Authoring Issues Beyond Tools

Ulrike Spierling[1], Nicolas Szilas[2]

[1] FH Erfurt, University of Applied Sciences, Altonaer Str. 25, 99085 Erfurt, Germany
[2] TECFA, FPSE, University of Geneva, CH 1211 Genève 4, Switzerland
spierling@fh-erfurt.de, Nicolas.Szilas@unige.ch

Abstract. Authoring is still considered a bottleneck in successful Interactive Storytelling and Drama. The claim for intuitive authoring tools is high, especially for tools that allow storytellers and artists to define dynamic content that can be run with an AI-based story engine. Two concrete authoring processes in different cases of recent Interactive Storytelling prototypes have been explored in depth, and feedback from these practical steps has been provided. The result is rather the presentation of general issues in authoring Interactive Storytelling, than of particular problems of a specific system that could be overcome by 'simply' designing the right interface. Priorities for future developments are outlined.

Keywords. interactive storytelling, interactive drama, authoring, creation process.

1 Introduction

The creation of an Interactive Storytelling experience is considered a difficult endeavour. It is aimed at an experience of an artifact that requires the execution of software constituting a dynamic story engine, which controls the unfolding of drama. This rather technical perspective is one of the main challenges that have recently been discussed at Interactive Storytelling conferences [14]. Dynamic story engines are complex software, equipped with Artificial Intelligence algorithms capable of reacting meaningfully to an interacting user, while maintaining a storyline model incorporated within the system.

Recent discussions about the issue of authoring suggest that it is hard to clearly define what steps of creation falls within the scope of authoring, and where the boundaries of so-called authoring tools are located. This is due to the fact that in Interactive Storytelling, on the one hand we assign a co-creating role to the user regarding the resulting story experience, and on the other hand we can't precisely tell the difference between authoring a dynamic storyworld and programming the engine. There are also differences inherent to several approaches, resulting in genre-like interpretations of what actually is Interactive Storytelling.

Therefore, it is necessary to first define the subject of this paper: "Authoring". After the definition and discussion of its boundaries, we will explore the state of the art of authoring for current story engines from a practical point of view. We focus our search on general issues that are most likely "here to stay" persistently, because of their independence from the (potential) lack in usability of some graphical user interface.

1.1 The Case of the Authoring Problem within Interactive Storytelling

We are discussing types of Interactive Storytelling (IS), in which a user (or more than one) interacts with a digital system of agents during the unfolding and concurrent experience of a narrative[1]. This system of digital agents is considered to be the created Interactive Storytelling (IS) artifact. Such an artifact consists of:

a) an IS storyworld, running on
b) an IS runtime engine.

The IS runtime engine enables the performance of agents' autonomous or semi-autonomous behaviour, which means that agents act independently of the author after the actual authoring phase is finished. This engine is a software architecture including specific IS platform components (e.g., story structure manager, planning, interaction/dialogue manager, representation managers, other agents ...).

The IS storyworld constitutes the actual "content", is created by a creator or author (or a team of creators / authors), and uses the agent functionality of the IS engine. Authors need for example to define the storyworld's specific characters as instances of the engine's generic agents. As a special difficulty, the user is as well an active agent (maybe a character) of the storyworld; the creator has to consider this in the creation of the storyworld. The content contains not only components and assets, but also rules and conditions for their occurrence and effects of actions. As such, the created content is code running on the IS engine.

Examples for such IS artifacts are Façade [15] and FearNot! [1], which are IS projects with integrated storyworlds and agent engines. Other IS research projects have built story engines that allow the authoring of various storyworlds. Examples are the engine 'Storytron' [16] which can run several storyworlds, such as *Balance of Power*, or the two examples discussed in the next section, IDtension (running the storyworld *The Mutiny*) and Scenejo (with the *Killer Phrase Game*). In all cases, there is an end-user who interactively experiences the storyworld by playing a role in it.

Authoring means delivering content for somebody else's (an end-user) experience. It is different from a potential kind of co-creation that can take place when end-users interact with a storyworld. However, there is a blurry borderline of the authoring of the storyworld as part of a delivered artifact and the end-users' co-creation during the experience. In Figure 1, this blurry line is symbolized between the "Interaction" level and the "Storyworld" level as part of the IS artifact. Another blurry line is drawn between the runtime engine and the storyworld. This refers to the circumstance that an IS storyworld can only work in co-existence with the runtime engine, which has been produced (mostly before) by a development team of computer scientists.

