PERGAMON

Computers & Graphics 26 (2002) 31–44

COMPUTERS & GRAPHICS

www.elsevier.com/locate/cag

Setting the scene: playing digital director in interactive storytelling and creation

Ulrike Spierling*, Dieter Grasbon, Norbert Braun, Ido Iurgel

Department of Digital Storytelling, ZGDV Computer Graphics Center, Rundeturmstrasse 6, D-64283 Darmstadt, Germany

Abstract

Interactive digital storytelling promises to be a new artistic discipline. But what does the artwork look like? And how much control does the artist have over the final result? By searching for the right answers, we draw several approaches from diverse fields such as filmmaking, game design, autonomous agents, psychology, and narrative intelligence.

We first give a definition of interactive storytelling that includes a scope between authorship and emergent narrative, and present two example projects with different emphases. Then, we introduce a multi-level concept for an experimental stage that can be used to explore interactive storytelling at varying degrees of flexibility versus predetermination.

Instead of inventing the ultimate virtual narrator, we suggest developing layers of run-time engines that allow authors to work with directions on each single layer separately—from selecting high-level dramatic structures down to directing animated actors. The concept of underlying models at each level is explained in detail by the example of a story model implementation. Further, we show variations of level adjustments causing graded degrees of semi-autonomy. First results of this experimental stage are presented, and the primary future tasks are pointed out. © 2002 Published by Elsevier Science Ltd.

1. Introduction and Motivation

1.1. New applications for storytelling

The increasing dissemination of digital applications and their number of users currently lead to new forms of content and interfaces, e.g. for education and electronic commerce. Storytelling and conversational agents, virtual characters, as well as game-like interfaces, are introduced to replace or supplement GUIs in order to provide more accessibility. For example, an e-commerce application would make use of a virtual car salesperson that holds sales conversations with a potential customer [1]. These new forms also result in market trends, such as infotainment and edutainment. Therefore, as with traditional linear media, interactive storytelling crafts are the focus of increasing interest, also from digital business application providers beyond entertainment. In the entertainment sector, the general acceptance of computer games is gradually exceeding that of the film market.

However, looking into various computer game genres exclusively, in order to adapt their methods for authoring and interaction to interactive storytelling for more general applications, shows that there is still a need for research into methods and principles that do not as yet exist. It seems impossible to define authoring tools before the working methods, that they are based on, have been realized.

In this article, we present our approach to solve the problems associated with finding new methods in interactive storytelling by building prototypes in limited areas with reduced complexity (see Sections 4 and 5). The results presented in this article are based on research done in several projects, as explained in Sections 3 and 6. The main application areas are game-like interactive storytelling for a historical information system in augmented reality, and conversational presentations at an info booth.

*Corresponding author. Tel.: +49-6151-155-229; fax: +49-6151-155-451.

E-mail address: ulrike.spierling@zgdv.de (U. Spierling).

0097-8493/02/$ - see front matter © 2002 Published by Elsevier Science Ltd.
PII: S0097-8493(01)00176-5

1.2. Simulation versus narration

Some game designers and writers envisage the future of interactive storytelling in "emergent narrative", which is a simulation of actions based on the interaction of parameterized social agents.

Authors, or rather narrators, that stem from traditional linear media have difficulty describing the parameters of an emergent narrative ad hoc. In this case, the story is supposed to emerge in a bottom-up way, based on characters' procedural behaviors that are described in a top–down parametric way by authors. But choosing a top–down approach for creating even linear stories is not what every writer would prefer—sometimes it is easier to reason by example instead of by logic and to just describe the phenomena instead of the gene pool. Even worse, the more the subject is meant to be a calculated simulation of behaviors or artificial life, the more complex is its definition in terms of programming. The question that arises here is whether authors of interactive emergent narratives will really *have to* program algorithms of storytelling.

Once a "genome" (as a set of parameters) is identified, however, one can play with it. In general, the skill of having real design choices means possessing a well-founded knowledge of the parameters that are subject to potential adjustments. An example implementation of that topic is the screenwriter's tool "Dramatica" [2]. Dramatica delivers a checklist of dramatic parameters with values to choose from; yet, it is not meant for interactive presentations, but is meant as a consistency check and for production planning of drama in linear presentations. Another example from the field of computer animation is the difference between the artistic but tedious definition of a keyframe animation of Inverse Kinematics effectors that define the specific walk of a human figure, and the simple assignment of a "walk" procedure to an encapsulated object that knows its own method of walking. The tedious work results in a maximum of artistic expression because the artist has control over every detail. The automated alternative might instead lead to results that are unpredictable—unless the author and the system master the parametric language of motion behavior.

The experiences from Dramatica, however, lead to the assumption that parameters that can control a simulation can also be used by authors to structure their work, and vice versa. In Section 4, we propose scalable autonomy in order to let authors approach simulation by experiments, starting with predefined narration.

