AI STRUCTURALIST STORYTELLING IN COMPUTER GAMES

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, case-based planning, computer games.

ABSTRACT

This paper is a description of our work in creating a story director agent which utilises AI techniques. The story director controls the storyline in an adventure computer game, with the player controlling the hero character, and the story director reacting to the player's actions. The story is told through subplot-level plans being formulated with a case-based planner, and a social simulation system that the story director is 'plugged in to', allowing consistent logical stories while allowing for player freedom. The system has been named OPIATE – Open-ended Proppian Interactive Adaptive Tale Engine.

INTRODUCTION

This paper follows (Fairclough & Cunningham 2002), and (Fairclough & Cunningham 2003); the former proposes the system, and the latter describes its development for use in multi-player games. After some background, this paper describes in detail the AI algorithms used in the system, the limitations of the approach, and possible future improvements.

BACKGROUND

Computer games are currently going through a number of contradictory trends. There is a new outcrop of mobile and internet-based games that emphasise short bursts of fun that are used for advertising, promoting websites and services, and even political messages. On the other hand, games that people invest more time in, such as adventure PC and console games, are becoming larger-scale, more complex affairs. This schism is serving to generate a wide range of new genres that borrow game concepts from each other, and from the older genres, through the short evolution of the computer game.

Genres that emphasise story and adventure are very popular, with 'Spiderman 2' currently topping charts around the globe. The current successful model for storytelling in games, popularised by GTA3, but initiated with Mario64, is to have a series of 'story missions' that advance the plot, with a selection of optional missions that enable a feeling of freedom of choice in a player. The variability of this model is based on the character abilities that the player has, so each game seems different, while this basic gameplay model is common to a lot of current games. This paper proposes a possible next step for this storytelling model, abolishing the more traditional pre-scripted main plot for a more open-ended, procedural, view of stories themselves. This approach has been developed based on previous work in the fields of structuralist analysis, and was inspired by such contemporary practitioners as Chris Crawford(Crawford 2002), Michael Mateas(Mateas 1999), Nicholas Szilas(Szilas 1999), Nikitas Sgouros(Sgouros 1999), Norbert Braun(Braun & Grasbon 2001), and many others.

Some Previous Work

AI in storytelling was first concerned with story generation as text. In the seventies, Meehan's Talespin(Meehan 1977) generated much interest as a simple computer storyteller that utilised character-level planning. Later, Turner's Minstrel(Turner 1992) expanded on this to include author-level goals in a case-based planner. Turner's biggest success was in formulating a complete set of rules and paradigms for author and character-level planning using what he called 'imaginative memory', and analogical reasoning, to generate novel situations and plans for the characters and author model.

Storytelling for computer games has always been faced with the problems that occur when a player is given choices that could affect the plot. For real-time story generation, it has been assumed that these problems can lead to combinatorial explosions in complexity for a computer story teller, yet our approach demonstrates that this is avoidable. Since the OZ project in CMU(Smith & Bates 1989) began, more and more interest in AI real-time storytelling has surfaced, although it is not a technology that has been in much use commercially, although Braun's work on the Geist project(Braun 2002) has been used in a tourist attraction, utilising augmented reality headsets for display of characters.

Every system that has been developed is necessarily focused on a particular genre of story. Mateas focuses on a small location with only three characters, for a drama-intensive story experience. Crawford focuses on using a large number of interaction types (verbs) with short story segments that can relate to each other. He also provides an author tool-kit that enables creation of new storyworlds using his technology, but this does not allow for the emotional expressiveness of Mateas's approach, and is notoriously difficult to use. Nevertheless, his work has shown some of the possibilities and promise of interactive stories.

The challenge of creating a mechanism, whereby a player is both engrossed in a story and immersed in a world, is one that has been steadily overcome over the course of the evolution of computer games. Simulation techniques, such as cellular automata, can enable a greater feeling of involvement and freedom in a living world, but the traditional concept of a story is incompatible with a world like this. Players are seen to create their own stories from their experiences in the world, as has been observed in 'The Sims'.

