A MULTIPLAYER CASE BASED STORY ENGINE

Chris R. Fairclough and Pádraig Cunningham,
ML Group,
Computer science dept.,
Trinity College Dublin,
chris.fairclough@cs.tcd.ie, padraig.cunningham@cs.tcd.ie

KEYWORDS
Interactive story, multiplayer, case based planning, believable agents.

ABSTRACT

This paper describes the development of an expert case-based agent director system which dynamically generates and controls a story, which is played out in a multiplayer networked game world. The system handles multiple users in a game world and directs the non player characters therein to perform for the users parallel storylines, interweaving character roles in each story. The story is told through a 'narrative of actions' and automatically generated dialogue. Much of the storytelling approach is based on the seminal work of Vladimir Propp, to which is applied the AI case based planning paradigm. Initial analysis of the system is based on a review of the system and its output, but future work will involve developing a more objective format for analysis.

INTRODUCTION

The system described here is based on previous work described in (Fairclough 2002). The original implementation was limited to one player taking control of a hero in a simple hero/villain story structure, with no AI storytelling capability. The current system includes a story director (SD) system which utilises the case based planning (CBP) paradigm. The system also facilitates diverse multiplayer stories, as a number of different possible story structures are allowed for by the approach described in this paper.

The planning and scheduling of stories, modelled as cases in a case based planner, is the primary activity of the system. The game world is run and updated on a C++ server, and a C++ client logs on to the server to control a character in the game world. The game type is a 3D adventure game, where the player can use objects and interact with characters. The introduction of a multiplayer element significantly increases the workload of the story director agent. It must handle the current situations of all the players, dynamically identifying possible story cases and assigning story goals that are relevant to each player.

The structure of the rest of the paper is as follows. The second section, entitled '**Background'**, is an overview of the area of computer mediated storytelling, referencing previous work, and describing the genre of computer game that this project is aiming to facilitate. The third section, '**Design'**, details the overall system's design and shows how the game mechanics and story mechanics work. The fourth section, '**AI elements'**, concerns the advanced AI techniques utilised in the story generation subsystem. In the fifth section, an **analysis** of the system and its operation is made, and the paper ends with a description of **future work** and presentation of **conclusions**.

BACKGROUND

The structural analysis of stories has its earliest example in Aristotle's 'Poetics' (Aristotle), and his basic structural elements can still be recognised in popular stories today. Most modern developments, such as in cinema, concern new techniques for the telling of stories, and adapting stories for new media, rather than the restructuring of plot elements.

Previous Work

Propp (Propp 1968) is the progenitor of modern analysis, and his structuralist approach is appropriate for computer mediated plot-based storytelling, as it characterises the plot as a closed causal system, with the *character function* as the primary building block. This is the approach used in this project.

There are many modern approaches to story systemation, described in (Fairclough 2002), most notably the OZ project in CMU (OZ), and connected work such as Brenda Laurel's drama management system (Laurel 1991) and John Laird's group (Laird). The division of the problem into believable agent research and plot management research has been taken from the Oz project and incorporated into this work. This approach has correlations to Propp's work, as he asserts that a character function is independent of the character that performs it.

AI in Story Generation

The use of AI techniques in story generation has been around since the seventies, and emphasis has been placed on goal based agent technologies in NPCs (non player characters) (Rizzo et al 1999). The first example of this is found in Meehan's classic 'Talespin' (Meehan 1977). The use of AI in an 'omniscient' agent which does not inhabit the game world, which monitors and controls the NPCs, and whose activity is directing a storyline, is relatively new and found in some research projects (Mateas 99, Magerko 2002) but not, to the author's knowledge, in currently available commercial games.

MMORPGS

The massively multiplayer online role playing game (MMORPG) is a very new form of game, and differs from other multiplayer games, as the goal is not simply shooting other players, but interacting in a complex, changing environment with teams of other human players. This is an environment which necessitates an author constantly keeping track of the state of the game, and continuously writing a story that takes into account the wishes of the player community, and the dictates of the game world. This requires complex tools that are related to the structure of stories, and it is this need that is targeted by the system in this paper. Although the system is built around a specific game engine, continuation of this work will be centred on the creation of tools for game author/designers. The game that has been developed is a MMORPG type game, reduced in scale for play on a LAN.

DESIGN

The design work was done in conjunction with initial development, and the system was playtested, with results fed back into the design process.