1.3. Story creation versus story telling

Let us have a closer look at the "interactive" in interactive storytelling. How does the user interaction affect a narrative? First, we have to distinguish between *story creation* and *story telling*. Where do we want to locate "interactivity"?

Interactivity is a live experience. In an interactive story, occurrences, phenomena, and topics of discussion are subject to change in real time, depending on the interactions of at least one person—a user.

Interactive story *creation*, for example, takes place in role-playing games that can be seen as emergent narratives of multiple authorship. The creation process is interactive, because several players interact while they create a plot. For example, an interactive play about Romeo and Juliet would put users in participating roles of main characters. A user, who manages to dissuade Juliet from committing suicide, could indeed affect a different ending including a lot of different actions in the story. Here, a discussion is in progress among the game developers community that the notion of authorship should be abandoned; others argue that in fact a story does not even exist during simulation time or playing time—*if* there is a story at least, it is told afterwards. Which leads to the next point:

Interactive story *telling* instead relies on a predefined story, a specific plot concerning facts and occurrences. Only the telling of the story is done interactively. The telling of a story involves presentation means, such as speech and body language of the narrator (poses, gestures, facial expressions), as well as mediated representations of narration, such as theater, film, graphics, comics, and the like. An interactive storyteller is an artist that reacts to the audience, while maintaining suspense and entertainment value. In order to accomplish this in an algorithmic way, we would have to implement not only anthropomorphic human behavior to a high degree, such as speech and gesture synthesis, but also a talent that not all humans share.

In our approach as explained in Section 4, we assume that we need to tell facts that have to be defined beforehand, as e.g. historical or business information. This implies that our emphasis is on the story *telling* version, and again puts authors in a central position. However, due to interactivity, we need a run-time engine taking care of the user's experience as the story is being told.

1.4. The artistic role of authors

Artistic creation of an interactive story in a virtual environment is not merely about writing a story. As in game design or in film, a whole team works on the product result, each member has a certain degree of freedom in decision making. To name some examples, the roles could be:

- Editor (being in charge of the topic and the overall story content)

- Playwright (planning of the presentation, mise-en-scene, scriptwriting for dialogues)
- Director (interpreting the script, working interactively with actors through stage directions, during rehearsal or during actual shot)
- Stage director (camera, lights and set design, dramatic creation by choosing viewpoints, lighting and props)
- Casting director (definition of characters and their visual representation by actors)
- Actors (following stage directions with a certain individual scope of interpretation)

We assume that for interactive storytelling, as in the games production, an interdisciplinary team works together, and more and more of the above-mentioned tasks are taken over by algorithmic representations. Programmers who work on Game AI, such as e.g. motion physics, or rule-based cameras and lighting, will take the artistic role. However, as in film production, we believe that in order to be entertaining and to provide dramatic closure, there can only be one person responsible for the overall storytelling, which is more than the sum of its parts. One single human author or director will be in charge of the user experience and the vision, while many collaborators (human or algorithmic) have to amplify that vision. The approach mentioned in Section 4 reflects the various stages in levels and lets a single author define how dynamic a level should appear.

1.5. Consequence: the need for an experimental stage

A very basic property of most artistic fields is that the creation can be examined by the artist immediately. From one second to the next, writers can step into the role of their readers. Painters watch their pictures while they come into being. Musicians listen to their music while they are playing it. Usually, the artist can step into the role of the audience while he is performing. As a matter of fact, the artist perceives being creator and spectator at the same time as the essence of the creative activity. The joy and inspiration comes from witnessing the work during its creation.

In the case of interactive storytelling, this has been very different. Due to the technical difficulties of the field, artistic work has been preoccupied with low-level programming tasks. However, since we are dealing with *interactive* storytelling, interaction is an essential part of perceiving the work and, therefore, should be an essential part of the authoring process itself.

If we take this imperative serious, then we cannot proceed towards a system that separates the authoring process from the reception of the work. Authors need generic systems for interactive storytelling as experimentation platforms for bottom-up approaches to define interactivity. Further in the future, when a common language for stage directions has been developed, the education of authors can also be based on this.

However at the moment, we realize that we are still far away from defining the tools. Instead, we are approaching an architecture that allows for an experimental development of new methods. In this article, we present a generic framework of modules, based on experiences that have been drawn from several storytelling projects.

2. Related work

Early approaches to interactive drama emphasized the development of autonomous believable agents. This philosophy was introduced by the Oz project led by Joseph Bates at Carnegie-Mellon University [3]. The underlying assumption was that a coherent story line would emerge from the agents' autonomous behavior, given the help of infrequent interventions from the side of a drama manager [4].

