A PRACTICAL APPROACH TO INTRODUCE STORY DESIGNERS TO PLANNING

Steve Hoffmann, Ulrike Spierling
Hochschule RheinMain, University of Applied Sciences
Faculty DCSM, Unter den Eichen 5, 65195 Wiesbaden, Germany
{steve.hoffmann, ulrike.spierling}@hs-rm.de

Georg Struck
Timetravel Entertainment
Immanuel-Kirch-Straße 2, 10405 Berlin
georg.struck@gmail.de

ABSTRACT

This paper presents a design methodology that allows technical leads in the game industry to introduce story designers with low technical background to generative techniques for automatic plot creation, in particular to a specific method of AI-based planning. The approach provides support to convey necessary technical knowledge by paper prototyping. Further, it demonstrates that paper prototypes are not only useful to learn these concepts but also as tools of creation.

KEYWORDS

Game Design, Planning Algorithm, Design Method, Visualization, Paper Prototype

1. INTRODUCTION

Research in Interactive Digital Storytelling (IDS) aims at inventing techniques that provide possibilities for players to have meaningful influence on the story of a game. Traditional game design often makes use of branching story lines to achieve this influence, leading to an explosion of paths and content elements that is difficult to manage manually. IDS focuses on overcoming that issue with generative techniques. One reason why only few successful IDS solutions have been adopted by the games industry is that they are technically more demanding than traditional branching storylines. Technical leads often work with team members who are experts for story creation with little technical background in Artificial Intelligence (AI) or computer science. This hampers the uptake of more advanced generative technologies for storytelling in games.

We assume that a potential future IDS industry will work with a similar role distribution as in the actual game industry, however also including story engineers as designers for generative story engines that co-define the interactive story experience (see Figure 1).

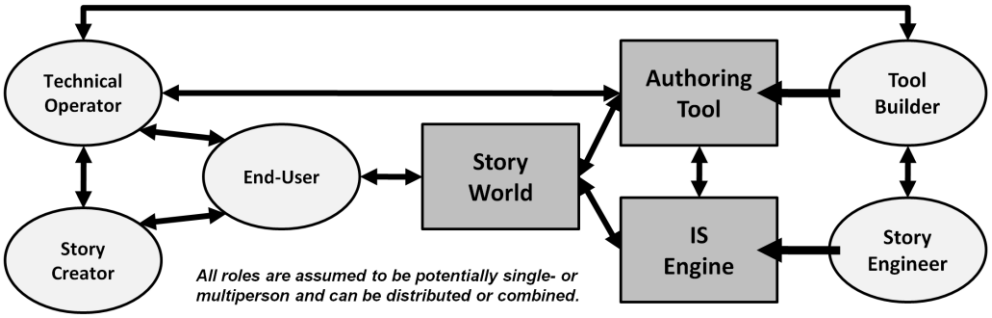

Figure 1. Assumed roles in the creation of an interactive story artifact.

The current state of the art in IDS shows that right now, roles like in the above Figure are often combined and it is not uncommon that engineering and authoring tasks are performed by the same person in small projects (Spierling et al., 2010). When extending the scope to larger development teams, there is a need to introduce game designers and story authors to generative concepts. Even though authors may not need to implement technical aspects, it is vital for their creative work to understand the nature and philosophy by which story engines generate new content.

While in traditional storytelling authors explicitly predefine all possible paths, directly controlling all decision points, in IDS the plot structure is in part defined only implicitly by the declaration of rules and abstract models (Spierling, 2009). Thus, we assume that current authors have to deal with two opposing ways of creating a plot. On the one hand, they create *models* for all parts that are to be generated automatically, and on the other hand, explicitly defined *actions, events and states* are still to be declared for the non-generative story parts.

In the following, we describe a case study in which we

- explore the transformation of a draft story into a story model and
- use the model as a strategy for educating story creators in technical concepts of IDS.

We decided to use AI-based planning, one prevalent generative method used for drama management in IDS (see below). The created model has then been converted into a physical card game (the paper prototype described below) and has been test-implemented with an authoring tool equipped with planning software[1] (Pizzi and Cavazza, 2008).