System Architecture

The system is designed to allow multiple users to log onto a server and control a character avatar that inhabits the game world. The single player game was extended using Winsock and a client-server architecture.

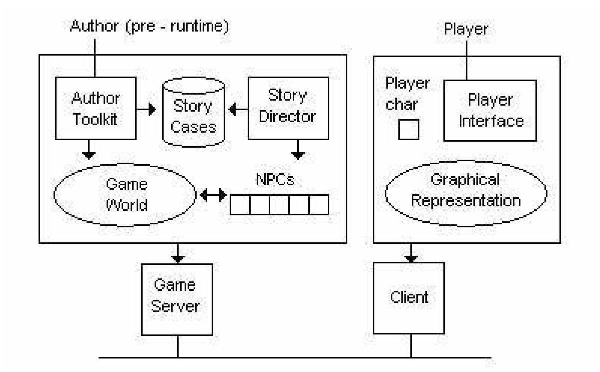

Figure 1: Overall System Architecture

The Server updates the client's view of the game world, taking into account the current location, within the world, of the client's player character. Character states and story progress is updated on the server, and the player interface receives commands, which update the client character, and update messages for this character are generated which are sent to the server.

Game Mechanics

The game is based on interactions between characters (NPCs and players), objects, and locations. The player has complete freedom of movement within the world, except certain story events can unlock 'portal' objects which enable transport between locations. Certain objects can also enable transport around the world. The NPCs are modelled in a layered structure, from low level behaviours to higher level targeted goals.

Character Modelling

- Low level - for example, collision detection which steers away from nearby objects and characters as they get too close.

- Social simulation - the NPCs use a basic gossip algorithm and inform other characters of events that have happened to them. They store the order they have met the other characters, and this ranking initially dictates who will be the target of their gossiping, and works as a general social rating that can be used by the Story Director (SD) agent in casting story goals.

- Idle behaviours - When an NPC does not have a goal to achieve, they can execute a behaviour such as patrolling around a house or following another NPC. These behaviours are pre-assigned by the author.

- Targeted behaviour - The SD agent assigns story goals to characters, and the characters search for the object of the goal and execute it. When targeted, a character will not execute idle behaviours.

- Attitudes - characters develop ratings for characters that interact with them, and characters that they hear about via the gossip algorithm. They remember the events that caused these rating changes, so they can gossip about them.

Objects in the game world come in different types, from background objects to 'action objects' that can be picked up by characters and enable specific actions. For example, a sword enables character A to injure character B which causes a negative change in B's rating for A. Movement around the world can also be achieved with the use of certain 'action objects', for example, a magic carpet. The game mechanics is primarily defined by the use of action objects, and a varied array of different objects has been defined so that each has different uses and effects.

A design goal of this project is to allow the progression of the story to be part of the gameplay, while allowing for a variation of gameplay types. For instance, a combat type game could have story based gameplay interspersed between fights, and the same story engine could be used in an adventure game where object puzzles separate the story elements.

Story Mechanics

The progression of the story must be influenced by the player's movements and actions in the world. To this end, the story director notes character attitudes to the player, which are changed every time a character interacts with the player, or hears about an interaction with the player. Characters are each capable of fulfilling a range of story functions. Some characters are defined as being primary and possess more abilities and influence, and are more likely to be assigned story goals.

The story is conveyed through characters performing actions and delivering lines to the player. The dialogue is dynamically generated by stringing character names, verbs, and object names together. Witnessed actions are noted and if the player needs to be informed of a certain event, a witness to that event will be assigned the goal of informing them.

A character that is assigned a story goal takes one of seven roles, as defined by Propp; the hero, the villain, the mediator, the donor, the helper, the false hero, and the princess. The characters in these roles can be initially defined by the author, yet are dynamically reassigned by the story director agent during the game, according to the development of the character's relationships.

AI ELEMENTS

The term ‘AI’ in games usually refers to the character behaviour algorithms, but in this system, the AI elements reside for the most part in the story director agent, and the characters use standard techniques found in plenty of games available today, thus are not addressed in this section.

Expert Knowledge

The expert knowledge represented is that from Propp’s work, and consists of a rule based system which works in tandem with the case based system. For each of the 31 character functions that Propp defined (Propp 68), a rule exists that roughly defines how that function will be assigned. Further definition is done as the function passes through the ‘assembly line’ of the SD. The character functions enumerated include Villainy, Guidance, Testing of the hero, and Receipt of a magical (useful) object. The Interdiction function, for example, is when the hero receives (I) an order to do something, or (II) a warning not to do something (see screenshot). This function, like a number of others, is paired with another function, in this case Violation.