Less work was done on interactive plot [5]. In recent years, however, several systems have been developed. The Erasmatron, a story engine developed and implemented by Chris Crawford, is based on a sophisticated world model. It seeks to balance character-based and plot-based approaches by using verbs as the basic components of action. Crawford does not believe in story generation through algorithms and therefore, plans for the author to create a useful set of verbs that the engine can work with [6]. When designing a story engine for the DEFACTO project, Nikitas M. Sgouros followed a different approach. Using a rule-based system, he aimed at shaping interactive plot into the structure of Aristotelian drama by modeling rising conflict between the characters [7]. Nicolas Szilas takes a similar approach [8]. Michael Mateas and Andrew Stern worked on an interactive story world. They defined beats as the basic units of plot and interaction [4].

What all these approaches have in common is a rather fine granularity of plot control, which in turn provides the user with frequent opportunities for changing the plot. A different approach suggested by Janet Murray [9] is to deal with interactive plot at a higher level—as the level of morphological functions that were defined by Russian formalist Vladimir Propp [10]. Propp's work, which was first published in 1928, is central for providing dramatic closure beyond mere action plot. Building the base for our main concept of a story model, Propp's theory is explained in detail in Section 6.

Finally, for a convincing stage presentation and interaction with users, the work that has been done in the context of embodied conversational agents, as described by Justine Cassell et al. [11], is also of high

interest, as well as animation principles that provide those agents with believable actor behavior, as developed by Ken Perlin and Athomas Goldberg [12].

Other examples from the game industry show that there are currently no generic systems for interactive storytelling that solve all problems involved. Interactive storytelling is a new topic that has many levels on which to develop new methods for authoring and interaction. We assume that the only way to progress is to create numerous prototypes and experiment with interactive storytelling on each level.

3. Project examples

The findings and the work presented in this article result from several projects that formed the groundwork for suggesting generic layers of autonomy in interactive storytelling.

Some projects include information systems that rely on a certain content that has to be talked about, and that cannot be subjected to change by interaction. Others are more user-centered and work with an agent base. In any case, the main focus is not on gaming, but on the inclusion of narrative in interactive applications that might have an entertaining aspect.

Two of the projects are presented here. They serve as examples for different aspects of interactive storytelling.

3.1. Historic storytelling in mobile augmented reality

The mission of the project "GEIST" is to develop interactive storytelling during a real city tour. The GEIST system is a mobile computer game that uses augmented reality technologies for the playful display of historical information in cities or at memorial landmarks. This technology is supposed to fulfill the age-long nostalgic desire to meet the spirits of the past in their element in the form of moving figures, gliding through the real scenery of a ruin or landscape, visible through semi-permeable glasses. Players can release a restless ghost by delving into the concerns of that time period and helping him to bring his story to a conclusion. Towards this end, they must walk around in the story's actual locality in order to meet further ghosts and explore stories from the past. Through the help of a positioning system included in the tourist's equipment, ghosts located at specific places perceive the presence of players and start to communicate with them proactively.

In GEIST, the interactivity of users mainly consists of changing the location by walking outdoors and dealing with the so-called 'magic equipment'. The system is designed to cater to a tourist's sense of satisfaction; therefore, it is unthinkable to accept the possibility that a user might get lost or may not manage to experience the denouement of the story in a given amount of time. Consequently, there is a need for adaptivity in the system that takes care of narrative functions of the story on the highest level. Narrative functions are morphological structures for the order of scenes based on the work of Vladimir Propp [10] (see also Section 6). Accordingly, the GEIST system should be able to adapt the flow of scenes to such aspects as the tourist's success, the dialogue history, the clock time, as well as the environment. It is also apparent that no historical plot (the backstory of the ghosts) is allowed to be affected by this interaction, but only the actions of ghosts in the present time, in which the user is a main character of the story.

3.2. Digital trade show booth

A different example is the design of a conversational multimodal communication with digital actors at a digital trade show booth (point of information). Here, the telling of the story is more like a conversation between two parties: the user and the virtual characters. The content of the conversation is business and product information. Since this application is more like face-to-face communication, the main emphasis was on modeling the conversational and social behavior of a character agent. The presentation had to follow a different style of narration and interaction than that used for playing with fictional ghosts. The creation of narrative functions also has to be adapted to the application domain of e-commerce, assuming that the structure of a sales conversation differs from that of a folktale.

4. Our approach: a modular system of components with scalable autonomy

4.1. Four levels of modules

By looking for a generic approach for convincing interactive storytelling systems with various peculiarities, we find that there are multiple problems to solve at different system levels: from drama structure up to staging for agents, and finally to the graphical animation of characters. We cannot expect authors to manage all the programming that is necessary to create autonomy for all these levels in one pass. Therefore, we need a multi-level platform to develop the playwriting craft by experiment, in order to test the influence of users, as well as the possibilities of influence by authors at each stage.