2. FUNDAMENTALS AND RELATED WORK

Although AI-based planning – as described by Russell and Norvig (2003) – is one prevalent method used in IDS research, for example by (Charles et al., 2003), (Thomas and Young, 2006), (Thomas, 2006), (Pizzi and Cavazza, 2008), (Porteous and Cavazza, 2009), (Riedl, 2009), (Roberts et al., 2009), and (Skorupski, 2009), there is a lack of explanatory material that can be used as an introduction combined with creative story conception. For authors, AI-based planning at first may appear like an alien concept from the world of computer science. However, one important and interesting fact is that there are basic parallels between planning and storytelling. Therefore, technical fundamentals have to be explained with graspable metaphors before such an approach can be employed.

As pointed out by (Barros and Musse, 2007), there is a clear correspondence between planning algorithms and stories. Planning algorithms operate on the basis of causal relationships between actions. Stories are sequences of actions/events related through some form of causality. Because plans are composed of discrete operations and stories can be seen as sequences of events, they can be converted to computer-based models. The role of planning in IDS applications *"is to define the actions or events that must occur during the story so that the world changes from its initial state to some goal state"* (Barros and Musse, 2007). This point is shared by (Li and Riedl, 2010) who sum up that *"plans closely resemble cognitive models of narrative"* and that *"cognitive science and neuroscience suggests that planning may be a very appropriate computational means for narratives"*.

Planners can create an order of actions (or events) dynamically. The use of planning software may not directly reduce the amount of actions that have to be authored, but offers a greater degree of non-linearity and variation within a prepared set of actions/events. Once a story is decomposed into single actions (operators) of partial order, a high number of story paths within the same created repository of actions/events becomes possible. Another advantage of some planning algorithms is the ability to adapt to changes of the world state during runtime and to perform re-planning if the user or other agents interfere with the current plan, and the firstly authored course of actions is possibly changed.

[1] EmoEmma-AuthoringTool, http://redcap.interactive-storytelling.de/authoring-tools/emo-emma

Traditionally, the scope of software-based planning is to create optimal and *efficient ways* to transform the initial world state into the desired goal state(s). Planning with respect to IDS expands the scope from the *efficiency* of this transformation to *interesting ways* of the transformation itself. The resulting order of actions should lead to an interesting and suspenseful experience for the user. In successful storytelling, primarily 'the journey is the reward', and reaching the goal (or end of the story) most efficiently or quickly may be even undesirable.

The fundamentals of planning in Artificial Intelligence are described by some basic conceptual elements. Addressing the interests of story creators for games, these can be divided on the one hand into elements that have to be explicitly created during an authoring process, and on the other hand into only implicitly created elements that are finally generated by automatic planning during the runtime of the generative story engine.

Explicitly authored elements include:

- *Facts* (propositions): These are sentences that can be part of the story, describing possible elementary and changeable situations of people and objects, for example: 'The cigarette is lit'. During authoring, a database of facts that are possible and relevant for the story needs to be collected.
- *Actions* (operators): These are events with the possibility to modify the validity of facts (propositions). For example: 'To light a cigarette'.
    - *Preconditions* (propositions): For each action, necessary conditions have to be declared defining a situation in which that action is allowed to be performed. This is a proposition, for example 'The cigarette is not lit'.
    - *Effects* (propositions): Also for each action, its transforming effect needs to be declared during authoring. This can be done by declaring which facts become true (are added to the world) or false (deleted from the world) after the action is performed. For example, the action 'To light a cigarette' adds the proposition 'The cigarette is lit' and deletes the proposition 'The cigarette is not lit'.
- *Initial state* and *goal state(s)* (collections of propositions): Authors need to specify a selection of their defined facts (propositions) that are true at the beginning of the story, which describes the initial state of the world before the plot starts. Further, they define one or more possible goal state(s) as a selection of their facts (propositions) that describes the end of the story.

Implicitly created (generated) elements include:

- *States* (world states of the story, each of which is a set of actually valid facts (propositions) at each step of plan execution).
- Chains (sequences) of plausibly ordered actions changing world states.