However, a story is not merely a series of causal events. Stories have their own innate structures and processes, independent of the characters they portray. This was asserted in the 19th century by Adolf Bastian(Koepping 1984), and emphasised by Vladimir Propp(Propp 1982) and Claude Levi-Strauss as structuralist theory was developed. To enable a simulation-level model of a story, these common structures of stories must be simulated using rules of dynamics based on the structuralist theory, and they must be active in the interactions of the storyworld. A believable gameworld can thus be augmented to create events that fit into the rules of world dynamics, but that also fit into a suitable story structure.

The story structures that we have elected to use are those of Vladimir Propp, who analysed Russian Folktales in 1928 and came up with an extremely empirical methodology for classifying his corpus. The applicability of folktale analysis to computer game storytelling is compelling, as the nature of folktales is ever-changing, allowing for an analysis that extrapolates the nonvariant elements of the tales. This has enabled the discovery of skeletal structures that can be fleshed out differently for each storyworld.

DESIGN

This section will deal with the storytelling architecture and detail the AI methodology that was used in the OPIATE system. The game architecture is detailed in our previous work, but consists of a 3D adventure game, with characters, objects, and locations being the most important components. The characters use a layered architecture and feature collision detection, idle behaviours, social simulation, attitudes, and goal-directed behaviours. The use of objects is how characters perform interactions, which generate events. The game engine handles some game events, but the independent story director agent initiates most events by being aware of the storyworld and giving relevant goals to the NPCs (non-player-characters). The most important element of the gameworld is the gossip system, which provides a dynamic social simulation where knowledge of game events is disseminated throughout the characters. This allows the player to effect the unfolding of the story, as the story director bases its decisions on these character dynamics. See (Fairclough & Cunningham 2003) for a more detailed description of the current testbed game engine.

Story Modelling

Stories are modelled as an interplay between autonomous character actions and story director- initiated story actions. The autonomous character actions occur as a result of a social simulation system, whereby each character builds up a set of attitudes for other characters, based on a memory of the actions that have happened directly concerning them, and actions that they have heard about or witnessed. Characters have a gossiping system, which propagates information about game events through the cast of characters.

The story director agent queries the game world about character attitudes and locations, and player feedback, and bases planning decisions on this information. The plans it creates are sequences of character actions, each of which can be enacted by any character that fulfills the criteria for that action. These are equivalent to Propp's 'character functions', defined as 'an act of character, defined from the point of view of its importance to the course of action'. The system has a case library of plans that were authored based on the corpus of Propp's analysis in (Propp 1968). This case library encodes the expert knowledge that does not represent Propp's expertise, or any one expert's knowledge, but the expert knowledge encoded in the folk tales themselves, concerning the skeletal structures that define the different types of stories Propp analysed.

Case-based planning encodes knowledge as a library of cases, and deals with new problems through the mechanisms of recalling previous similar cases, adapting them for reuse, and assessing and storing the resulting new solution. Thus, a learning, adaptive system can efficiently solve problems similar to old ones. The story director(SD) in OPIATE uses the scheme shown below (Figure 1) to plan and cast story goals to characters. Each component of this process will be detailed in the following sections.