We propose an architecture of four hierarchical levels, shown in Fig. 1. In Section 5, a detailed description of every module in the architecture is given. At the heart of each level, there is a specific run-time engine that works on the basis of authored data and underlying models.

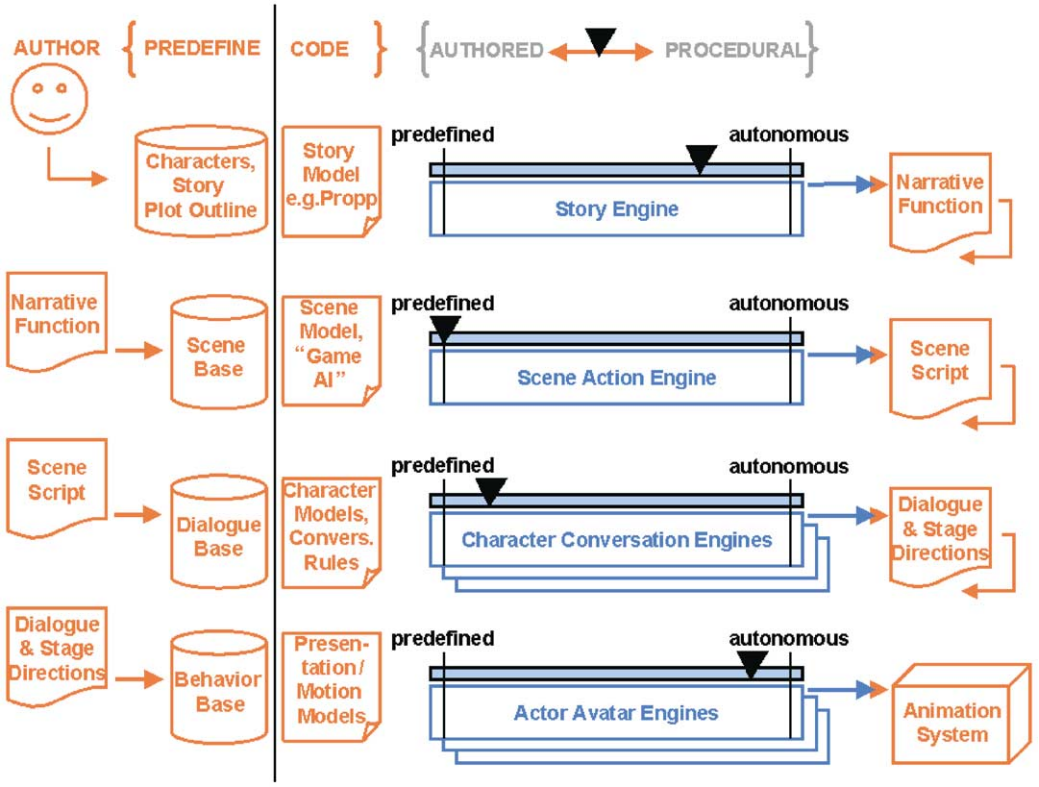

Fig. 1. Four levels of engines with scalable autonomy, relying on models and predefined data. Authors have to predefine and code with varying gradations at each level.

The output of each run-time engine is handed as input to the next lower level.

4.2. Scalable autonomy

The underlying philosophy is to provide the author with an efficient and artistically satisfying means of control at each level. Our goal is to preserve the human author's authority in story creation. We believe that the responsibility for this artistic task cannot be handed over to the machine. At the same time, we—as much as possible—need to relieve the author from any programming, modeling or design activity that can be automated. A similar approach has been suggested by Janet Murray [9].

Accordingly, each of the levels should work under a certain degree of autonomy—but this autonomy should be scalable and adjustable. Scalable autonomy means several degrees of semi-autonomy. In Fig. 1, each engine has a scale between "predefined" and "autonomous", which refers to either choices of predetermined (linear) material from a human author or procedurally generated results from the relative engine. As a result, the source for decisions and actions on every level is either a human being or an agent or a combination of both. In any case, it is the author who decides to which degree of autonomy each level works in a particular application.

According to the degree of autonomy, genres of interactive storytelling might be changing. Some examples are as follows:

- All scales are put to "predefined": the result might be a linear or a static branching film. In other words, only the author has control over every little detail that is supposed to happen on any level. This could also be a starting point for authors to experiment with the platform.
- The story engine is scaled down to act predefined, but the scene and character engines are autonomous: the result might be a simulation of autonomous agents, but the author has to take care of dramatic act structures.
- The scales are adjusted as shown in Fig. 1: this reflects our application of the "GEIST" project that is explained in Sections 3 and 6. The story engine makes dynamic choices of the given scenes that display characters showing almost autonomous

animations in real time with some conversational flexibility.

