Österreichisches Forschungsinstitut für / Austrian Research Institute for / Artificial Intelligence

TR–2011–22

Nicolas Szilas, Marcos Aristides, Paolo Petta

A Music Engine for Interactive Drama

• Freyung 6/6 • A-1010 Vienna • Austria •

• Phone: +43-1-5336112 •

• mailto:sec@ofai.at •
• http://www.ofai.at/ •

Österreichisches Forschungsinstitut für / Austrian Research Institute for / Artificial Intelligence

TR–2011–22

Nicolas Szilas, Marcos Aristides, Paolo Petta

A Music Engine for Interactive Drama

The Austrian Research Institute for Artificial Intelligence is supported by the
Austrian Federal Ministry for Science and Research and the
Austrian Federal Ministry for Transport, Innovation and Technology.

Citation: Nicolas Szilas, Marcos Aristides and Paolo Petta (2011) A Music Engine for Interactive Drama, in: Licinio Roque and Valter Alves (eds.) Proceedings of the 6th Audio Mostly Conference: A Conference on Interaction with Sound (AM '11), Coimbra, Portugal, ACM Press, 2011. Poster.

A Music Engine for Interactive Drama

Nicolas Szilas
TECFA-FPSE
University of Geneva
CH 1211 Genève 4, Switzerland
nicolas.szilas@unige.ch

Marcos Aristides
TECFA-FPSE
University of Geneva
CH 1211 Genève 4, Switzerland
martima4@etu.unige.ch

Paolo Petta
Austrian Research Institute for Artificial Intelligence, OFAI
Freyung 6/6
A 1010 Vienna, Austria
paolo.petta@ofai.at

ABSTRACT
A number of Interactive Drama prototypes have been created during the last decade. These Artificial Intelligence-based systems usually aim at enabling the user to drive the story as the main character. Despite the acknowledged role of sound and music in visual narrative, almost none of these prototypes includes interactive background music. In this paper, a Music Engine for the IDtension narrative engine is proposed that is able to adapt in real-time to current user's action and narrative states. In the lineage of the branching music approach developed in some video games, the Music Engine being developed in Max/MSP uses a pre-composed graph-based score to enrich the whole interactive narrative experience. In particular, the reactivity of the Music Engine is aimed at corroborating the user's subjective feeling of agency, and thereby at enhancing the experience of Interactive Drama system's main components—user interface, narrative engine, and the theatre—as an integrated whole.

Categories and Subject Descriptors
H.5.5 Sound and Music Computing. J.5 [Arts and Humanities]: Music, Literature.

General Terms
Design, Human Factors.

Keywords
Interactive Drama, Interactive Digital Storytelling, Music Engine.

1. THE HOLY GRAIL OF INTERACTIVE DRAMA
For thirty years, the computer has been used to convey narrative meaning through interactive fiction; narrative hypertext; adventure games; action-adventure games; multi-player games; Sim games; digital storytelling; literary blogs; etc. One particular potential of the computer in the narrative realm has been identified early: it is the user's ability to drive the story by embodying the main character of a fiction and being able to influence in depth the course of the story, as is expected of a story's main character. However, this objective has quickly turned out to be more difficult to manage than expected. The limitations of simple solutions such as branching narrative in hypertext in terms of user's influence have soon become apparent: The number of paths growing exponentially with the number of choices, authors have to limit themselves in terms of modeled user's influence. Conversely, much more interactive approaches such as the very popular game *The Sims™* remain poor in terms of narrative conveyed by the author.

The concept of Interactive Drama, as initiated by the research in the Oz Project at CMU [1], aims at pushing the current limits of existing methods. The general approach consists in generating the story on the fly, by calculating the best narrative event according to the user's actions. Actions are either portrayed by animating virtual 3D characters [2,3], 2D characters [4] or generated text [5,6]. Several research prototypes have been produced so far following different computational approaches. For example, *FearNot!* simulates the cognitive and emotional behaviors of characters and actors [2]; *Defacto* simulates the Aristotelian conflict between goals and norms [7]; *Mimesis* uses planning to recalculate a path to a story ending according to the user's action [8]; *IDtension* uses narrative acts and a model of the user to calculate an optimal narrative experience [6], *Façade* combines reactive characters and larger semi-scripted narrative units to provide a fluid interaction with the user [3]; *Storytron* uses a rule-based system that generate *verbs* describing various narrative events [5]; *Virtual Storyteller* imports concepts from improvisational theatre to enhance a character-based approaches [9] (see [10] for an overview).