Figure 2: An Interdiction Function in the Star Wars Demo Game

The rules consider whether to accept decisions made by role casting, or re-cast a certain story goal. They assign the current goal based on the casting system’s roles (see next section). The rule based system effectively draws the decisions made by the other components together, before the characters can act them out. An example using pseudo-code:

```
If (currentGoal.functiontype = Villainny )
And (currentmove.currentVillain.suitability > threshold )
Assign ( currentGoal TO currentmove.currentVillain )
```

Case-based Planning of Stories

Case based reasoning is a popular AI field which aims to solve problems by extrapolating from past, solved, problem situations to find solutions for new problems, adapting them, as needed, to the dictates of the current situation. The k-nearest neighbour algorithm is used to find cases that are similar to the input case. This approach has been identified as suitable for constructing story scripts from a base of authored scripts. An introduction to the field of CBR is found in (Cunningham 1998).

Each case in the case base represents one *move* of a story, and consists of a script that is interpreted and assigned by the SD. ‘Move’ is defined by Propp. A script line can contain very abstract instructions, such as a one word ‘Villainy’ instruction. This will trigger the SD to finding the character with the villain role, (or a character close to the villain as regards the social simulation) and instruct it to perform an act of villainy on the hero or a character close to the hero, such as a murder. More specific instructions, such as assigning the mediator character the goal of notifying the hero of a villainy event can also be scripted. The cases are converted to a goal stack by the SD to be assigned to the NPCs. Below is an example, from Propp, of a two move story, converted to script form. Each move is represented as a case in the case based planner.

No. 95 in Appendix 3 of (Propp68):

<u>Move I</u>
villainy (expel) ;
mediation (transport) ;
donor (test) ;
reaction ;
provision (wanted) ;
return ;
<u>Move II</u>
lack ;
mediation (transport) ;
counteraction ;
departure ;
donor (test) ;
reaction ;
provision ;
return ;

There are 80 cases in the case base, from the 44 multi-move story scripts given by Propp. In the k-nearest neighbour part of the algorithm, the cases are compared to the current state of the story world to find the best fit for a case from which to select the next story goal. The comparison is based on an analysis of the character and object resources needed to execute a story script, and also on the story functions performed so far. In the story that the example above is abstracted from, move II directly follows move I, and the link between cases that follow each other is preserved in the CBP system. Thus, move II would be more likely to be executed after move I has completed than other candidate cases. Cases similar to move II would also be considered. The cases in this system are best described as story templates, whose details are filled in by the SD AI. This 'filling in' is done using techniques similar to constraint satisfaction, as described below.

Casting

Casting roles to characters, and finding objects that can help fulfil goals is a task that is implemented using constraint satisfaction techniques. The trend of modern successful games is towards large, complex, simulated game worlds containing many objects and characters. In this environment, a consistent storyline must be rigidly pre-scripted, or else the story must be dynamically generated, with a mechanism for identifying the appropriate object or character, for a given story goal, provided. To find the right one to achieve a certain goal, the properties of each must be searched to fit the constraints dictated by the current goal. In this game engine, characters and objects have properties which are matched to the dictates of the story function being carried out.

Each story function (of the 31) has a set of constraints for a character to satisfy, and characters have social ratings, possessed objects, loyalties and a current location that are matched as well as possible to the constraints. The character that matches the constraints the most will be assigned the goal.

Objects also have different types, and a character can be assigned an intermediate goal to find the right one, in order to perform a certain goal. For example, a sword would be used to injure a character over a magic carpet, and if a character needs to acquire the object, a goal to pick it up or be given it is added to the goal stack for that move.

Multiple Parallel Storylines

For a multiplayer environment, the SD handles the story from multiple viewpoints. Each player's client sends messages to a server-side connection manager that notes where that player is and what they are doing. The main SD instantiates a new story director for each one, each looking after most of the planning and casting relevant to each player. Two players could both act as hero characters for two separate hero stories using the same or intersecting set of NPCs. In this case, there is two SDs maintaining a set of roles and list of story functions for each player's hero story. Alternatively, one player could be the villain or the false hero in the same story as the first player's hero. Ideally the SD could dynamically recognise the roles that each player wishes to adopt, based on their playing, but a set of options could also be presented to the player as they join the game world, to find out what sort of character they want to portray. The latter approach is being developed for use in an evaluation version of the multiplayer game.