[1] We are aware that this is a rather technical definition. It is necessary to distinguish from other ("branching") phenomena that might be grouped under the term "Interactive Storytelling".

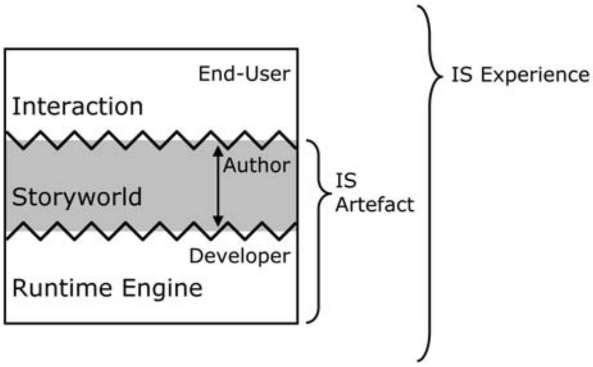

Fig. 1. Definition of the boundaries of authoring. There are blurry lines on the border to the development of a runtime engine, as well as on the border of interacting with the content.

We assume that the developers in this model are computer scientists and that authors are from the creative media fields, for example writers, designers etc. Authors create something for somebody else – this statement excludes that we would also be talking about the end-user co-creation in the narrative pardox. Recent discussions about "Authoring" have been addressing the development of authoring tools that allow creative media experts the creation of a dynamic storyworld without knowledge in programming. The goal of this paper is to present an overview of general problems that currently exist in this authoring process between the two levels of developing and authoring.

1.2 Related Work

Recent discussions on authoring have been followed up in workshops [14], and publications at conferences on that topic, e.g. [13]. However, there was less work on dealing with general authoring problems than with the suggestion of new authoring tools, which often provide GUI representations for unique engine functionalities. There have been few attempts to propose general authoring principles or tool classifications and outlines. For example, Pizzi [8] divided authoring tools according to their dimensions of generative abilities of the underlying engine and the visibility of their content structure, while focusing on the aspect of visualizing and debugging of plan structures. Louchart et al. [4] proposed a metaphorical landscape as a visualization for emerging plotlines. Medler and Magerko [6] defined rather general requirements such as usability, debugging, control of pacing/timing and generality. A similar problem of that presented here was the basis for Mateas and Stern's article on procedural authorship [5], with the conclusion that "authors must program". While we agree that authors must have some level of procedural literacy, we don't see a reason to not develop better tools that educate authors in what they shall do. Further, we believe that programming skills and authoring tools alone don't solve the problem, and that there are a number of general issues that have to be considered.

The goal of this paper is to give an illustration of "real" problems that are present in current content development for IS. It is the first step of an effort to bridge a perceived gap between creative authors and obscure technology by analyzing the affordances of current tools for creation.

2 Experienced Feedback from Real Authoring Exercises

In the following, *general* authoring problems are outlined that have been observed during the practical creation of storyworlds, which in the end effectively run on interactive narrative engines. We take examples of our own systems and authoring tools to illustrate these problems: IDtension [17, 18], Scenejo [13], Rencontre[10]. To complement the data, we also use feedback found in literature, since the goal is not to blame one specific tool but to generalize the issues.

The interactive storyworlds we created with IDtension [19] and Scenejo [13] have been:

- "The Mutiny"; synopsis: *As a sailor jailed in a 17th century galleon, your goal is to take the leadership by preparing a small riot.* IDtension grants the player action possibilities with a higher range as in other systems, by a text interface.
- "The Killer Phrase Game"; synopsis: *As the moderator of a public debate on airport extension, you have to control the fairness level, otherwise the dispute escalates.* Scenejo allows users to text-chat along with 2 virtual characters.

2.1 Story Ideas that don't Fit into the Engine's Approach

2.1.1 Finding Authors

The initial phase in starting a project in IS is to find authors. This initial phase has been skipped in many recent research projects, when the author was the system designer himself, the best example being Façade. But in the general case, and for the sake of dissemination of interactive narrative, a specific author must be found to create new stories that run on a system. This initial phase has often turned out to be less easy than expected. Of course, because the IS systems we are working with are research-based prototypes, we do not expect finding authors who 'a priori' understand the authoring framework. Approaching authors always implied having to explain in detail the approach and principles of the system. But the outcome of this explanation still appears unpredictable.