4.3. The role of the authors

In storytelling, good personal rhetoric matters. To enable audience interactivity, still more talent is needed in terms of quick-wittedness: for example, the talent of an improv comedian who reacts in intelligent ways. Phoebe Sengers points out that narratives are always particular [13]. It is in the details that the shaping influence of the author becomes visible. We do not believe that machines are able to create convincing details of narration. So, either the author has to prepare a lot of details and store them in databases (e.g. behavior base of canned animations), or deliver coded information to the engine (based on a motion model).

Besides writing scripts and code, experienced authors in different roles (e.g. writer or stage director) should also be able to access the underlying models. For example, the story model can be adapted according to the genre of stories that are to be told—e.g. either a fairy tale or a product presentation. The presentation models can vary in order to display various styles.

Since our approach is meant to present an experimental platform, inexperienced authors can begin with rudimental predefined versions as a starting point, and add more and more autonomy successively. For example, most beginning authors with no programming background currently tend to start with branching stories to apply to user choices, because the user experience is comprehensible directly. However, there is a common agreement that branching leads to uninteresting storytelling. With the experimental platform, these first experiences can lead to new authoring approaches step-by-step.

4.4. The role of the audience

What is the user's realm of influence on the various levels?

One question here is whether or not the player should step into the role of one of the characters of the plot. He could be the protagonist, or a contagonist, or play some other role. Also, he might be allowed to switch perspectives. An alternative approach would be to let the player watch the story from an outside perspective. These alternatives influence at different levels.

The modular system allows us to test what the user can influence at various levels; for example, the order of scenes (choice of different ways to go), a change of plot or of scene variables (influence at scene level, e.g. lose or win a fight), or details in conversational behavior of agents (on the conversation level).

5. Architecture of components with scalable autonomy

The architecture consists of four engine layers (see Fig. 1). These are run-time components that take care of the story presentation at different levels of abstraction. The engines make use of a variety of models, which can be predefined for certain genres or soap styles. These models can also be edited by experienced authors in order to guide the presentation according to their expectations or create different rules for different styles.

5.1. Engines

5.1.1. Story engine

The task of the story engine is to coordinate the flow of the story at its highest structural level. In a linear scenario, the author decides on the strict order in which the different acts or scenes of the play have to appear. On the other hand, if the engine is given some autonomy, it can dynamically choose which scene to play next. As a basis for this decision, the engine uses a story model. Rather than choosing a specific scene, the engine could decide on the narrative function that the next scene has to fulfill. This function is then handed over to the scene-action engine, which has to come up with a scene that fulfills this function. If that is not possible, then the story engine consults the story model again in order to suggest an alternative narrative function to the scene engine.

5.1.2. Scene action engine

The scene action engine plays scenes that fulfill the narrative function that the story engine has selected. In a scenario with low autonomy, the scene engine performs choices from static descriptions (scripts) of scenes that are stored in a scene database. These static scripts are written by human authors, and include details such as the setting, user's viewpoints, the characters on stage and their actions. In a more dynamic scenario, the scripts are generated by the scene engine. The choice of the setting and the characters, as well as of the actions carried out, are based on a scene model that consists of dramatic functions, or e.g. on rules currently used in Game AI.

In the end, it translates the scene result, which is also based on user interactions with the underlying characters, into semantic abstractions, the so-called story acts that are handed back to the story engine. Thus, the story acts of the user have a significant influence on further development of the plot.

Emergent narrative not only fails to develop narrative structures with sufficient complexity, but also presents a lot of details that is not relevant for the story. In contrast, the scene engine is supposed to guide and filter

interactive drama at action level according to dramatic functions of each action.

5.1.3. Character conversation engines

The character conversation engines ascertain that the story characters act according to their personality traits, social roles and relationships, as defined in the character model. On a very low level of autonomy, the character engine can be thought of as a storage place for predefined dialogue scripts, while on the maximum level, the character is an intelligent agent that makes its own decisions. However, our goal concerning the involvement of agent technology is to allow a maximum of behavior control by human authors. Looking at the state of the art in the field, preferring narrative control over agent autonomy reflects a shift in paradigm that can be seen in the work of Nicolas Szilas [8], and Michael Mateas and Andrew Stern [4], as well.

As a part of the character engine architecture, a conversation engine allows the agent to take part in dialogues with other characters and the user by synchronizing turn-taking. Therefore, the conversation engine plays a vital role as it supports user interaction at a comparatively high conceptual level. Its results are transferred in the form of stage directions to the next lower level, the actor avatar.

5.1.4. Actor avatar engines

The actor avatar engines are output services for animation, speech and sound. Each autonomous actor avatar engine should take into account the personality of its character as defined in the character model. Further, it follows the dialogue and stage directions that are handed down by the character conversation engine.

