be applied to many kind of interactive content system, such as education and advertisement.

•Acknowledgement.

This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation

REFERENCES

[1] O. Bimber, B. Fröhlich, D. Schmalstieg, and L.M. Encarnação, “The Virtual Showcase”, IEEE Computer Graphics & Applications, vol. 21, no.6, pp. 48-55, 2001.

[2] Jefferson Y. Han, "Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection" UIST'05, 2005, Seattle, Washington, USA, 2005.

Directed Emergent Drama vs. Pen & Paper Role-Playing Games

Maria Arinbjarnar and **Daniel Kudenko**[1]

Abstract. This is a position paper on the difference in role-playing between Computer-Based Role-Playing Games (CB-RPGs) and Pen & Paper Role-Playing Games (PP-RPGs) and how the architecture of the Directed Emergent Drama (DED) engine will facilitate more of the core characteristics of PP-RPGs than current CB-RPGs do.

1 Introduction

Current computer games fall short of providing a complete RPG experience for a number of reasons, the most prominent being that the storylines are pre-authored and do little to accommodate an independent Role-playing style. Role-playing refers to how the player can enter into a role, such as an evil wizard, and play that specific role through the game.

These are some of the core reasons for the creation of the Directed Emergent Drama (DED) engine. The DED architecture facilitates the guided emergence of an interactive drama using actor agents, a director and schemas. Schemas are generic structures used by the director to guide the actor in creating an engaging and immersing drama for the player. For a detailed description of DED see [1].

Due to lack of space we omit the technical details of DED and emphasise the differences between role playing in PP-RBGs and CB-RPGs and how the DED architecture will facilitate the core characteristics of PP-RBGs into future CB-RPGs.

It remains to be seen whether increasing the level of role-playing and emergent drama in CB-RPG will increase players enjoyment. This would need to be measured with empirical user studies.

2 Managing the Game

The Game Master(GM) in PP-RPG acts as a storyteller for the other players guiding them through an interactive drama, an adventure-world full of monsters and NPCs played by the GM. The GM is also the judge in all matters concerning game rules [4].

2.1 The Storyteller

"Pen-and-paper role-playing is live theatre and computer games are television." Gary Gygax, as quoted in [9]

In Neverwinter Nights [2] character alignment affects the players progress in the game by for instance hindering players playing druids in getting full access to their powers and the druid community if the player has not been careful to keep the character alignment as good. It will also affect what sub quests are available especially when it comes to quests that are for the advancement of a specific role and need a specific alignment. Still it does not change the main story or the main story plot, the player will need to finish specific main quests in order to advance and complete the main story. Nothing the player does will affect the pre-authored plots of the game.

Fallout 3 [6] has brought this idea even further, players actions greatly affect the characters reputation and opportunities. The availability of quests is directly linked to players actions; how much they explore the world, what they say in conversation, and what skills they pick when levelling up. Still the same can be said for Fallout 3 as with Neverwinter nights: nothing the player does will change the pre-authored plots or the main storyline. The player is always playing along one branch of a multi-linear story that essentially follows the same fundamental plot-points each time. This means that players will not get a novel story experience when replaying the game, which makes it very repetitive and lacking replayability.

in PP-RPGs the story emergence depends on player actions and the GMs skill in providing an engaging and rich environment. The core difference is that PP-RBGs are interactive dramas while CB-RPGs are incorporating a narrative into a game. The latter has a number of problems that Juul relates well in [5]. Essentially you can't have narration and interactivity at the same time any more than you can have panorama and drama at the same time. The two are a contradiction.

The DED architecture facilitates the emergence of interactive dramas using schemas. The director will constantly monitor the drama and deploy schemas as needed. Schemas structure the emergent drama by giving actors goals, knowledgebase and appropriate actions to choose from. The schemas are not small pre-authored stories like the vignettes in [8] or plot points in a plotgraph as in [7]. The schemas are generic structures used by the director to structure improvisational acting. This means that an actor receives goals to accomplish and relevant generic actions to use, the actions are further supported by a knowledge base which the actor can then use to determine what is an appropriate action each time with respect to the characters; emotion, situation and personality. This facilitates the emergence of a drama where a user can interact with the actors and story world freely and directly influence the unfolding drama.

2.2 The Actor

Another important role of the GM is that of playing all NPCs that the characters encounter. Currently the NPCs in CB-RPG are prescripted rather than autonomous and in Fallout 3 there is even a voice enactment for each line of text.

In the DED architecture the actors are autonomous and choose an action dependent on the situation each time like an improvisational actor would. The actors are not following a plotgraph. The director does not direct the actors, instead they receive guidance from schemas on what is appropriate with respect to the current status of

[1] The University of York, England, email: {maria,kudenko}@cs.york.ac.uk

the drama. This means that the actors use the actions and knowledge-base that they have received from schemas to form sentences and action sequences directly in response to the environment and interactions of other characters, the user, and the personality and emotions of the character they are playing.