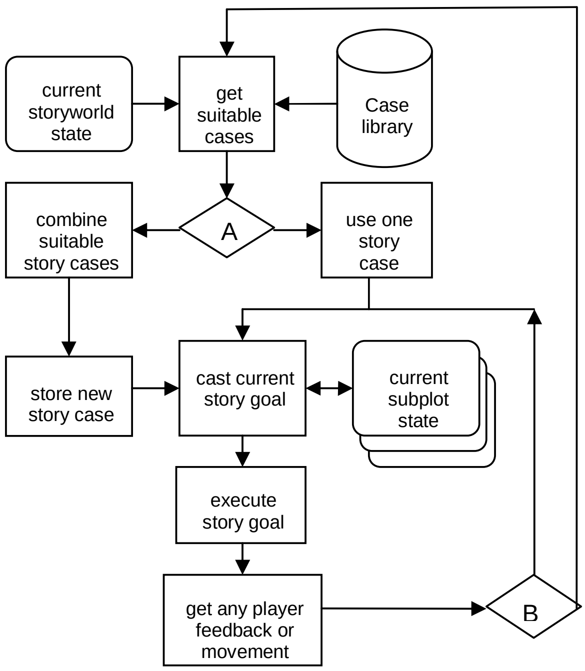

Figure 1. A flowchart showing the planning process

Suitability of Sub-Plots

The case based planning system uses a k-nearest neighbour algorithm to find suitable cases based on the heuristic shown below (Equation 1). The heuristic can be termed a *suitability* metric, instead of the normal similarity metric used in case based systems. It finds the most suitable sub-plot to be enacted given the current state of the characters and storyworld, taking into account attitudes of the characters to each other, and to the player character. The core features that are used in this metric concern *roles* and *actions*. Roles are occupied by characters when they are enacting story functions, and the relevance of a character to a certain role is calculated based on past and present attitudes and memories concerning the player/hero character, and the other characters. Actions are enabled by *actionObjects* that occupy the storyworld, and allow characters to perform distinct types of interactions. They can all be picked up, given to other characters, and gossiped about.

Eqn 1.: $$Sn = \left(\sum_{i=1}^{Ln} (Wr * Sri) + (Wa * Sai) \right) / Ln$$

Where *Sn* is the suitability of case *n*, *Ln* is the length in functions of case *n*, *Wr* and *Wa* are the relative weights attached to roles and actions, and *Sri* and *Sai* are the suitabilities of the roles and action(s) present in function *i*. *Sri* is given by Eqn 2, and *Sai* is given by Eqn 3.

Eqn 2.: $$Sri = \sum_{j=1}^{\#C} Raj$$

Where *#C* is the number of characters currently available to the SD, and *Raj* is the relevance of character j to the role given by function *i*.

Eqn. 3.: $$Sai = \sum_{k=1}^{\#A} Rak$$

Where *#A* is the number of actions currently available to the characters, and *Rak* is the relevance of those actions to the actions required by function *i*. The relevance values are binary, as an action object either fulfils the action given in function *i*, or it doesn't.

Case Combination

Once an ordered list of suitable cases is found, using the quicksort algorithm, a decision is made to use the most suitable case (decision diamond A in Fig. 1), or combine cases to get a new one. If a hardcoded suitability threshold is reached, the former choice will occur, but if a combination of cases gets a better suitability, the latter will occur. Combination of cases is done on a per-function basis. As each function has its own suitability rating, the most suitable can be interchanged with less suitable functions in the target case. This is done by taking the most suitable case and replacing its less suitable functions with equivalent, but better scoring ones from the second or third ranking cases.

An important element in combining cases is to maintain integrity of the structures when they are transferred, so Propp's *groupings* of functions are used to facilitate this. If a function is selected for transfer, and it has associated functions from the source case, these are also transferred to the target case. This can entail replacement of target case functions, so when the new case is constructed, it is reassessed for suitability. The groups are only of two or three functions, so this is not a difficult operation.

Casting

Once a suitable subplot plan is selected, it must be converted from a list of abstract story functions into a series of events in the gameworld, interpretable by the player as a storyline. To this end, the story director uses a casting system which dynamically casts the game characters into eight of the nine possible roles. Propp defined the seven roles: Hero, Villain, Mediator, Donor, Helper, False Hero, and Princess, and these have been augmented with two roles that he mentions, yet in his schema fall into the other categories. These are the roles of Family, and King. The hero character is always occupied by the player, even if they don't act particularly heroic. The usefulness of Propp's schema would be reduced if this was not the case.

These roles are cast as needed by a subplot. I will mention here that the term 'subplot' has been used in this paper where in Propp's work and our own previous papers, the term 'move' is used. This is to aid readability, as the general understanding of 'subplot' is roughly equivalent to the sense of Propp's 'move'. The roles required of the current subplot are dynamically cast as the subplot is being enacted, so that for example, a character can take the role of a Donor, and later can be the False Hero if the player/hero character falls out of favour with that character.

Casting is done using a set of criteria for each role. The villain role is filled by the character that opposes the hero the most, or else is a character close to that character. Opposition to the hero can come out of an attitude developed from author-defined backstory, or from events that occur in the course of the game. In this way, acts of villainy can be carried out by 'henchmen', depending on availability of characters. A Mediator can be any character that is available and nearby, even if the character is antagonistic to the hero. The Donor role can be filled by an available character that has not met the hero or has a slightly positive attitude. The Helper is filled by a previously met character that is fulfilling a positive previous encounter. The False Hero character must be a character with a previous positive attitude to the hero, who has either developed a negative attitude, or else has developed a positive attitude to the villain. The princess role is one that a character close to the hero can occupy, or a character that has not met the hero, but has been pre-authored as a possible princess character. The characters with positive attitudes to the hero can all take the Family role, and the King is taken by a powerful character, that a large number of characters have positive attitudes to.