With IDtension, we had the experience of spending two hours explaining the system in detail to a potential author, who later produced a first document totally out of the scope of the engine. In another case, the author produced a document that was not incompatible with the system, but she preferred to remain at a general level of a synopsis, leaving the fine detail of content specification to the system designer. This was the same experience as in the design of the "Killer Phrase Game" for the conversational platform Scenejo. There, we assumed the underlying chatbot principle to pose technical challenges of implementation of the dialogues. But more than that, it also constituted a mental model of the system that was hard to grasp for developing dialogue structures at all, even if at first just "on paper".

A typical situation we encountered in these early stages of looking for authors was that authors were simply reluctant to the idea of reducing human affairs to logical models.

2.1.2 Abstraction

Given their generative nature, IS systems require authors to write at the level of story-related abstract structures. For example, many systems represent stories in terms of characters' or story's goal [2, 18, 21]. These systems use the notion of generic/instantiated data. Such abstract concepts, with which Artificial Intelligence practitioners are well accustomed, remain aloof from usual creative ways of thinking. The author who wrote “The Mutiny”, the scenario used to demo the IDtension system, reported that this way of writing was quite remote from his usual writing activity [17].

When working with RENCONTRE [10], a system that could be considered less abstract, since narrative fragments are not generated (only their sequencing), authors also reported difficulties in grasping the abstract concept of hypersections. More remarkably, the designer and programmer of the other system “IDtension” also found it difficult to write at the particular required abstract level. This observation shows that this authoring difficulty cannot be reduced to a lack of programming skill or procedural literacy of the author.

Similar to Rencontre, also Scenejo has not more ‘generative’ features than slightly restructuring the ordering of predefined utterances, and offering to get interrupted by user’s actions and respond accordingly. However, dialogue states can be tracked by the system such as the increased stress level. Therefore it was necessary for authors to not only write utterances in direct speech, but to model a dynamic system of influences and meanings of abstract speech acts. Experiences showed that computer science students, capable of programming in general, but not with AI, had no advantage in modeling the dialogues. Specific creative knowledge of dialogue abstraction and design was necessary.

2.1.3 Formatted and Constrained Writing

Current IS systems require filling some precise data structures. For creative authors, this may be perceived as “filling a form”, a typical non-creative activity close to using templates that abridge creativity.

With IDtension, surface text had to be written in a spreadsheet file, that was then processed by the runtime engine. The author did not comply with this constraint, and spontaneously chose a word processor, to be able to freely phrase sentences. As a consequence, the produced sentences were partly incompatible with the engine’s text generator, and some rewriting by the system designer was required. In this case, the creativity of an author was limited by the interactive narrative formalism used within the engine.

For Scenejo, an authoring tool was provided that enabled – and forced – authors to directly write in chatbot terms of patterns and templates [12], where a pattern is a precondition that has to become true before an utterance is made, or in other words, the pattern provides the stimulus for each uttered response of a character. The whole dialogue between two characters had to be written separately for each actor, in order to work according to the character-centric approach taken in Scenejo.

Although these are issues that could be partially enhanced with better GUI support through a better authoring tool, the GUI often only replaces typing by clicking, and doesn't avoid the formality of the implementation that's simply necessary with given formalisms in story engines.

2.1.4 Algorithm-Centered Design

Given the constraints just mentioned, an often adopted strategy consists in first looking closely at the computational model and its limitations, in order to then find a story that suits at best this model.

For the engine IDtension for example, we deliberately chose a story ("The Mutiny") with 8 characters, because it fully expresses the richness of the model [19]. But when applying IDtension to an existing training context, with less characters and less inter-character interaction, the resulting global story was less interesting [11]. The pedagogical content, extracted from linear cases, consisted mainly in procedures to be applied by the main character. This context did not leave much room for possibilities such as influencing other characters to perform actions or getting helped by another character of your choice.

All the same, the idea for the Killer Phrase Game that runs on Scenejo was highly dependent on the potential that Scenejo offers to an end-user: Joining in a discussion between two or more chatbots (quite similar to the interaction paradigm in Façade). Starting out with the bot platform in mind, the creative task was to develop situations with real reasons to interrupt an ongoing dialogue between two or more characters, and the objective of moderating a debate suited that paradigm of the platform.