The criterion for scaling up the autonomy on this level is the ability of the engine to adapt the display to the context. E.g., an autonomous actor avatar is able to follow abstract stage directions to move around, relying on some representation of space, and its movements could be parameterized according to the character's mood. The ability to generate visual speech (lip sync and speech accompanying movements) should be present even in less autonomous avatars. A non-autonomous, dependent actor engine would simply play stored animations.

5.2. Models

5.2.1. Story model

This model contains rules that describe the flow of the story. Since it is a separate module, it can be easily adapted to meet the requirements of different applications. We are currently using a model that is based on the work of Vladimir Propp [10], consisting of morphological structures.

The author should be enabled to adapt the story model according to the set of stories that are to be told. Different models can be used depending on the scenario of the application. The story model consists of a meta-program, as well as a set of rules. The meta-program directs the flow of the story. Rules model the inter-dependencies between different scenes.

5.2.2. Scene model

This can also be described as drama model. Similar to the story model, the scene model contains dramatic functions at the level of concrete actions and settings. Further, it contains rules based on dramatic laws, that are similar to those used for single shots in film, as summarized, for example, in [14]. They can also be taken from solutions in Game AI. Analogous to the story model, rules here model the interdependencies between different single shown actions, as well as their function for the scene result.

5.2.3. Character models

These models will allow the character conversation engines to ensure that the interactive narrative stays consistent in terms of motivations and personality traits of the figures involved. This is necessary, since the story model cannot take the personality and motivation of each agent adequately into account.

Personality parameters, for example (compare Fig. 3), have an influence on

- what the character feels in a certain situation,
- how he appraises an event,
- his values and norms,
- his goals,
- his affective relations, and
- his decisions.

Possible example dimensions of personality are extraversion, tenderness or intelligence.

5.2.4. Conversation rules

The conversation engine, which is included in each character engine, will make use of a conversation model. Here, the author can specify rules that will direct the discourse between characters according to his intentions. Different conversation rules apply to different characters. For example, an aggressive dog has its specific way of engaging in a dialogue with intruders to its territory—very different from the fair hostess at a trade show booth.

The conversation model contains conversational aspects between human–human communication, which are applied to the interaction between agents, as well as between agents and users. This is done in a generic and amodal way (not related to the media used to perform a conversation or dialogue). This abstract information is a

means for authors to influence the manner in which the system relays lines to the user. Depending on the genre, this could also serve as a user interface. The amodal conversation instruction is then handed over to the avatar engine. This way, the storyteller can use conversation engine-driven actors in an interactive setting in much the same way as a playwright or a director can influence the direction of interaction between actors in a play.

5.2.5. *Presentation/motion models*

First of all, the presentation model comprises the geometries and textures used for display. It is extended by models about typical movements, speech characteristics, or about the proper usage of signs. Task accomplishing procedures like "sit down" are also a part of this model. Authoring tools empower the artist with control not only over static features, but also over the expressiveness of movements and signs. They include stage directions that can be used to direct the virtual actors.

6. Realization of modules

This section will provide a more detailed study of concrete implementations that have been carried out in the projects mentioned in Section 3: "Historic storytelling in mobile augmented reality" (GEIST), and the "Digital Trade Show Booth" application. It can be noted again that depending on the application details, we made different shifts in emphasizing the selected autonomy at different levels. The explanation focuses on the description of the particular modules.

For GEIST, pursuant to the type of user interaction (walking through a real scenario), the story model for plot planning on the highest level is of main interest and is explained in detail. For the Digital Trade Show Booth, the user interaction is more conversation-like and, therefore, a main emphasis was placed on the conversational aspects of characters. Both projects make use of the same actor avatar engine that is responsible for the dynamic presentation of characters. For the scene engine, there is currently no implementation; both projects rely on pre-authored dialogues.

6.1. *Story engine and story model*

Due to its rapid prototyping capabilities, we have implemented the story engine in Prolog [15]. A detailed description of the system is given in [16].

For the GEIST project, we have chosen to implement a story model based on Vladimir Propp's "Morphology of the Folktale" [10], which was written in 1928. As the basis of his study, Propp used an arbitrary sample of 100 Russian fairy tales. The narrative structures of those folktales can be easily adapted to the narration of the ghost story that forms the background of our scenario. The application of Propp's morphological approach to interactive storytelling was suggested and sketched by Janet Murray [9].

Propp's classification of fairy tales is continuously aware of the storyteller's branching possibilities between morphological variants of narrative elements. Propp defines function as follows: "Function must be taken as an act of dramatis personae, which is defined from the point of view of its significance for the course of action of a tale as a whole" [10].