The specificity of these roles and rules was formulated using a familiarity with Propp's work and its applicability to the game that has been developed, yet they could be editable through a toolkit if this system were to be used for other games. The rules are not arbitrary, and have been designed to maximise a sense of believability of the characters in their enactment of subplots.

Once a character is selected for a given function, the means of carrying it out is selected through a search of all actions available to the character. A character can be given a sub-goal to find and pick up the object, or it can be given by another character. The enactment of the story function consists of finding the target of the function, animation of the actionobject, and the generation of suitable text for dialogue. The dialogue is generated with simple verb-noun structures, with characters capable of talking about characters, objects, events, and attitudes in a simple manner. Descriptive or emotional text is not used, and syntax is kept extremely simple. Despite this, a story can be seen to emerge based on the simple dialogue.

Because the system presents stories with animation, and is less dialogue-based, it is not suitable to present the output of the system here as a listing of dialogue. However, a presentation and some video files that illustrate the output of the system are available at www.cs.tcd.ie/faircloc/.

Player Feedback and Numerous Subplots

The player can elect to do what is asked of him in certain functions, e.g. the Donor function where a character tests the hero's worth with some challenge or request, or can ignore the request, whereby a recasting of the Donor goal is done. If the hero ventures into a new area, with new actions and characters available, or if new elements enter his current situation as a result of the simulation, an entirely new subplot can be selected for enactment (decision diamond B in Fig. 1). If this happens, the old subplot is not forgotten, but can come back into play if it is found to be again suitable for enactment. A set of active subplots is maintained, and the player can choose which ones to follow.These are the chief mechanisms which allow for player freedom of choice in the game, yet because the whole case-based planning mechanism works from data that is directly alterable by the player, the plot can also be directly influenced by player action in this way. For instance, if the player is 'liked' by a character, but performs some action that alters that characters attitudes towards them negatively, a plot with the Falsehero role would be more likely to be selected.

LIMITATIONS AND IMPROVEMENTS

The OPIATE system is limited by the home-grown game engine that is the current testbed. A more believable game world would help in evaluating the system's usefulness. It was decided not to use an available commercial game engine, such as the 'Unreal' engine, due to the need for flexibility and the required presence of the omniscient story director agent. The game has been developed to the point where a player testing scheme is possible, for a more objective analysis of the 'storyness' of the game experience. This is necessary for evaluation of the system, as the assumptions that were made in building the system are in question. Some of these assumptions are:

- That Propp's classification system is correct and shows structures that are actually present in tales.
- That a story can be 'reverse engineered' using these structures, and incorporated into a set of character dynamics.
- That sophisticated dialogue is not necessary to convey a story, but is used to enhance its quality.

The last assumption indicates an improvement that could be made to the system by incorporating a more advanced natural language module into the characters, which can be customised to each character. This would involve an author defining 'turns of phrase', colloquialisms, and typical adjectives that a character uses. This could form part of the social simulation, where characters can grow to use the pet phrases of the characters closest to them, and serve as an implicit indication of social connections.

An important component of creating a story in this system is the authoring of the game world and the interactions present in it. This is the chief method of authoring the high-level 'flow' of the story, defining the paths of movement and interaction through the game environment. By placing certain actionobjects in key locations or with certain characters, an author dictates the sort of interactions that will occur in certain stages of the game. An author also defines a backstory for the characters by giving them attitudes and memories of events. These attitudes and events are equivalent to the in-game ones, and effect the character dynamics in the same way.

The gameplay in our demonstration game is quite limited, and poses no real challenge to a player. If the player chooses to ignore a puzzle or challenge, then they can simply pursue a different subplot. However, there are only a limited number of puzzles authored, and no real possibility for emergent puzzles in the game. Puzzles are authored as specific problems a character wants solved, or the required use of a certain object to progress. As there are not a large number of locations (22), the player's options are quite limited. There are 28 characters and 18 types of actionobjects, however, so the player's choices primarily lie in their interactions with other characters. Characters can develop desires for certain types of object as a result of a subplot requiring it, but these are not authored puzzles, and play out somewhat artificially, as the desire is not based on any internal drives on the part of the character model. In a more large-scale, fully simulated world, the OPIATE system should perform better, with characters' problems and desires emerging from more fully realised character simulations.