According to Marie-Laure Ryan, Façade's story [15] is chosen according to the limitations of the engine itself: *"As the conversation turns into a domestic fight, it is not too surprising that Grace and Trip increasingly ignore the visitor. With its theme of marital feud, Façade is very successful at minimizing the limitations of its AI module."*[9]

It is difficult to judge if algorithm-centered design is a good or bad thing. It certainly characterizes the emerging field of Interactive Storytelling from the authoring point of view. For Laura Mixon, who has authored stories for the Erasmatron [7], one should not look too closely at the algorithm when designing: *"The first and among the biggest of my mistakes was to try to use every single, pea-pickin' one of the Erasmatron's wide array of features. If there was a button or menu item, I wanted to bring it into play."*

2.1.5 Potential of Engines Underused

Since it appears difficult to grasp the specifics of an engine, and therefore to ground any story design around the underlying computational models, some authors tended to use only a subpart of the engine's features. As a typical experience in first authoring attempts with each of our engines, an author would naturally try to reduce the functionality to a linear or branching structure, which is more intuitive.

For example, the first story that was written with RENCONTRE by an author external to the project did not use fuzzy hypersections, which constitute one of the distinctive features of this engine. Similarly, IDtension implements a system of ethical values which, in fine, has not been exploited enough in existing stories.

Authoring seminars with students have shown that with Scenejo, first attempts to think of story adaptations resulted in ideas for quiz game-like question-answer structures. First, these are more akin to the well-known classical chatbot interaction, than to the potential to have more characters debating with each other, and second, a quiz comes with a built-in branching structure of right and wrong answers. In other words, the result was far from conversational storytelling. It rather resembled well-known structures of casual or adventure games.

This simply told us that the field of Interactive Storytelling is still so immature that because of a lack of examples, the effort for imagination of novel ideas beyond known structures is high. This was the case for example with students of media informatics who found it easy to use the abstract tools, but on the other hand had few ideas. At the same time, it was hard for creative authors to arrive at conceptual models for creation that fit the engines underlying drama or interaction models.

2.1.6 When Authoring and Programming Intersect

Theoretically, an often assumed modus operandi has been that runtime engines should be built first, after which storyworlds can be written based on the runtime engines. Practically, things have tended to happen differently. It hasn't been uncommon that during the writing of content for IS, the engine designer decided to modify the engine with new functionality that accommodated a specific story with new features. In that case, authoring and programming were performed simultaneously, blurring the line between the storyworld and the engine (see Figure 1).

For example, when adapting *The Little Riding Hood* to IDtension, we significantly improved the management of locations, that *The Mutiny* did not use. Motivated through the development of the *Killer Phrase Game*, Scenejo had been equipped with better functionalities for managing the turn taking between the digital bots and the user.

This kind of intersection between writing and programming can definitely be associated to a certain immaturity of the medium of Interactive Storytelling (compared to cinema for example). However, we also presume that there are some aspects of it that are here to stay, because they are inherent to the digital nature of the medium. Given the flexibility of the computer, it must be accepted that such instability is not only unavoidable, but certainly desirable, because it allows to constantly improve the technology instead of freezing it.

2.2 Painful Process of Storyworld Implementation

We grouped in this part the feedback from authors related to the process of story making. It concerns the day to day work with runtime engines and authoring tools while creating an interactive storyworld.

2.2.1 The Time-Consuming Task of Entering the Content

Generally speaking, we still lack usable enough authoring tools to enter the content, although there has been previous work tackling this issue [12]. Still, entering content

– at first sight – resembles programming activities too much, because at least partially, data structures must be entered in text files (such as XML structures) directly. Even with graphical templates that help creating the correct syntax, entering the data takes time and prevents from quickly seeing the result of the created content.

Typical problems that slowed down the processes in our examples include the lack of usable graphical interfaces supporting different perspectives on the content, the lack of control mechanism preventing authors from entering erroneous content, and the existence of several distinct files that are needed for running one storyworld, such as configuration files for various modularized elements, characters, dialogues etc.

With IDtension, we ended up writing narrative structures twice: an initial schema is established in a simple graphical software, which provides a clean overview of the narrative structure but is not connected to the XML effectively needed by the engine. The author had to write the schemas and then enter them into the system. These two files have been hard to maintain and keep synchronized.