The planning process can be divided in two parts, performed step-by-step creating a generated sequence of actions changing the world state:

1. Depending on the current world state, *find* possible actions (where preconditions are met), and
2. if multiple actions are possible, *choose* the 'best' action.

Each meaningfully authored action contains preconditions, describing under which circumstances the action is possible, and effects, which describe changes to the world after the action was performed. Both are defined by facts, the so-called propositions. The effects are divided into sets of propositions that are added and/or deleted from the current world state. The world state itself is defined by these propositions, describing the world with facts.

The choice of the 'best' next action is determined by a quality function. In planning-based IDS systems, designing these quality functions has the strongest impact on the final outcome and is central to the authoring process.

3. DESIGN PROCESS OF AN EXAMPLE STORY

This section describes a suggested process of design by a concrete example story ‘Harold in Trouble’. The process can be used by technical leads to collaborate on a non-linear story with a team of story designers. Parts of it can be performed as a collaborative workshop or seminar and can be scaled up or down depending on the size of the team and the intended results.

3.1 Story Outline

The first step is to create a written story outline and to define scenes, characters and their goals within the story. This can be done in a brainstorming phase, in which possible scene descriptions are written down on cards. In our example, we created a kind of ‘James Bond’ story, with criminal super brain ‘Silvertoe’ wanting to blackmail the world.

The second step is to analyse the draft story and find a suitable genre. In our example we decided to choose parody to create a story with some humorous situations. So we developed a comic character, wannabe womanizer ‘Harold’ as the clumsy assistant to the agent who wants to stop Silvertoe. As a design principle, all effects of his actions have a contrary outcome to what would be proposed, creating havoc.

Every scene needs a goal to be reached. In our scene designed for the planner, the goal is reached when Silvertoe leaves the party with anger. While Harold tries to seduce the female guests, he creates chain reactions that let the event get out of control, like inflaming a poodle, poisoning the punch, drenching other guests and damaging the music equipment – thereby raising the anger level of Silvertoe. The goal of this scene is to make Silvertoe so angry that he cancels negotiations with other villains and leaves his own party.

3.2 Events and Actions

The next step is to extract possible meaningful actions and events from the story draft. In our story ‘Harold in Trouble’, two abstract actions at a high hierarchical level (subsuming other actions) were identified, which are crucial for the flow of the intended story experience: *‘creating havoc’* and *‘seducing’*. The comical theme of the story can be well expressed by an involuntarily created chaos resulting from chain reactions making the happenings worse in each step. These chain reactions are also suited for planning in general because they allow for dynamic connections of different chains to each other. To initiate the chain reactions, the abstract action ’seduce’ was chosen, fitting one main characteristic of Harold. Table 1 shows some concrete actions as variables that can be used to represent this abstract action.

3.3 States and Attributes

The next step is to extract story-relevant variable states and attributes, leading to the definition of propositions. The final goal of our scene is to make Silvertoe so angry that he leaves the party. So the most important changing states in the story are the various anger levels of Silvertoe, which have to be described as facts (propositions). The resulting exact order of events is not that important any more for the outcome of the scene, as long as actions increase that value. Further, we also need to describe all possible states of attributes in the world of the story, again as propositions in the form of sentences about possible facts, such as ‘Cigarette is lit’.

3.4 Groups and Alternatives

As implied above, we suggest the use of abstraction layers to group and structure collected ideas for actions and for variable attributes. Table 1 shows the example of the high-level action ‘seduce’, which gets increasingly specified at lower levels of abstraction in the sense of concrete actions that are to be finally used in the story. In practice for creation, such a table is also helpful for structuring the results of the brainstorming of alternatives for actions, because it is possible to fill in actions at different abstraction levels and to think later about higher or lower levels.

Table 1. Example of grouping and abstraction.

