

Transferring Game Mastering Laws to Interactive Digital Storytelling

Federico Peinado and Pablo Gervás

Dep. Sistemas Informáticos y Programación
Universidad Complutense de Madrid, Spain
fpeinado@fdi.ucm.es pgervas@sip.ucm.es
<http://gaia.sip.ucm.es>

Abstract. The Interactive Dilemma is the inevitable conflict between author's determinism and interactor's freedom. There are some approaches that try to tackle it, using strategies and heuristic rules that can combine on the fly the previously designed author material with the run-time decisions of the interactor. Interactive Narrative is a relatively new field and it is difficult to find formal studies that shows how to create this kind of art. Our proposal is based on the theoretical study of tabletop Role-Playing Games and it involves the practical implementation of those ideas for managing the interaction in a simple text adventure game. Game Masters are the best models we have found in real life for designing and directing interactive stories. In this paper we transfer their player modeling, their rules for interaction management and their improvising algorithms from the real world to a new Interactive Storytelling system.

1 Introduction

The main problem of Interactive Digital Storytelling (IDS) design is structuring an intensive and meaningful interactive experience at the same time as enabling the development of a good pre-authored plot. This is a real dilemma, because while the interactor is taking a lot of decisions about the performance of his character that change the plot development, the author's plan may have been developed according to a different storyline.

Many approaches are found in the literature that try to solve this conflict in an automated or semi-automated way. Basically, they make interactive storylines by adapting author's plot units or other pre-established resources to the interactor behaviour at run-time. This task requires a computational solution that can react appropriately to "unexpected" user behavior.

Because Interactive Narrative is a relatively new field and it is difficult to find formal studies about interactive plot development, we have done a review over the pen-and-paper methodology of Role-Playing Games (RPGs) and now we are coding a practical implementation of these ideas in a textual game.

2 The Main Pillars of Our Approach

Our approach is based on developing a Case-Based Reasoning (CBR) model of a particular set of algorithms and heuristics for role-playing game mastering, and applying this model to drive the interactions in a multiplayer i-fiction engine.

2.1 Case-Based Reasoning in Interactive Storytelling

In searching for the right computational solution, widely different approaches have been proposed by researchers in the area. For example, there are proposals based on dynamic behavior of non-player characters that achieve dramatic goals [1,10]. Other approaches give more responsibility to a central dramatic planning algorithm, using directable non-player characters [5,2] or just a stand-alone dramatic planner that controls the most important narrative elements, like characters [6] or the whole fictional world [3]. In the work of [2] the CBR full life cycle -retrieval, adaptation, reuse and repair of previous solutions for new problems- is used for storyline representation and a strategy formalization that allows for storyline adaptation.

2.2 Role-Playing Game Mastering

Tabletop RPGs are exercises in intellect and imagination: a group of players sitting around a table, rolling dices and playing out an imaginary role in a complex shared fantasy, true collaborative narrative.

The Game Master (GM) is a special kind of player, he is the “interactive storyteller”. He designs all the elements of the story and he manage all the possible events that can occur in its development, improvising the dialogue contributions of non-player characters, resolving players actions, etc.

The degree of interactivity in RPG can be enormous, limited only by the players imagination. This implies that no GM, however experienced, can have a deep enough plan to react appropriately to all the possible actions that players can come up with in the world of fiction. To operate successfully without such a plan, GMs must use their imagination, improvise adequate solutions, and continuously rewrite their plots on the fly.

The figure of GM is the best model we have found in real life for designing and directing interactive stories. For the development of the work presented here we have used the description of the relevant heuristics given by Robin Laws [4].

Player Models and Plot Hooks Laws identifies seven basic types of role-players according to their motivation and the sort of characteristics that they expect of a game in order to consider satisfactory. These motivation characteristics are referred to as *plot hooks*.

Power Gamer searches for new abilities and special equipment.

Butt-Kicker always waits for a chance to fight with someone else.

Tactician feels happy in the anticlimax scenes, thinking about logical obstacles to overcome.

Specialist just wants to do the things that his favorite character do.

Method Actor needs situations that test his personality traits.

Storyteller looks for complex plot threads and action movement.

Casual Gamer remains in the background (and has a very low degree of participation in general).

Improvising Algorithm According to Laws, the decisions that a GM must take to achieve a satisfactory game of all his players must take into account the player models into which he has classified them and the particular plot hooks that these models require. This is achieved by applying an *improvising algorithm* that must guide the choices that a GM makes in response to a game situation.