None of these systems has solved the challenge of Interactive Drama yet. Some successfully convey a narrative but provide a limited influence to the user (such as *Façade* or *FearNot!*) while others succeed in providing a wide range of options to the user but are less successful in terms of narrative quality (*Storytron* and *IDtension*, for example).

Regarding the output modalities, these prototypes make limited use of sound or music. Sound is used in dialogs by characters [2,3]. *Façade* uses some background music at the end of the story. Most other systems remain silent.

The possibility of using dynamic music to enrich the user experience has remained largely unexplored. To our knowledge, the only study concerns a theoretical classification of various elements of music that could be handled by a Music Engine within an interactive narrative environment [11]. It could be argued that the introduction of sound and music in Interactive Drama is superfluous as long as the core problem of Interactive Drama is not solved. However, we believe that current prototypes would benefit from using the sound modality. Sound and music have proven important in narrative media such as film (see section 3), contributing also to tackle core issues: Therefore, sound should also be explored in Interactive Drama.

Even though sound and music have been used for a long time in video games [12], the field of Interactive Drama opens up new possibilities to dynamically control a music track: These will be explored in the rest of the paper. In particular, we describe a Music Engine that connects to an already developed narrative engine by the first author. We briefly describe this engine in the next section, to enable the readers to understand the potential of dynamic music in the context of such Interactive Drama. We then investigate the role of music in interactive narrative experience from a theoretical point of view (Section 3) before proposing a concrete design solution for a Music Engine (Section 4). In Section 5, the implementation of the engine within an open architecture for Interactive Narrative is described.

2. IDtension

IDtension is an Interactive Drama system that has now been under development for more than a decade [6]. It has been used for several stories and can work either with either text or graphical 3D output modalities. Compared to other existing systems, it is characterized by a high level of interactivity, in terms of choices offered to the user at any time. For example, in the testbed story the user on average has more than 90 options [6].

IDtension calculates story events dynamically according to user's actions. Figure 1 illustrates the engine's internal architecture:

The *World of the Story* contains basic scenario entities (e.g., characters and goals) and facts describing the context (e.g., the story takes place on a boat, where the main character who is being held captive is preparing a riot).

IDtension's *Narrative Logic* calculates the set of all possible actions for each character, based on the data stored in the World of the Story. Actions are decomposed into two components: Generic actions (or action types) correspond to narratology-inspired narrative actions such as "inform", "encourage/ dissuade", "accept/refuse", and "congratulate/condemn". Tasks are story-dependent and describe specific performances such as to read a book, to steal an object, and to flatter a character. Combined with other elements of the system, actions allow the story to develop in several directions without having to script each direction in advance. This twofold articulation enables higher generativity than most existing systems, which means that more can be generated with less authoring effort.

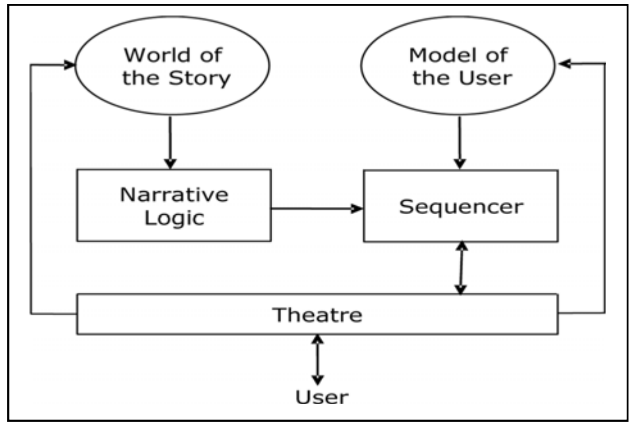

Figure 1. The internal architecture of IDtension

The *Model of the User* estimates the state of the user at any given moment in the narrative in order to provide the Narrative Logic and *Sequencer* with a means of gauging the impact of each possible character action on the user. Relying on the Model of the User, the Sequencer ranks the appropriateness of each action according to a set of narrative criteria designed to ensure the global coherence of the story; the timely occurrence and appropriate complexity of actions; and the maximization of *dramatic conflict*. This last criterion is of particular interest regarding music, since it aims at conveying a kind of affective impact that music is naturally able to reinforce (and we will revisit it in section 4.2).