As already mentioned in section 2.1.3, what made entering content in Scenejo a tedious task was that dialogue parts and rules had to be written for each character separately, following a character-centric approach. There was a lack of visualizing potential inter-character conversation results of these rules, so that authors of the Killer Phrase Game kept separate Excel files and external drawings to maintain an overview of the planned dialogue sub-lines.

At this point, future work in graphical authoring tools is worthwhile to speed up these processes. More than just providing templates for data input, different perspectives on the same data are necessary, as well as simulation possibilities of the outcome.

2.2.2 Understanding What is Going on under the Hood

In our examples, after the data for a conceived storyworld was entered, the first attempt has rarely been conform to the author's expectations. A process of play-back, testing and tuning has taken place, as it is quite common as well for linear media. But in the case of Interactive Storytelling, modifying the content is much harder, due to both the complexity of the models and the unfinished nature of runtime engines (see 2.1.6). Typically, when perceiving unwanted behaviour of the storyworld during its tested experience, three hypotheses can be made:

1. The storyworld has not been implemented properly by the author. The content including its elements and rules must be tuned accordingly.
2. The runtime engine has a “bug”, in other words, according to the logics of the model, it should behave differently from the way it actually does. The engine must be repaired (debugged) by the developer.
3. The underlying model does not allow performing what the author expected. In this case, either the runtime engine must be extended and enhanced accordingly, or authors need to develop a better conceptual model of the engine’s potential and underlying dramatic model.

During our own experience with IDtension, we found that it was not easy to establish which of the three cases occured. Finally, only the engine designer was able to tell. The adding of debugging/monitoring interfaces, allowing the visualization of internal structures during the execution (such as a list of all possible actions and their multifactor rating by the system) helped to understand better what was happening during an execution.

In the implementation phase of the Killer Phrase Game on Scenejo, it was necessary to discuss regularly between the designers / authors of the conversations and the engine programmers, to find out which of the above three interpretations of an error applied. This communication process slowed down the implementation significantly.

The conclusion to this aspect is that although there is great potential for improvements through better debugging tools, we believe that this issue is something inherent in Interactive Storytelling production in the future. Similar experiences have been made in the beginnings of the 3D animation production area, when graphical designers started to use complex shaders and renderers that provide many parameters for tuning. Experienced designers usually get a good grip on intuitively finding "work-arounds" with given technical constraints. In the case of Interactive Storytelling, however, we have to deal with an even larger complexity.

2.3 Deliberating the End-User Experience

As motivated in section 1.1, the authoring process in IS aims at creating a storyworld that together with a runtime engine forms an artifact to be delivered to end-users. Not before end-users interact with this artifact, Interactive Storytelling occurs as an activity and experience. Depending on the design of the engine model as well as the particular storyworld, the end-user plays a certain role within the storyworld, which is associated with particular possible actions and influences on the outcome.

This experience, which has often been discussed in relation to the notion of "the interactive narrative paradox" is actually something that the author has to conceive. In our view, it is an important – if not the most important – part of the authoring responsibility to care about the whole IS end-user's experience.

Within recent conferences and published literature, IS research has been more focused on algorithms for interactive narrative management than on end-user experience, which has consequences in terms of authoring.

2.3.1 Foreseeing the End Result of the Storyworld Possibilities

While entering data for the storyworld, authors might have difficulties to get an idea of the final result of the interactive narrative.

With IDtension for example, the author in general needs to enter a significant amount of data before getting an idea of the interestingness of the resulting interactive narrative. While testing the story, if no specific surface text is entered, the sentences appear in a crude form, which prevents a proper vision of the final product.

In Scenejo, dialogue pieces could be entered piece by piece and changes could be directly experienced after starting the play mode. This resulted in hearable and readable utterances, spoken by talking heads through a text-to-speech (TTS) converter. Preparations for this realistic playback included that the scene with modeled characters was built in advance and that TTS was connected and set up. Unfortunately it was not possible to change content "on-the-fly". This meant that there has been a long design cycle, because it was necessary to stop the engine, go back to the authoring tool, make changes, and restart the engine from the beginning. With the prototype of Scenejo used in the authoring project, it was hard to focus on a specific situation that occurred late after some playback time, because it was only possible to initiate at the start, but not at a later plot point with given init values at this advanced state.