Even in the single player game, it has been found that multiple parallel storylines need to be executed, due to the nature of a non-linear, free-roaming type of game such as this. A player may leave the locale of a story case being executed, to enter an area or situation where a different case would be more suitable. To account for this, the SD can plan and assign up to three different story cases at once per player. This number was chosen because a greater number could mean the player could become confused and the individual stories could lose cohesion (also the historical significance of the number three, both in computer games and traditional stories).

ANALYSIS

The primary task in analysing the usefulness of this work consists of evaluating how the quality of the stories generated by it is maintained, while allowing interactivity. The structures described by Propp are derived from his study of stories, and the stories generated by this system are generated from those structures. The validity of the whole approach relies on Propp's assertion that these structural features are the primary classifying features of stories. This being true, the structures should form a good basis for the generation of new stories. Initial playtesting is promising, but a more focused approach, with a number of playtesters and a rigorous test scheme, is in development with more objective results in mind.

Story Structure

Although it is technically limited in scope to folktales, many authors have noted Propp's structural analysis's remarkable flexibility in its applicability to popular stories. In cinema, TV, and most computer games, a large number of stories can be fitted into a recognizable structure, hence the predictability of a lot of that material. The hero/villain interplay is enriched with the other five character roles in Propp's analysis, yet it remains a simple form. Literature and higher forms of storytelling have far more complicated structures that are best left to artists, and are not addressed in this work.

An emphasis in both Propp and Aristotle's work is that of the conception of the tale as a *causal* system. Ideally, every event that happens in the story has either an effect on some other story event, or is an effect of one. The system described above allows for this by abstracting the social, and action-object based game events from their context, effecting attitudes each character has for the others. These ratings, along with the list of preceding story events, are fed back into the SD agent's deliberations on generating the next story goal.

Completed Demo games

Two demo games have been developed to date, the first a simple original environment entitled 'Bonji and the magic peanut'. The other is based on the first half of 'Star Wars – A new hope', TM Lucasfilm, and features the familiar characters in a more variable form of the traditional 'rescue the princess' idiom. The characters are allowed some autonomy and Luke is channelled through the story not by limiting his location, but by giving story goals to nearby characters, allowing variations, such as a Jawa informing Luke of the attack of the stormtroopers instead of Obi Wan.

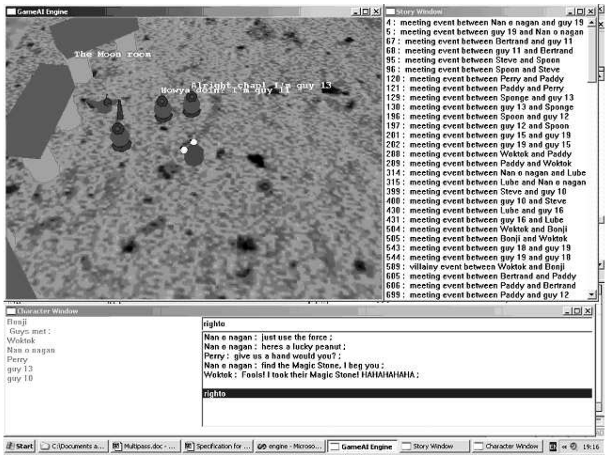

Figure 3: 'Bonji and the Magic Peanut' with Story Progress Window and Character Status & Control Window

Advantages

There are three main advantages of using a system such as this in a game engine. Firstly, the ability to choose different paths through the same story, with different characters displaying their abilities in achieving the same story goal, is a powerful means to increase replay value of a game.

Secondly, allowing the player to influence the progress of the story, so a different story structure is brought into the game according to player choices is enabled by using case based story planning. The range of cases available can be limited for each game world, allowing the author a higher level control of story events.

Thirdly, with a multiplayer option, and because the story generation is real-time, players can interact in more complex ways, playing with NPCs' loyalties to gain the upper hand over each other. This will allow for adventure-game type stories that remain consistent as players meet other players. A common element in RPGs is a team of characters under player control, and in multiplayer versions, players can meet up with others to form a team. Many online MMORPG developers have had problems in implementing a satisfying storyline in this type of environment, as each player may take different paths through the game, teaming up when they see fit.

FUTURE WORK

The next version of the demo game will incorporate the multiplayer option, as well as a wider range of abilities in the player interface, facilitated by a more interesting set of action objects.