IDtension evaluates impact (and especially the conflict) according to *values* such as honesty; non-violence; and friendship, and thus weighs the user's choices in ethical terms. This feature is inspired by the central role of values in narrative.

The IDtension *Theatre* displays the story; it manages the computer-user interface. Each user choice is sent to all system modules: the Sequencer receives the set of possible actions from the Narrative Logic, ranks these actions in terms of likely impact on the user and chooses the action to display. In doing so, it interleaves actions chosen by the user and the system. The architecture (see Fig. 1) emphasizes the independence of the narrative logical level from any particular physical manifestation. IDtension can thus be used both for 2D texts [6] and 3D virtual environments [13]. In the text version, the user chooses an action via a system of dynamic menus, while in the 3D version, the user controls the main character in the physical world, choosing the addressee spatially. Figure 2 gives an impression of the text interface.

Figure 2. The text-based user interface of IDtension

This short description of IDtension (see [6] for more details) is to convey how a single sentence displayed to the user is the result of complex processing where the rendered sentence is coded in narrative terms. The system also maintains an estimate of the state of the user. These internal data that are computed on the fly are particularly relevant for musical expressivity.

3. THE FUNCTION OF MUSIC IN INTERACTIVE DRAMA

There is a well established body of knowledge on the role of music in drama in general and film in particular [7]. Through inheritance from the non-interactive ancestor, several functions of music in Interactive Drama can be inferred. It must be stressed, however, that we want to avoid designing an Interactive Drama user experience where music only mimics the cinematic relation of music and narrative. Among other reasons, this relation has matured over the years, establishing a range of motifs and conventions, while Interactive Drama is in its infancy, with its own new conventions still to be developed: To establish an appropriate set of functions of music in this new medium, the specificities of Interactive Drama must also be taken into account.

Furthermore, progress towards this set is achieved iteratively. Therefore, the design of the proposed Music Engine starts with a focus on only the two functions described in this section. Far from meaning to reduce the role of music in Interactive Drama to such a small range, our aim is to use it as a seed for a first implementation, to be improved progressively.

3.1 Reinforcing affective impact

As explained in the previous section, the IDtension narrative engine calculates the estimated affective impact of the dynamically generated action candidates. Whether such impact is actually achieved depends also on a variety of factors related to the presentation of the action to the user, such as the formulation of the text; the adequacy of pace; the intonation of speech; or the body language and facial expression of 3D characters. Music is certainly a powerful means to convey such affect and to some extent compensate weaknesses of automatically generated actions.

We therefore extract affect-related information from the narrative engine and transmit it to the Music Engine, which is responsible for expressing it by adapting the background music.

3.2 Integrating the User Interface

The user interface is usually seen as a distinct layer “on top of” the fictional world, a sheet that should be as thin as possible to favor immersion [14]. However, in menu-based Interactive Drama in particular, this layer is “thick” in that it cannot be fully transparent to the user. This is due to the fact that it is this very interface that enables the user to choose among important and numerous dramatic choices. While this is generally seen as a severe drawback of menu-based graphical user interfaces, we adopt an opposite stance, investigating how the materiality of the interface can be important and useful.

In support of this perspective, we point out how background music is also external to the fictional world (as opposed to diegetic music [15]). But in contrast to the user interface, its role in narrative experience is accepted, also because it is more or less synchronized with diegetic events (events in the fiction). Along this line, our idea is to better integrate the user interface within the whole interactive narrative experience by connecting the interface to the music.

By having the Music Engine react to interaction events occurring at the level of the user interface, we expect that:

1) The interface will be more engaging;
2) The user will feel some local agency, that is, they will *feel* that their actions do *have* an impact on the story, independently of the effective impact.

Many interaction events can be captured at the interface level: the chosen action; the current set of mouse-clickable actions (actions the user could choose next); the number of user clicks since last action (possibly informing on the user’s hesitation); the speed of clicks (possibly denoting the user’s impatience); the absence of interaction; the time spent by the user in making their characters walk/run (in a 3D world).