2.3 The Judge

The DM should listen to the players and weigh their arguments fairly when disagreements arise, but the final decision belongs to the DM. The Dungeon Masters word is law! Gary Gygax [4]

Perhaps the only GM task that a computer can currently master is that of handling the game rules because of how exact it is in using calculations akin to that of the dices used in PP-RPG. Still it falls short even there, both because players can frequently use bugs to their advantage and cheat, and because it is unable to adapt the rules to allow for a more fluid story-emergence like a GM can.

A GM needs to reed volumes of books before being sufficiently familiar with the game rules to skilfully guide players through a game. Novice GMs sometimes fall into the beginners trap of being too strict on rules so that it becomes detrimental to game play [3]. The DED architecture shies away from strictly rule based scenario towards preserving the fluency and interaction that PP-RPGs offer. The DED architecture facilitates this by not having the same levelling up mechanism that is in RPGs, and no possibility of dying or loosing in the drama. Each drama that the user experiences will be scored and in addition to informing the users on how well they did the result will be used to offer dramas that best fits the users style of play, dramas that provide a challenge and entertainment for the users.

"The secret we should never let the gamemasters know is that they don't need any rules. " Gary Gygax, as quoted in [10]

3 Storytelling in PP-RPG's and DED

Quality GMs do not author a complete adventure for each game, rather they decide on certain key elements that are needed to start the adventurers on their way. These elements include key characters of the world and what their aims are were they hail from and their relationships. The GM will also decide on an inciting event to gain the adventurers interest.

The DED architecture facilitates the same using the DPGE to create a past for characters and their relationships and an inciting event for the interactive drama. It is important not to set a requirement that the player needs to enter a certain room or talk in a specific way to some NPC to progress a pre-authored story. Forcing the player down a specific narrative path is called railroading.

3.1 Railroading

"Sometimes G.M.s will write out an adventure's entire plot, soup to nuts. This is a dangerous thing, because it inherently contradicts one of the fundamentals of role-playing: the players should be allowed to determine the pace and direction of the adventure." Bill Coffin [3]

As discussed before this is the core difference between PP-RPGs and CB-RPG

"Making the players figure the story out for themselves is tantamount to having something hidden in your hand and forcing your friends to guess ad inflnitum what it is, even after their desire to guess has long since gone away." Bill Coffin[3]

This type of game play was very common in the early CB-RPGs, players needed to find that item or button hidden in one of the 80+ rooms of a castle or dungeon or the games story would not continue. In current games this has all but been removed. What remains though is the game's requirement of completing quests and talking to NPCs in the correct order. For instance in Neverwinter Nights the player needs to take care to find and speak to all quest givers before entering any dungeon, because killing that zombie or villain before having received that quest will result in not receiving experience points and rewards that go with it.

"In the end, having a solid plot line is not nearly as important as being able to make up a decent plot line on the fly." Bill Coffin[3]

Having the plot develop on the fly is exactly what the DED architecture facilitates, the director is constantly monitoring the success of each schema deployed and if schemas fail, for instance because the user does something that violates their constraints then that schema is revoked and the director searches for a new schema that better suits current status in the drama and the player and actors current actions and mental status. The director is responsible for having the drama follow a dramatic arc and to conform to some specific genre. The director does not try to guide the user in any way. Only the actors interact with the user.

4 Conclusions

It is the lack of a truly vicarious adventure that the DED architecture aims to fill by facilitating the emergence of a structured drama from player and actors' interactions. In this paper we shared how the DED architecture facilitates similar experiences as provided by PP-RPGs, due to parallels in the fundamental approach to interactive drama.

Increasing basic role-playing concepts in CB-RPGs is very likely to increase player satisfaction by appealing to a larger audience and by increasing the degree of engagement and immersion. This remains to be tested with empirical user studies.

5 Acknowledgements

We want to thank our anonymous reviewers and Game Master Florian Berger for their excellent suggestions.

REFERENCES

[1] M. Arinbjarnar and D. Kudenko, 'Schemas in directed emergent drama', in *proceedings of the 1^{st} Joint International Conference on Interactive Digital Storytelling ICIDS08*, Erfurt, Germany, (2008).
[2] Bioware. Newerwinter nights. http://nwn.bioware.com/, 2002.
[3] B. Coffin, *Rifts: Game Master Guide*, Palladium Books Inc, September 2001.
[4] G. Gygax, *Dungeon Module B2: The Keep on the Borderlands - Introductory Module for Character Levels 1-3*, Berkeley Top Line Distributing, 1980.
[5] J. Juul, 'Games telling stories?', *The International Journal of Computer Game Research*, **1**(1), (2001).
[6] Bethesda Softworks LLC. Fallout 3, 2008. http://secondlife.com/.
[7] B. Magerko, 'Story representation and interactive drama', in *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment conference (AIIDE)*, (2005).
[8] M. Riedl and N. Sugandh, 'Story planning with vignettes: Toward overcoming the content production bottleneck', in *proceedings of the 1^{st} Joint International Conference on Interactive Digital Storytelling ICIDS08*, Erfurt, Germany, (2008).
[9] S. Schiesel, 'Dungeon masters in cyberspace', *The New York Times*, (2006).
[10] A. Varney, 'Thoughts at non-random', *AMBER DICELESS ROLE-PLAYING*, (1992). Published as a sidebar to Lester Smith's review of Amber Diceless Roleplaying.