Abstract Level 1	Abstract Level 2	Abstract Level 3	Abstract Level 4
seduce	impress women (boast)	riches	house
			car
		skills/experiences/ knowledge	climbing
			riding
	help women (gain debt)	short term	get a drink
			light a cigarette
		long term	offer better job
			invite to travel

3.5 Chains

As stated above, the creation of plausibly ordered sequences of events or actions is actually meant to be generated automatically by planning software, processing on the authored structure of actions and their situational conditions. Still, we suggest that in order to help define these situations, the anticipation of potential chains of events is an important step in authoring. Actions collected previously are then examined for possible connections and are linked together if they seem to be believable and interesting as a chain. This can easily be visualized by a graph. Beginning to create chains with actions at a higher abstraction level reduces the complexity in the first steps. Actions of lower level can then be filled in later into the already existing graph to allow for alternatives and variable paths.

3.6 Action Sets Including Situational Information

Each action needs to be equipped with information on how it changes the world of the story. This is accomplished by adding preconditions and effects. For each action, at first, circumstances have to be considered in which this action becomes possible. These are defined by variable cases of attribute states, described for each action by a list of propositions to be checked as true. For example, the action ‘Harold lights cigarette with a match’ can only happen if three facts are true, namely that a woman is bored, has a cigarette in her hand, and that the cigarette is not lit. Table 2 shows how this action is described with these preconditions. Likewise, the effects of the action on the resulting world state are noted. In our case, they are described by the removal of one fact ‘Cigarette is not lit’ and the addition of the new fact ‘Cigarette is lit’. (compare Table 2).

This representation of the story as a structured model can be directly transferred into a planning system. However, having only created the first draft, we want to experiment with the resulting dynamic sequencing before implementation, which would be also a barrier for non-programmers. The use of paper prototypes, for example with index cards, is very helpful in this design phase, because it is possible to write down only one part of the description and think about others later, for example leaving the preconditions open and focusing on the effects at first. We can immediately play with the incomplete cards such as in the card game described below, although the model is not yet technically consistent to be successfully run by software.

Table 2. Example of three full sets of actions and corresponding preconditions and effects.

Preconditions	Actions	Effects	
		Add	Delete
Bored woman is present Woman has cigarette in hand Cigarette is not lit	**Harold lights cigarette with a match**	Cigarette is lit	Cigarette is not lit
Cigarette is lit	**Harold carelessly throws match away**	Fire is spread	Cigarette is lit
Fire is spread Poodle is present	**Poodle burns**	Someone burns Silvertoes anger raised by 1 Poodle is burning	Fire is spread

4. IMPLEMENTATION

In our proposed paper-based design method, there are two types of cards: 'action cards' and 'proposition cards', which are dynamically arranged during playing of a card game to describe the ever-changing world state. The cards allow story designers to dynamically visualize and test the possible playout of their story, while gently introducing them to the structural requirements of planning systems. Figure 2 shows an action card of our card game prototype on the left and a suggested layout for the game on the right.

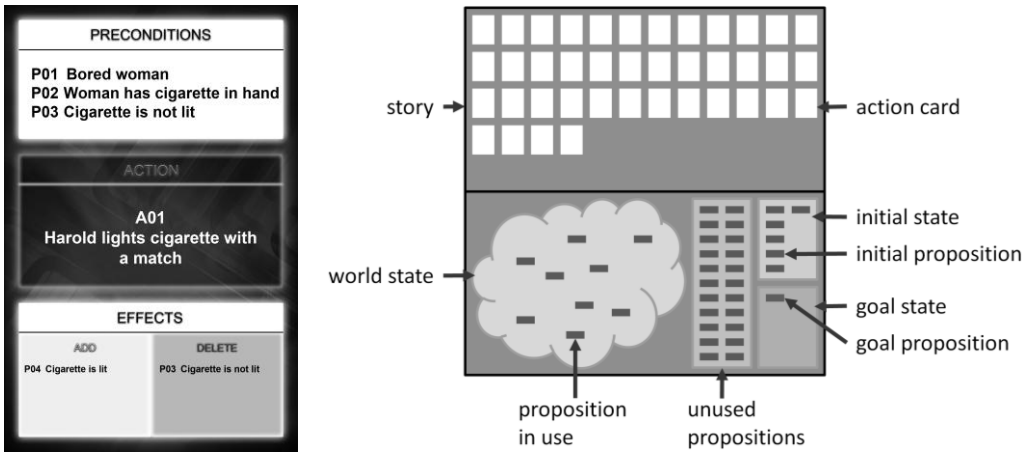

Figure 2. Left: Example action card. Right: Suggested layout of a card game.