4. DESCRIPTION OF THE MUSIC ENGINE

4.1 General principle

To delineate the scope of our Music Engine, we established as a basic design constraint to base it on tonal music. Even though we do not claim that tonal music is inherently narrative (as put forward in [16]), will do subscribe to the view that due to its historical relation with the lyric art tonal music does evoke narrative events and affect. Accordingly, the Music Engine should adapt in terms of melodic and harmonic patterns, contributing an interactive score to the interactive narrative.

A large body of research has been conducted in the field of automatic composition and algorithmic music generation, much of which using Artificial Intelligence technologies (see [17] for a review). For this prototype, we have decided to adopt a simple approach: branching music. The composer writes a score in several fragments and indicates the possible transitions between these fragments. If each piece is a node and each transition is a connection, a graph can be obtained. Each transition must be annotated according to a narrative condition. An example graph is depicted in Figure 3.

Such a graph-based solution has been already used in video games [18]. K. Collins terms it “branching parameter-based music” ([12], pp. 147ff). The advantage compared to more generative techniques is that it is more intuitive for a composer; the drawback is that both variability and reactivity are inherently limited. Increasing variability requires writing a large number of differing fragments and conditions; this leads to an overwhelming authoring effort. Reactivity likewise requires reducing the granularity of the fragments, which also contributes to a more complex graph. We are well aware of these limitations, since they are analogous to the limitations of the branching narrative approach encountered in hypertexts (cf. the introduction section of this paper). However, we argue that the main goal of music in Interactive Drama is to support the narrative experience, not to constitute an independent musical object. Therefore, and as a first step, we prefer to be able to rely on a guaranteed level of musical coherence rather than jumping headlong into generative music, where the user may feel more disoriented. What we primarily aim to explore at present are not so much techniques for flexible music generation than the parameters which can be extracted from the narrative engine to inform the Music Engine. These parameters differ from those commonly employed in video games.

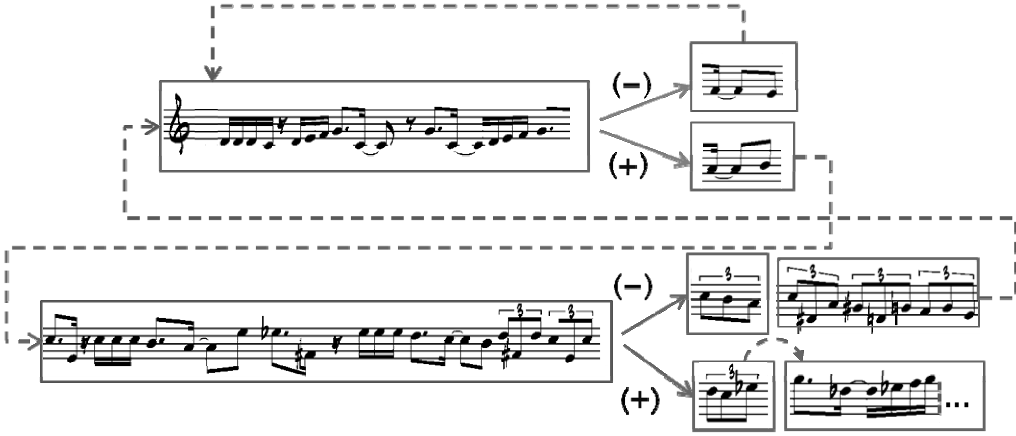

Figure 3. Extract of a branching musical composition. Transitions marked (+) denote conditional transitions when tension is high, while those marked (–) denote conditional transitions when the tension is low (See Section 4.2). Dashed arrows denote transitions without alternatives.

4.2 Narrative conditions

In our first phase of development, we focus on reactivity to the user's actions. Therefore, actions of Non-Player Characters are not considered. When an action is chosen by the user, the narrative engine knows of several attributes related to this action, which include:

- The level of conflict (*c*): as described in section 2, it is the extent to which this action disagrees with the character's ethical values.
- The motivation (*m*): it is the extent to which this action will enable the character to reach an important goal.
- The progression value (*p*): it is the extent to which the action makes the story progress.
- The perceived risk (*r*): it is the probability of failure (defined for some actions only).