Through the feedback of the authors who really wanted to achieve a usable storyworld, more suggestions for changes in the authoring tools have been gathered. They concerned the possibility of on-the-fly changes as well as the possibility to scale down parts of the engine, because it also was perceived as a burden to always having to start the 3D world, even if only text occurrences within a dialogue had to be tuned.

2.3.2 Interaction Design

Only after a significant period of authoring effort, first real “play” tests were possible, which here means that other end-users than the authors themselves were called in to interact with the content.

At this point, the next problem occurred in the experience that end-users would not know what to do and how to interact with the storyworld. For example, in the conversational story of Scenejo, the interaction paradigm and style is quite obvious: End-users can type any text to phrase utterances directed at the two bots of the Killer Phrase Game. However, similar as in Façade (and rather worse, since we only developed a fractional amount of content in comparison), only a few of the users’ utterances could potentially lead to perceivable changes in the dialogic turns of the bots. In the limited prototype built, this was addressed by reducing the game to a narrow task assignment for the user of moderating by reacting to killer phrases. We also built in some visualizations of the state changes, to give end-users the possibility to perceive effects of their actions if they influence state values.

We were aware that with these adaptations, we moved the original plan of having a free dialogue towards more narrow task assignment-like game features. On the other hand, this raised the issue that interaction design has to be an immanent job part of authors of a storyworld.

In IDtension, the used mode of interaction – the history-based interface [20] – came late into the project, two years after the beginning of writing *The Mutiny*. Preliminary end-user feedback informed us that this interface has a huge impact on the experience. As for the *Killer Phrase Game,* end-users who interact with *The Mutiny* do not necessarily know what to do, and their behaviour sometimes consisted in clicking everywhere rather than trying to interact with characters in a meaningful way, as expected by the author. Adding a help section within the interface helped in the first instance.

The consequence is that we perceive as important part of the authored storyworld the possibilities for interaction and role-adoption, as well as the interfaces with views on what’s going on, which an author offers to the end-user and provides with a designed shape. In recent discussions on authoring, this issue has been mostly ignored.

3 Conclusion for Overcoming Authoring Issues

In this article, we have presented feedbacks coming from the collaboration of authors and developers in real Interactive Storytelling projects. Not all of the reported issues are to be overcome by simply building the next generation of usable GUI for the immature tools (although enough proposals for this immediately filled the to-do lists). We argue that the current state of the art in creation is far from what's needed to fully embrace the procedural potential offered by future IS engines.

Quite naturally, there are two general ways to overcome the gap between current complex systems and the access for authors more sustainably:

- Listen to authors: Make tools that are intuitive to use for media designers and content creators
- Educate potential authors: Make procedural principles of Interactive Storytelling understandable

As we believe, it is necessary that both lines have to develop in co-evolution. There is a vicious circle at the beginning of this co-evolution, as there are mutual dependencies between the two actions. As was revealed between the lines of some sections (2.1.1, 2.1.5, 2.2.2),we can't expect that newcomers as authors in Interactive Storytelling provide us with proper specifications of their needs, when they still can't grasp the potential offered by engines and by the medium. Authors need prior design experience with the medium. However, designers and other non-AI- practitioners will require tools to get this first design experience, since they will not be able to program the engines directly.

In order to educate authors, procedural principles of Interactive Storytelling – grounded in Artificial Intelligence – have to be generalized to understandable conceptual models and metaphors. Further, design cycles need to be shortened, i.e. authoring tools need a direct connection to runtime engines in order to support these conceptual models, by letting authors experience the interactive quality of their decisions.

Ackowledgements

This work has been funded (in part) by the European Commission under grant agreement IRIS (FP7-ICT-231824). [3]

References

1. Aylett, R., Louchart, S., Dias, J., Paiva, A., Vala, M., Woods, S.: Unscripted narrative for affectively driven characters. IEEE Computer Graphics and Applications, 26(3), 42-52 (2006)
2. Cavazza, M., Charles, F., Mead, S. J.: Characters in Search of an author: AI-based Virtual Storytelling. In Proceedings of the First International Conference on Virtual Storytelling - ICVS 2001. Lecture Notes in Computer Science 2197. Springer Verlag 145-154 (2001)