1. Imagine the most obvious result.
2. Imagine the most challenging result ¹.
3. Imagine the most surprising result.
4. Imagine the result most pleasing to the players.
5. Pick the one that feels right ². This may be done at random, though some weighting should be applied to preserve credibility.
6. Think of consequences ³. If the consequences seem bad, try another choice.

2.3 Internet Adventure Game Engine

Text Adventure Games, broadly known as the Interactive Fiction genre, appeared as the first narrative games at the end of 70's. Originally, interactive fictions are like interactive books, only made of text chunks. They have complex plots and offers a narrative presentation to the player. In this kind of games, story and world simulation are tightly coupled.

Internet Adventure Game Engine (IAGE [9]) is a Java Open Source project for the development of a multiplayer interactive fiction system. In contrast to Massive Multiplayer Online Role-Playing Games (MMORPGs), which maintain a lot of players playing at the same time, with as many ongoing stories as users connected to the server, IAGE allows one pre-authored storyline with the added possibility of having more than one player immersed in the same story. IAGE can be also used to create single player adventures like traditional systems as Inform [7].

¹ Find a challenge based on the goals of the PC that has had the least spotlight-time in the recent interactions.

² Maybe two or more at the same time.

³ A choice should not endanger the success of a climax that had been previously planned by the GM.

3 Our Computational Model of Interactive Storytelling

We are trying to design a Multiplayer & Directed Interactive Fiction Engine driven by Role-Playing Heuristic Rules. We propose a Knowledge Intensive CBR (KI-CBR) approach to the problem of generating interactive stories from the game state and a set of cases specifying how the next scene will be performed. In our model of Interactive Storytelling, adapted from the original RPG conventions, we separate the world simulation from the story control. The IDS system that we are considering has a narrative environment with a high level of interactivity (textual commands) and uses IAGE as a low level world simulator. Over that we have a CBR system that can guide the development of plot, building creative new situations on the fly from the case base.

Our system involves a certain amount of explicit knowledge, an heuristic for assigning player models, and a plot driving algorithm.

Explicit Knowledge: the Ontology and the Case Base For the development of our system structuralist models of conventional narratives have been used. In particular, the character functions of Vladimir Propp's Morphology of the Folk Tale [8] are used as basic recurrent units of a plot. In order to be able to use them computationally, they have been translated into an ontology that gives semantic coherence and structure to our cases. The case base is built by the game author using these functions as building blocks.

An additional ontology provides the background knowledge required by the system, as well as the respective information about characters, places and objects of our world. This is used to measure the semantical distance between similar cases or situations, and maintaining a independent story structure from the simulated world. The domain knowledge of our application is the classic might-and-magic world with magicians, warriors, thieves, princesses, etc.

The author fills the case base with scenes that contain preconditions and postconditions. Each case is structured in terms of an ontology of domain knowledge: character, places, models of players, etc. The semantic constraints between scene transitions are loosely based on the ordering and co-occurrence constraints established between Proppian functions. Because the case base is made using cases proposed by the author, we know that the system makes variants of the same "interactive plot" so the author creativity (at the scene-level) is preserved.

Interactor Models and Scene Postconditions In order to apply the algorithm described above, it is crucial to identify correctly the character profile of the player. In a RPG, this task must be carried out based on limited information like the description of the character explicitly required by the rules, observation of the player's reactions, and possibly the player's record in previous games. To simulate this dynamically in an interactive system is the next step of this study. For our current purposes, it is enough to provide players with a set of seven pre-defined characters, such that each one of them is related with a specific interactor model. It is hoped that players of a specific model will under such conditions

choose the type of character most related to their preferences in playing. The initial adscription of player model to character type will be used by the system to assign player models to the players.

The plot hooks for Power Gamers, Butt-Kickers, Tacticians and Specialists are the easiest to implement. If a scene involves obtaining special abilities, engaging in combat, solving logical puzzles, or displaying particular skills - like magic in the case of magicians, for instance -, this appears explicitly in the postcondition of the scene. This means that the scene cannot finish unless at least one player carries out the postcondition.

In the case of Method Actors and Storytellers, their plot hooks are more complex because they require that not only the choice of content for particular scenes but also the structure of the ongoing story be modified to fulfill their particular requests. When players of one of these models take part, the system must retrieve a sequence of scenes marked as interesting for the model involved. For instance, the system would retrieve for a Method Actor a sequence that involves the development of a moral conflict, and one that introduces a secondary plot for a Storyteller.

In the application of the current version of the plot driving algorithm, the system does not take into account Casual Gamers that take part in the story.

Plot Driving Algorithm Our Algorithm is very similar to Law's one. The case retrieval takes place using similarity assessment over the case base.