Thus, functions are independent of the specific characters, as well as of the particular actions that perform them. For his classification of morphological functions, Propp developed a notation system that uses capital letters. His functions can be easily applied to the context of our GEIST project. For example, there is a function in which the hero departs from home. In GEIST, the tourist takes over the role of the hero. Since he sets out on a journey through the old town of Heidelberg, he is departing from home and thus fulfilling that narrative function. Another example would be the sequence of narrative functions denoted by the capital letters D, E, and F. In D, a potential donor involves the hero in a test of some kind. Our scenario provides us with a variety of opportunities to introduce ghosts that involve the player in different tasks. By dealing with the situation, the user fulfills the next narrative function: the reaction of the hero to the test (E). Depending on the reaction, the hero may or may not be given a magical agent (F).

Propp's analysis revealed that the number of functions to be encountered in a large body of narratives is rather limited. Furthermore, they tend to appear in one and the same order in the fairy tales examined. Stories differentiate themselves in regard to which functions have been left out. Many functions exist in different variants that are in part correlated with variants of other functions. Functions themselves often appear in pairs. For example, any misfortune caused by villainy should be followed by a liquidation of that misfortune. Propp suggested grouping functions according to the dramatis personae that perform them. By *dramatis personae*, he was referring to different types of roles in the abstract plot, and not to specific characters. Among those types, he counted the villain, the donor, the helper, the person sought after and her father, the dispatcher, the false hero and the hero. Most of these types could take the form of any particular character, a group of characters, an animal or even a thing, depending on the morphological variant. Propp also discovered that several narrative sequences (in his terminology, '*moves*') could follow each other or even overlap.

We have chosen to implement Propp's system for a variety of reasons. Bride Linane-Mallon and Brian

Webb argue that stories cannot be divided into discrete modules, since their elements are highly interrelated [17]. Phoebe Sengers points out that the cause and effect of narrative events are more important than the events themselves. We believe that Propp has solved these problems by defining functions in the context of the surrounding story. Also, the narrative functions of fairy tales provide a perfect match for our ghost story scenario GEIST. However, there is a reason to believe that Propp's functions can be applied to different genres, as well. For example, very similar narrative structures can be found in modern tales like Star Wars. Arthur Berger points out that Propp's functions constitute the core of all narratives to this day [18].

6.1.1. Functions and scenes

The GEIST system works with two levels of abstraction. At the upper level, a sequence of functions is selected in real-time. We use the term *function* in the same way as Propp does, as an abstraction from characters, setting and action while emphasizing the function's role in the surrounding plot. User interaction and other constraints can result in functions being skipped, as well as in the choice of a different variant of a specific function.

At the lower level, a particular scene is executed for each function. At present, these scenes have to be authored in advance. The author creates linear scripts for each scene that are stored in a scene database. Each script is annotated with the morphological function it has for the remaining story, as well as its shortest and longest possible duration, context requirements, setting, characters, levels of violence, etc.

We see functions as classes, and scenes as instances of those classes. A scene possesses unity of time, space and action, thus facilitating temporal, spatial and emotional immersion of the user. Between one scene and the next, leaps in time are possible and even desirable, since we do not want to bother the player with less exciting events.

User interaction cannot change the morphological function of most scenes after their beginning. We have implemented a few polymorphic functions, however. Our conception of polymorphic functions was inspired by the notion of polymorphic beats, introduced by Mateas and Stern at a different level of detail. They refer to the most basic elements of action, which can have different outcomes depending on user interaction [4]. We are referring to scenes composed of a number of actions at a specific location. When the user enters a scene that corresponds to a polymorphic function, its outcome is not yet determined. It is important to note that by *outcome,* we do not mean the content of the scene (which we allow to be flexible in the case of non-polymorphic functions, as well), but rather its function for the overall story. Thus, user interaction and other dynamic factors influence the morphological variant chosen for the scene after its execution.

For example, we do not know how the player will perform in a test or riddle in GEIST. Therefore, this function (denoted by "E" in Propp's system) is polymorphic, since its outcome is not yet determined at the beginning of such a scene.

We have implemented polymorphic functions for the outcome of attempts of villainy and for the reaction of the hero to riddles and tests. User interaction decides if these functions succeed or fail, which has direct consequences for the remaining plot. If we need these functions to succeed (as in the case of villainy), we either repeat them with different scenes or choose a non-polymorphic variant with the desired outcome.

6.1.2. Function selection

Regarding the selection of the next function, Propp gave an informal description of an algorithm which we have implemented: "It is possible to artificially create new plots of an unlimited number. All of these plots will reflect a basic scheme, while they themselves may not resemble one another. In order to create a folktale artificially, one may take any A, one of the possible B's, then a C, followed by absolutely any D, then an E, then one of the possible F's, then any G, and so on. Here, any elements may be dropped (apart, actually, from A or a), or repeated three times, or repeated in various aspects [10]".