3. Cavazza, M., Donikian S., Christie M., Spierling U., Szilas N., Vorderer P., Hartmann T., Klimmt C., André E., Champagnat R., Petta P., Olivier P. : The IRIS Network of Excellence: Integrating Research in Interactive Storytelling. In: U. Spierling and N. Szilas (eds.) ICIDS 2008. Proceedings, LNCS, Vol 5334, pp. 14-19. Springer, Heidelberg (2008)
4. Louchart, S., Swartjes, I., Kriegel, M. and Aylett, R.: Purposeful Authoring for Emergent Narrative. In: U. Spierling and N. Szilas (eds.) ICIDS 2008. LNCS, Vol 5334, pp. 285-296. Springer, Heidelberg (2008)
5. Mateas, M., Stern, A.:. Procedural Authorship: A Case-Study Of the Interactive Drama Façade. Digital Arts and Culture (DAC) (2005)
6. Medler, B., Magerko, B.: Scribe: A Tool for Authoring Event Driven Interactive Drama. In: Proceedings of TIDSE 2006, Darmstadt, Springer LNCS (2006)
7. Mixon, L.J.: I CAN"T BELIEVE I DID THAT: A Reflection on Some of My Biggest Shattertown Blunders. http://www.erasmatazz.com/library/StS_Blunders.html (accessed July 2009)
8. Pizzi, D., Cavazza, M.: From Debugging to Authoring: Adapting Productivity Tools to Narrative Content Description. In: U. Spierling and N. Szilas (eds.) Proceedings of the First Joint Conference on Interactive Digital Storytelling (ICIDS). Lecture Note in Computer Science, Vol 5334, pp. 285-296. Springer, Heidelberg (2008)
9. Ryan, M.L.: Peeling the Onion: Layers of Interactivity in Digital Narrative Texts. http://users.frii.com/mlryan/onion.htm (May 2005)
10. Rety, J.-H., Szilas, N., Clément, J., Bouchardon, S: Authoring Interactive Narrative with Hypersections. In Proc. of The 3rd ACM International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA 2008), Athens, Greece, 10-12 Sept. 2008 (2008)
11. Richards, D., Szilas, N., Kavakli, M., Dras, M.. Impacts of Visualisation, Interaction and Immersion on Learning Using an Agent-Based Training Simulation. International Transactions on Systems Science and Applications, 4(1), 43-60 (March 2008)
12. Spierling, U., Weiß, S. and Müller, W.: Towards Accessible Authoring Tools for Interactive Storytelling. In: Goebel, Malkewitz, Iurgel (eds.): Technologies for Interactive Digital Storytelling and Entertainment 2006, Conference Proceedings, Springer LNCS (2006)
13. Spierling, U.: Adding Aspects of “Implicit Creation” to the Authoring Process in Interactive Storytelling.In Proceedings of the Fourth International Conference on Virtual Storytelling - ICVS 2007, Lecture Notes in Computer Science, n. 4871, 13-25. Springer Verlag (2007)
14. Spierling, U., Iurgel, I.: Pre-Conference Demo Workshop “Little Red Cap”: The Authoring Process in Interactive Storytelling. In: Proceedings of TIDSE 2006, Springer LNCS (2006)
15. Stern A., Mateas M.: Integrating Plot, Character and Natural Language Processing in the Interactive Drama Façade. In Göbel et al. (eds) Proc. TIDSE’03. Frauenhofer IRB Verlag, (2003)
16. Storytron, http://www.storytron.com
17. Szilas, N., Marty, O, and Rety, J.-H.: Authoring Highly Generative Interactive Drama. In Proc.of the Second International Conference on Virtual Storytelling - ICVS 2003 (Toulouse, France), Lecture Notes in Computer Science, n. 2897, 37-46. Springer Verlag (2003)
18. Szilas, N.: A Computational Model of an Intelligent Narrator for Interactive Narratives. Applied Artificial Intelligence, 21(8), 753-801 (2007)
19. Szilas, N.: IDtension – Highly Interactive Drama (Demonstration). In Proc. of the 4th Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2008), Stanford, California, USA, October 22-24, 2008
20. Szilas, N. and Kavakli M:. PastMaster@Storytelling: A Controlled Interface for Interactive Drama. In Proceedings of the International Conference on Intelligent User interfaces - IUI 2006 (Sydney, Jan.-Feb. 2006). ACM Press (2006)
21. Young, R. M., Riedl, M. O., Branly, M., Jhala, A., Martin, R.J. Saretto, C. J.: An architecture for integrating plan-based behavior generation with interactive game environments, Journal of Game Development, 1, 1 (2004)