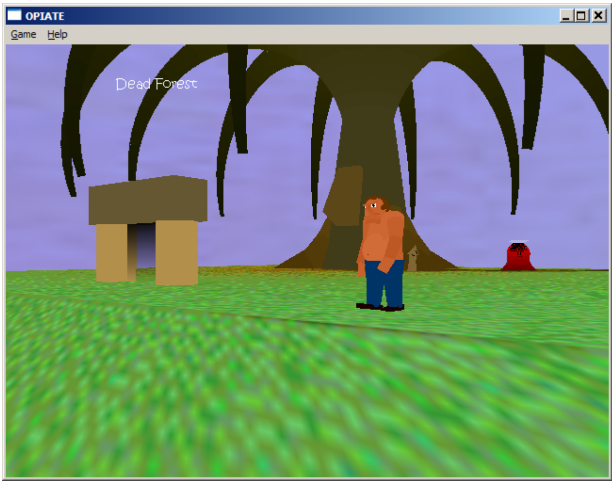

Figure 2: A screenshot of the demo game.

The OPIATE demonstration game is called 'Bonji's Adventures in Calabria' (Figure 2) and features three distinct 'locales' with about 8 locations in each. Progress from locale to locale entails solving a number of puzzles that are incorporated into the dynamic storytelling, with characters taking different roles depending on the player's interactions. However, the variability of the stories is limited by the initial setup of the storyworld, so the first subplot to be selected is always the same. Once a locale has been reached, the player can go back through the locales, revisiting characters and prompting new subplots to be selected based on the history of interactions.

Non-player characters can move around locales, but not between them. However, every time a subplot is successfully concluded, and as long as the subplot finishes with the 'Wedding' character function, a new character is available for player control. Propp's Wedding function is used as the hero's final reward at the end of folktales, and the decision was made to use this function to concurrently reward the player, by allowing them to control a new character. When a new character is selected, the previous hero character behaves like all the other NPCs, gossiping about attitudes and events, and is available for story goal enactment, consistent with previous interactions. This character can then be re-selected for use at any time. It would be theoretically possible for the player to gain control over every character in the game.

Overall, the system has turned out to be a success, blending ideas from a number of different projects to achieve an attractive option for a storytelling paradigm in computer games. The approach is experimental and not fully realised yet, but could help in developing more flexible story experiences for players. Future work on the system could help in its applicability to other game engines. This will neccesitate the building of an author toolkit to allow for greater author control of the processes that the SD uses to direct the story, and a plot script editor for designing new subplots and new types of character function, outside the ones Propp defined which were used in this work. One serious limitation of the approach is that it does not seem to be incompatible with the current trend of pre-recorded speech in games. However, some games, notably ICO by SONY, manage to tell a story with almost no dialogue at all, so the more action-based storytelling approach of OPIATE could be useful in this type of game.

REFERENCES

Braun, N. 2002, "Narrative Semiotics for Computer Games" Challenge of Computer Games, Lodz.

Crawford, C. 2002, *The Art of Interactive Design* No Starch Press

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

Fairclough, C. and Cunningham, P. 2003. “A Multiplayer Case-Based Story Engine”, *Proceedings of the 4th International conference on Intelligent games and Simulation*. Eurosis. pp 41-46

Grasbon, D. & Braun, N. 2001 "A Morphological Approach to Interactive Storytelling" Proceedings of the Conference on artistic, cultural and scientific aspects of experimental media spaces.

Koepping, K. P. 1983 *Adolf Bastian and the Psychic Unity of Mankind* University of Queensland Press

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

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.

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

Propp, V. 1984 *Theory and History of Folklore* University of Minnesota Press

Sgouros, N. M. 1999, "Dynamic Generation, Management and Resolution of Interactive Plots" Artificial Intelligence, vol. 107, no.1, pp 29-62, Elsevier Science

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

Szilas, N. 1999, "Interactive Drama on Computer: Beyond Linear Narrative" *Working notes of the Narrative Intelligence Symposium* AAAI Fall Symposium Series. Menlo Park, CA: AAAI Press.

Turner, S. R. 1992, "Minstrel: A Computer Model of Creativity and Storytelling" University of California Tech Report CSD920057