How these attributes should impact the background music is a complex issue that points towards advanced algorithms for dynamic music. In our first prototype, we have reduced these four attributes to a single dimension that we denote *tension*. This reduction is calculated as follows (Eqn. 1):

Let *c, m, p, r*, be an action's conflict, motivation, progression, and risk, respectively, and σ_c, σ_m, σ_p, σ_r the respective corresponding *thresholds*. Let *t* be the tension of the action.

$$\text{If } c>\sigma_c \text{ or } p>\sigma_p \text{ or } m>\sigma_m \text{ or } r>\sigma_r \text{ then } t \leftarrow 1 \text{ else } t \leftarrow 0 \quad (1)$$

This calculation is made by the Music Engine on the data sent by the narrative engine. The subsequent mapping to transitions of the music graph uses the tension as narrative condition in a binary manner. Each transition is binary, with one branch being triggered if *t* equals 1, the other if *t* equals 0.

This simple calculation can be improved in the future, for example, by distinguishing more than two tension values, or by typing transitions by conflict, motivation, progression, and risk separately.

4.3 Smoothing the transitions

When a branching occurs between two fragments, the turning point must be repositioned to occur as late as possible, so as to enable an optimal reactivity of the music to the narrative events. For example, if one variant contains the notes ABCDE, and the other variant contains the notes ABCDC, the turning point must be positioned right after the D (i.e., fragment ABCD followed by fragment E or C), rather than after the C (fragment ABC followed by fragment DE or DC) or after the B (fragment AB followed by fragment CDE or CDC). However, if the tonal music contains a usual accompanying track with chords resonance, the composer may want this track not to change abruptly with the transition to avoid the breaking of resonance. In the above example, if the first chord sets in with the note C, the composer may not want to change it right after the D. Neither may they want to have the transition occur just after the B, as it would make the music less reactive.

The proposed solution consists in managing different transition point positions for the two tracks of the music. The first track has a *reactive transition*, triggered according to some narrative conditions (see below). The second track has an associated *anticipated transition* before the reactive transition. When the second track reaches the anticipated transition, two fragments are played together: They contain identical music at first. Then the first track reaches the reactive transition and bifurcates towards one of the two candidate fragments; at the same time, on the second track, one of the two parallel fragments is muted, according to the same transition.

In the above example, the reactive transition occurs just after the D, but the anticipated transition for second track (chords) occurs just after the B, when the first chord starts. Note that the desynchronized transition mechanism implies some constraints on the fragments' compatibility.

A fragment is thus composed of several musical tracks that in some cases are connected to the next fragment in a desynchronized manner.

4.4 Reactivity to the user interface

Besides the affective values of the chosen action, two mechanisms are added to make the music more reactive to the user's activity. The first mechanism deals with amount of user activity. For example, if a user stops choosing actions, the composer may want to progressively suppress the music.

To this end, a new kind of transition is introduced, not based on the latest user's action but on the estimated state of the user, more precisely in this case on a state called “activityLevel”. This state is calculated by the IDtension narrative engine and is provided to the Music Engine. Some transitions are triggered according to this state, as illustrated in Figure 5.

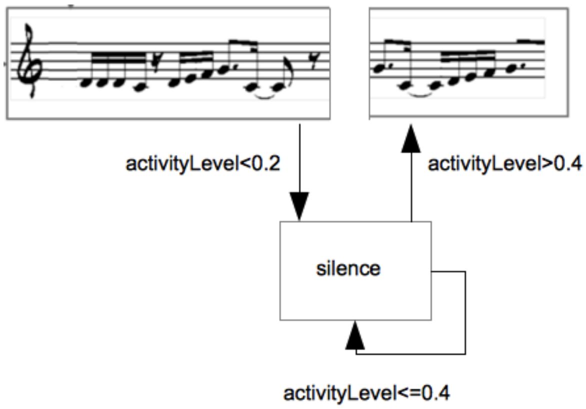

Figure 5. Extract of a branching musical composition in which transitions are triggered according to the level of activity. If the user activity falls below a lower threshold (here: 0.2), the music pauses.