Two future additions to the story engine are:

Learning – In case based systems, the 'retain' step (Cunningham 1998) consists of storing the new solution that has been adapted from the old case, or the combination of multiple old cases. This facilitates learning from experience, and for it to work in the story engine, an algorithm for properly (pun intended) combining two or more cases is needed. The SD would then be able to generate and learn new cases and find good fits for new situations more easily.

Deception – When a NPC uses deception, they inform the player of events that did not happen in the game world. The fabricated events should be calculated to induce a certain reaction in the player. For instance, to make the player think NPC1 is on the side of the villain, NPC2 could tell the player that NPC1 did an act of villainy on a character that is close to the player. Propp's role of 'false hero' has close links with deception and relates to the hero uncovering the deception of one who takes responsibility for the acts of the hero. The functions related to the false hero form part of the conclusion of the tale, according to Propp. However, deception could also be used in other parts of the tale.

A possible addition to the story engine is:

'Harmonies' - There are sometimes short tangential storylines found in many stories, used for introducing main characters and for other reasons. These often allude to the main plot in their similarity to it, and they could be seen as harmonies over the main plot structure. They could be implemented by taking the current story case, and executing it in the context of the required function.

The system, as is, is not scaleable for use in an abstract large scale game world, but an important aim of the project is to make it so, in order to facilitate the creation of tools for the incorporation of the story director system into other games. The architecture is fully scaleable, as it is designed with the complexity of a large-scale simulated world in mind.

CONCLUSIONS

Currently, analysis of the system is based on a review of the system and its output. Although this is not good enough for a rigorous analysis, there are no readily available scientific criteria for analysing the quality of a story. In the interest of a more objective analysis, the networked system will be combined with a separate user interface for getting feedback on the users' experiences with the game, based on story criticism criteria. The result of this will include an analysis of believability, consistency, drama, and the level of user interactivity.

The system described here is unique in its combination of AI techniques, software architecture, and game style. However, there are other systems which take into account emotional modelling, cinematography, and others of the multitude of elements which make up compelling storytelling. Magerko's work (Magerko 2002) is similar in approach to ours, and this is encouraging as it seems to be a useful paradigm for the problem. There are other, varied approaches to computer mediated storytelling, and this is an indication of the richness of the subject matter. Stories can be told in many ways and human imagination will always be the best way to create them, but systems such as these can provide advanced tools for the creation of the next generation of story vehicles.

REFERENCES

Aristotle. *Poetics*

Cunningham, P. 1998. "CBR: Strengths and Weaknesses"

Fairclough, C. and Cunningham, P. 2002. "An interactive Story Engine", AICS 2002, LNAI 2464, p 171-176

Kahn, M. 1979. "Creation of Computer Animation from story descriptions." PhD thesis, AI lab, MIT

Laird, J. and van Lent, M. 2000. "Human level AI's killer application - computer games" AAAI National Conference on Artificial Intelligence

Laurel, B. 1991. *Computers as Theatre*. Addison-Wesley

Magerko, B. 2002. "A Proposal for an Interactive Drama Architecture", AAAI Spring Symposium on interactive entertainment.

Mateas, M. and Sengers, P. 1999 "Narrative Intelligence" *Working notes of the Narrative Intelligence Symposium* , AAAI Fall Symposium Series. Menlo Park, CA: AAAI Press.

Mateas, M., Stern, A 2000. "Towards Integrating Plot and Character for Interactive Drama." *Working notes of the Social Intelligent Agents: The Human in the Loop Symposium*. AAAI Fall Symposium Series.

Meehan, J. 1977. *The Metanovel: writing stories on computer*. University microfilms international.

OZ. The OZ project at Carnegie Mellon University, Pittsburgh .http://www.cs.cmu.edu/afs/cs.cmu.edu/project/oz/web/

Propp, V. 1968. *Morphology of the Folktale*. University of Texas Press

Reilly, W. S. and Bates, J. 1992. "Building Emotional Agents" CMU-CS-92-143

Rizzo, P., Veloso, M.V., Miceli, M. and Cesta, A. 1999 "Goal-based Personalities and Social Behaviours in Believable Agents." *Applied Artificial Intelligence*, 13:239–272

Smith, S. and Bates, J. 1989. "Towards a Theory of Narrative for Interactive Fiction", CMU-CS

'Star Wars' TM Lucasfilm