This paper prototype works as kind of a card game that simulates some working aspects of an AI planner through a team of players, who form a human representation of the planning algorithm. One additional player can take the role of an interacting user with that 'human engine'. To play-test with the paper prototype, the initial state has to be created first by placing the corresponding proposition cards into the world state area on the table (see Figure 2). The participant in the role of the interacting user can add or remove facts to/from the state of the story world, forcing the 'planner' - the card game players - to react and to perform dynamic re-planning. They have to compare the propositions listed as preconditions on the cards in their hand with the current world state. If there is a matching card, it can be played and as such become an action in the evolving storyline. With each action, its effects have to be performed by adding and removing propositions as written on the played card. This is to be repeated in turns until a point is reached at which the world state corresponds to the goal state, ending the sequence.

The team of players – during story prototyping – can discuss different rules of the card game that allow some experimentation and exploration of planning strategies. For example, it is possible to decide whether or

not an action card can be reused after it is played once. If it returns to the hand of a player, wanted or unwanted loops in the story become possible, because single cards and therefore whole chains can now be experienced again. If more than one action card become possible in any current world state, the question arises to decide which card is given priority. This problem needs to be solved by the definition of a so-called 'quality function', determining the 'best' next action of all possible. This is a creative decision and can depend on several things, for example on the potential to open up many possible following branches, or the potential to end the story in a way desired by the author, or the potential to create a desired world state (such as 'havoc'), if that turns out as a design goal for the chosen story genre. The prototype allows story designers and technical leads to test, discuss and refine these quality functions in a visual and playful way.

After some refinements and for the support of increasing complexity, the paper-based model can also be implemented with an AI-based planning software, making it easier to modify and quickly test changes. We used the EmoEmma Authoring Tool (Pizzi, 2009) to evaluate the feasibility (Spierling et al., 2010).

5. DISCUSSION AND FUTURE WORK

Working with the prototype shows that in the first phases of story creation and with smaller projects, using planning in IDS and games does not reduce the authoring effort. Generative systems based on planning do not create new actions. Instead, authors/designers still have to predefine them together with their acting conditions. However, the result is a dynamic story world which offers end-users a non-linear experience concerning the order of events, which – if designed well enough – may influence each other and result in new directions within the story. Beyond small projects, planning becomes all the more useful the more actions are created and the more difficult it would be to design all story branches in a 'hard-wired' way.

The shown modeling process will be further explored with increasingly complex stories, which is necessary to enunciate understandable design principles for story creation making use of AI-planning. We expect a certain level of story complexity to be a threshold beyond which automatic prototyping methods outmatch manual solutions, such as our card game. This threshold needs to be explored by further creation experiments.

6. CONCLUSION

We presented a paper-based design method intended to introduce story designers to the field of AI-based planning. By playing with the cards, story designers gain a deeper insight into the underlying principles, opportunities and limitations of planning technology. We invite lead game designers who already have technical knowledge in the field of planning or are interested in it to use the presented card game to introduce their team to this field in a playful way. By the help of the proposed modeling process, they can design further paper prototypes similar to the card game, and can consider whether planning would be a promising approach for their project. The resulting card game and further material is available for public access[2], provided by the IRIS Network of Excellence[3].

ACKNOWLEDGMENTS

This work has been funded (in part) by the European Commission under grant agreement IRIS (FP7-ICT-231824).

[2] http://iris.interactive-storytelling.de
[3] http://iris.scm.tees.ac.uk

REFERENCES

Barros, L. M., Musse, S. R., 2007. Planning algorithms for interactive storytelling. In: *ACM Computers in Entertainment*, Volume 5, Number 1, pp. 1-14.

Bateman, C. (ed.), 2007. *Game Writing: Narrative Skills for Videogames*. Featured by IGDA. Charles River Media, USA.