The second mechanism operates on top of the branching approach described so far. It consists of stressing the music each time the user selects an action. For example, as soon as the player chooses an action the next note, if it occurs within a certain time span, changes in both volume and timbre. This simple mechanism is meant to corroborate the user's feeling of local agency, as discussed above.

5. IMPLEMENTATION

Designing a Music Engine requires more than building an interactive music experience. It consists in specifying and developing an autonomous software component with a well specified set of inputs and outputs that communicates with other components involved in an Interactive Drama system. The Music Engine is integrated into a larger project, that is, an open architecture for interactive storytelling.

This open architecture includes, around the Music Engine, the following modules (see Figure 6):

- A *Narrative Engine* that is in charge of the logical calculation of narrative events. It corresponds to the elements of the internal architecture shown in Fig. 1, but for the Theatre.
- A *Theatre* that includes the space where the action is displayed and the mechanism that enables the user to choose an action.
- A *"Shared Narrative State Repository" (SNSR)* where some data are stored and shared between components.
- A *Director* that manages the flow of communication between modules by transmitting messages to the right destination, and convert it if necessary.

All messages circulating across the modules are encoded in XML. Modules communicate via the TCP/IP protocol (sockets), which improves compatibility with a range of technologies. All modules are clients, except for the Director since it is the server.

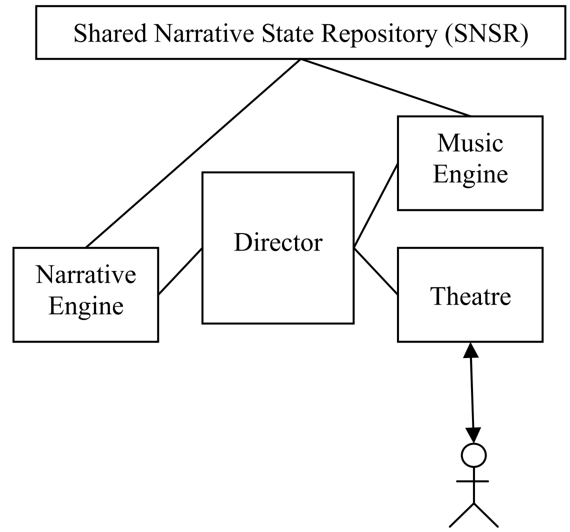

Figure 6. Open architecture containing the Music Engine.

The Music Engine takes the following information as input:

- The action chosen by the user (encoded via a unique identifier) and emitted by the Theatre via the Director.
- The attributes of the actions, which are stored in the SNSR (accessed via a query).

Note that the Music Engine does not receive any information on the content of the portrayed action, because it does not need it. Note also that in the first prototype the Music Engine does not send back any information to the rest of the architecture; it only plays the computed music.

The Music Engine itself is realized using *Max/MSP,* a popular visual programming language for music. Max/MSP is used in conjunction with a socket library to connect to the rest of the architecture. The music format used to process music fragments is MIDI, which is one of the most flexible common formats in terms of dynamic scores [12]. It supports many dynamic modulations that are not used much in this paper but should prove useful in future development. MIDI fragments are written separately using a music notation software (*Finale* in our case) and then read by *Max/MSP*. The *Max/MSP* module manages the transitions described in the previous section according to information received or queried from the rest of the architecture.

At the time of writing, the Theatre and the SRSN are fully developed. The narrative engine needs to be updated to store information inside the SNSR, as specified. The Music Engine is in development.

The story chosen as testbed is "The Mutiny", which is already written in IDtension [6]. The user plays a sailor jailed in a 17th century galleon with three other prisoners after a failed plunder against the ship. His goal is now to gain leadership of the galleon by preparing a riot. This story supports a high degree of interactivity, as the user is given many possibilities of actions such as ask/trade/rob/rob armed with a knife an object; ask another character about their taste (in order to offer them the object they like); or try to flatter other characters to have allies. The theatre is text-based (see Figure 2): Interaction is menu-based; actions are rendered in text form, via template-based language generation techniques. The interactive (branching) score has been composed by the authors specifically for this story. It consists in an “energetic” theme evolving towards more or less tense moments, depending on the branches.