The CBR system uses two similarity functions. The first one is used to recover the scene that leads to the most obvious transition from the current scene. The inverse of this similarity function is used to find the scene that provides the most surprising transition.

The second similarity function is used to retrieve the scene that involves the most pleasing transition from the current scene. The definition of *how pleasing a scene is* is given by the number of easy tasks matching the players plot hooks that appear - and disappear - in the transition. The inverse function is used to find the scene that provides the most challenging transition, one which involves the appearance of a high number of difficult tasks that also match the plot hooks of the players.

In this way the algorithm includes a number of obvious paths and other paths that may progressively get more positive for the players interests. Additionally, it may include surprising or negative scenes.

An Example of Scene Transition Selection Group of players: 2 Butt-Kickers, 1 Storyteller and 1 Causal Gamer.

- Scene A: The adventure starts.
- Preconditions: none.
- Description: Players are in a pub, drinking beer and talking about their adventures. There are at least two different non-player characters that have a mission for them.
- Postcondition: accept a mission.

When the algorithm is executed, it decides to select a surprising scene transition. The system retrieves a scene from the case base and the following scene is obtained:

- Scene N: Surprise attack by orcs.
- Preconditions: the previous scene did not involve combat.
- Description: Orcs appear suddenly and attack the players with no motive.
- Postcondition: all the orcs are killed.

By virtue of a heuristic defined by the author, no enemies may appear in the story before the fourth scene. This invalidates the retrieved scene as a possible continuation, so it is rejected and another one is retrieved from the case base.

This time the algorithm chooses an obvious transition, a scene is retrieved and the system comes up with:

- Scene N: The mission starts with a journey.
- Preconditions: players have accepted the mission.
- Description: A long trip must be undertaken to the place where the mission will take place.
- Postcondition: arrive at the journey's end.

The presence of two Butt-Kickers in the group implies that scene N is topped up with arbitrary combats during the journey. This is possible because the scenes in the case base only specify partial descriptions of their content, and the algorithm must fill them in dynamically during the adaptation stage of the CBR cycle. As described, this is done taking the players preferences into account.

For instance, the fact that there is a Storyteller in the group requires the system to carry out additionally some long term planning before it can fill in the scene. In this case, the man that offers the players a mission also charges them with carrying a mysterious box to their journey's end, a box that may not be opened at any time. This introduces the seed of a possible secondary plot that the system can use at later stages to satisfy the plot hooks of the Storyteller.

The presence of a Casual Gamer has played no part in the algorithm.

4 Conclusions

Although the system is not fully-implemented yet, the progress so far points to a reasonable solution to the Interactive Dilema. Once it is fully developed, our approach has applications for every project that is looking for heuristics and algorithms to tackle the interactive dilemma on IDS systems.

We intend to make the system available on the web to let the people play and study the effects of the automated director on the game.

5 Acknowledgements

The first author is supported by a FPI Predoctoral Grant from Complutense University of Madrid. The work was partially funded by the Spanish Committee of Science & Technology (TIC2002-01961).

References

1. M. Cavazza, F. Charles, and S. J. Mead. Character based interactive storytelling. *IEEE Intelligent Systems*, 17:17–22, 2002.
2. C. Fairclough and P. Cunningham. A multiplayer case based story engine. In *GAME-ON Conference*, pages 41–, 2003.
3. A. S. Gordon and N. V. Iuppa. Experience management using storyline adaptation strategies. In *First International Conference on Technologies for Digital Storytelling and Entertainment*, Darmstadt, Germany, 2003.
4. R. D. Laws. *Robin's Laws of Good Game Mastering*. Steve Jackson Games, first edition, 2002.
5. S. C. Marsella, W. L. Johnson, and L. Catherine. Interactive pedagogical drama. In C. Sierra, M. Gini, and J. S. Rosenschein, editors, *Fourth International Conference on Autonomous Agents*, pages 301–308, Barcelona, Spain, 2000. ACM Press.
6. M. Mateas and A. Stern. Façade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference, Game Design track*, 2003.
7. G. Nelson. *Inform Manual*. Interactive Fiction Library, USA, fourth edition, 2001. <http://www.inform-fiction.org/>.
8. V. Propp. *Morphology of the Folktale*. University of Texas Press, Austin, 1968.
9. R. Rawson-Tetley. Internet adventure game engine (IAGE), 2002. <http://www.ifarchive.org/if-archive/programming/iage/>.
10. R. M. Young. Notes on the use of plan structure in the creation of interactive plot. Technical Report FS-99-01, AAAI Fall Symposium on Narrative Intelligence, AAAI Press, Menlo Park, 1999.