Boon, R., 2007. Chapter 3: Game Writing. In: Bateman, C. (ed.): *Game Writing: Narrative Skills for Videogames.* Featured by IGDA. Charles River Media, USA.

Charles, F., Lozano, M., Mead, S. J., Bisquerra A. F., Cavazza M., 2003. Planning Formalisms and Authoring in Interactive Storytelling. In: *Technologies for Interactive Digital Storytelling and Entertainment, Proceedings of TIDSE 2003*, ZGDV Computer Graphik Edition Band 9, Fraunhofer IRB Verlag, Stuttgart, pp. 216-225.

IRIS, 2009. *Integrating Research in Interactive Storytelling*, IRIS NoE FP7-ICT-231824, http://cordis.europa.eu/fp7/ict/content-knowledge/iris_en.html (Last access: April 2011).

Li, B., Riedl, M. O., 2010. Planning for Individualized Experiences with Quest-Centric Game Adaptation. *Proceedings of the ICAPS'10 Workshop on Planning in Games*, Toronto, Canada.

Pizzi, D., 2009: *EmoEmma Authoring Tool ‐ Manual.*
http://www-scm.tees.ac.uk/d.pizzi/LRRH09/EmoEmma-AuthoringTool-Manual.pdf (Last access: April 2011).

Pizzi, D., Cavazza, M., 2008. From Debugging to Authoring: Adapting Productivity Tools to Narrative Content Description. In: Spierling, U., Szilas, N. (eds.): *Interactive Storytelling, Proceedings of ICIDS 2008*, LNCS, vol. 5334, Springer Verlag Berlin-Heidelberg, pp. 285-296.

Porteous, J., Cavazza, M., 2009. Controlling Narrative Generation with Planning Trajectories: the Role of Constraints. In: Zagalo, N., Iurgel, I. Petta, P. (eds.): *Interactive Storytelling, Proceedings of ICIDS 2009*, LNCS, vol. 5915, Springer Verlag Berlin-Heidelberg, pp. 234-245.

Riedl, M. O., 2009. Incorporating Authorial Intent into Generative Narrative Systems. In: *Proceedings of AAAI Spring Symposium (AAAI 2009),* Stanford, California, USA, March 23–25, 2009.

Roberts, D. L., Riedl, M. O., Isbell, C. L., 2009. Beyond Adversarial: The Case for Game AI as Storytelling. In: *Proceedings for the Conference of the Digital Games Research Association (DiGRA 2009),* London, UK, 2009.

Russell, S., Norvig, P., 2003. *Artificial Intelligence – A Modern Approach*. 2nd Edition, Pearson Education International, Prentice Hall, Upper Saddle River, USA.

Skorupski, J., 2009. Storyboard authoring of plan-based interactive dramas. In: *Proceedings of the 4th International Conference on Foundations of Digital Games (FDG 2009),* Orlando, USA, April 26-30, 2009.

Spierling, U., 2009. Models for Interactive Narrative Actions. In: Ryan, M. (ed.): *Interactive Entertainment, IE 2009, Conference Proceedings*, Sydney, Australia; ACM Digital Library.

Spierling, U., Hoffmann, S., Szilas, N., Richle, U., 2010. *Educational Material for Creators (Non-Computer-Scientists) on AI-based Generative Methods*. Deliverable 3.2 of the IRIS project (EU-FP7), http://iris.scm.tees.ac.uk/publications (Last access: April 2011).

Thomas, J. M., 2006. Collaborative Authoring of Plan-Based Interactive Narrative. In: *Proceedings of Workshop on AI Planning for Computer Games and Synthetic Characters (ICAPS 2006),* Ambleside, The English Lake District, U.K., June 6-10, 2006.

Thomas, J. M., Young, R. M., 2006. Author in the Loop: Using Mixed-Initiative Planning to Improve Interactive Narrative. In: *Proceedings of Workshop on AI Planning for Computer Games and Synthetic Characters (ICAPS 2006),* Ambleside, The English Lake District, U.K., June 6-10, 2006.