6. CONCLUSION

We have investigated in this paper the development of a Music Engine that plays a score depending on narrative data sent in real-time by a highly generative Interactive Drama system. While the potential functions of music in such interactive narrative experience are manifold, we decided to focus on two functions that are specific to the genre of Interactive Drama, avoiding the “trap” of poorly mimicking movies conventions that might not even be effective in the interactive case.

Following an iterative design philosophy, early experimentation with the first prototype should give us valuable insights regarding future direction of this research, should it be towards more generative adaptive music techniques, increase use of narrative data, adding of MIDI modulation techniques, composition improvement, etc.

7. ACKNOWLEDGMENTS

This work has been funded (in part) by the European Commission under grant agreement IRIS (FP7-ICT-231824). The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry for Transport, Innovation, and Technology.

8. REFERENCES

[1] Smith, S., & Bates, J. 1989. *Towards a Theory of Narrative for Interactive Fiction* (No. CMU-CS-89-121). Pittsburgh, PA: Department of Computer Science, Carnegie-Mellon University.

[2] Aylett, R., Louchart, S., Dias, J., Paiva, A., Vala, M., Woods, S., & Hall, L. 2006. Unscripted Narrative for Affectively Driven Characters. *IEEE Comput. Graph. Appl.* 26, 45-52.

[3] Mateas, M., & Stern, A. 2005. *Structuring content in the Facade interactive drama architecture*. Paper presented at the Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2005).

[4] Rank S., Petta P. 2005. Appraisal for a Character-based Story-World. In Panayiotopoulos T. et al. (eds.), Intelligent Virtual Agents, 5th International Working Conference, IVA 2005, Kos, Greece, September 2005, Proceedings, LNAI 3661, Springer Berlin Heidelberg, pp.495-496.

[5] Storytron, 2011. http://www.storytron.com

[6] Szilas, N. 2007. A computational model of an intelligent narrator for interactive narratives. *Appl. Artif. Intell., 21*, 753-801.

[7] Burt, G. 1994. *The Art of Film Music*. Northeastern University press, Boston.

[8] Sgouros, N. M. 1999. Dynamic Generation, Management and Resolution of Interactive Plots, *Artificial Intelligence, 107,1*, 29-62.

[9] Young, R. M., Riedl, M. O., Branly, M., Jhala, A., Martin, R.J. and Saretto C. J. 2004. An architecture for integrating plan-based behavior generation with interactive game environments, Journal of Game Development 1, 1 (2004), 51-70.

[10] Swartjes, I., Kruizinga, E., & Theune, M. (2008). Let's Pretend I Had a Sword: Late Commitment in Emergent Narrative. In U. Spierling & N. Szilas (Eds.), Interactive Storytelling (Vol. 5334, pp. 264–267). Berlin / Heidelberg: Springer.

[11] IRIS IS Systems, 2011. http://tecfalabs.unige.ch/mediawiki-narrative/index.php/IS_Systems (visited July 2011)

[12] Berndt, A. 2009. Musical Nonlinearity in Interactive Narrative Environments, in: G. Scavone, V. Verfaille, and A. da Silva (eds.) *The International Computer Music Conference (ICMC) 2009*, (McGill University, Montreal, Canada, August 2009), 355-358.

[13] Collins, K. 2008. *Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design*. MIT Press, Cambridge, MA.

[14] Szilas, N., Barles, J., & Kavakli, M. 2007. An implementation of real-time 3D interactive drama. *Computers in Entertainment* 5(1).

[15] Murray, J. 1997. *Hamlet on the Holodeck. The future of narrative in the cyberspace*. New York: Free Press.

[16] Berndt, A. 2011. Diegetic Music: New Interactive Experiences. In *Game Sound Technology and Player Interaction: Concepts and Developments,* M. Grimshaw Ed, IGI Global, Hershey, PA, 2011, 60-76.

[17] Tarasti, E. 2004. Music as a Narrative Art. In *Narrative Across Media*, M.-L. Ed. University of Nebraska Press, Lincoln.

[18] George Papadopoulos, G. 1999. AI Methods for Algorithmic Composition : A survey, a Critical View and Future Prospects. In Proceedings AISB Symposium on Musical Creativity.

[19] Bernstein, D. 1997. Creating an Interactive Audio Environment. http://www.gamasutra.com/view/feature/3238/creating_an_interactive_audio_.php