Fig. 2 illustrates this algorithm. The section is traversed from left to right, while each square represents a specific scene. The scenes depicted in the diagram fulfill the functions A, D, E and F. It is important to note that the connections between them are not fixed but rule-based. Function E is polymorphic—user interaction decides if this function succeeds or fails. Depending on the outcome, different variants of function F are selected. If the player succeeds in the test, he is given a magical agent (F(+)). If he fails, the reaction of the donor is negative (F(−)). In the latter case, the sequence of functions D, E and F is repeated.

However, there are additional criteria for choosing the next function. The user chooses a certain time frame for the overall experience that has to be met. If the player's pace of interaction is very slow, then many functions will be skipped. User timing, therefore, has an indirect effect on the remaining plot. Before selecting a function, the engine checks whether the corresponding scene can be played in the remaining time. If that function requires other functions to follow, then the durations of their corresponding scenes have to be taken into account, as well.

If there is enough time remaining, a second *move* (Propp's term for a self-contained sequence of action) could be started. Before doing so, the engine has to make sure that there are enough scenes available for the

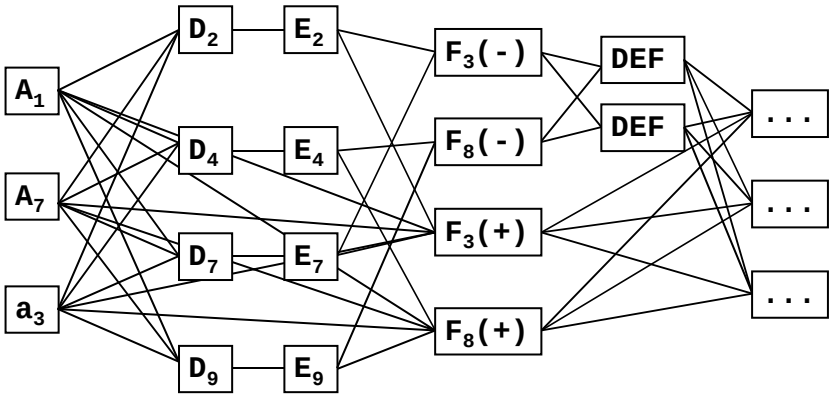

Fig. 2. A section of the story space.

additional plot without duplicating or contradicting previous events.

6.1.3. Context-dependent scene selection

After the story engine has decided on the next morphological function, the scene action engine has to carry it out. Since several scenes might fulfill the desired function, additional selection criteria have to be taken into account.

Out of these criteria, the current context of the story is most important. Currently, the author encodes a list of context requirements with each scene description. Scenes can create new context (i.e. if they introduce characters to the story). They require a context to be present (i.e. a certain type of misfortune) and but they can also remove context (i.e. if that misfortune is liquidated). Context-dependent scene selection ensures that the story told by the GEIST system stays consistent with the previous events. Before introducing new context to the story, the system checks if this context can be correctly resolved later on. Another criterion is the location of the user in physical space. In GEIST, the tourist is allowed to navigate freely through the old town of Heidelberg. Therefore, the setting of the selected scene has to match with the surroundings of the player.

If the system is still left with a set of choices after processing these selection criteria, then a random decision is made.

6.2. Character engine and character model

In the Digital Trade Show Booth application, the agent is responsible for involving the user in varying entertaining and social constellations. The deliberative capacities of the agent are implemented in a BDI-style architecture (using a symbolic representation of belief, desires, and intentions, cf. [19]). Part of the processing is done with Jess. Due to their personality traits (compare [20–22]), the agents can react differently to user input and to the contribution of other agents. For example, the sympathy of one certain agent for the user will decrease if he should show disbelief, leading even to the aggressive behavior of the agent, if so tuned by the author.

For the character model, we followed the path shown in e.g. [23] and [24] by choosing a subset of the emotions eliciting conditions described in OCC (the work of Ortony, Clore, and Collins, (cf. [25]) to drive the emotional reactions of our agent prototype. Additional features lead to personality-dependent appraisal strategies and social attitudes of the agents. The ''Big Five'' of personality theory—Surgency, Agreeableness, Conscientiousness, Emotional Stability, and Openness—(cf. [26]) combined with other parameters describing e.g. intelligence or the temporal behavior of the agents' arousal level allow for a concise authoring process. Fig. 3 shows the experimental interface for authors. The author here has access to the character model's parameters that are connected to an avatar engine. Authors are able to tune characters by experimenting with their respective visible behaviors.

6.2.1. Conversation model

Since, in GEIST, the behavior of characters is mostly directly controlled by predefined scene scripts, the task of a character conversation engine is basically restricted to the control of pro-active behavior, as a reaction to users arriving on a stage in the real environment. In the current implementation, there is no output of this module to the higher layers of control and no influence on the deployment of the story is possible.

The types of user interaction at the Digital Trade Show Booth are suited better for exploring the conversation model. Here, the modeling of conversations on a high abstraction level is used to control animation layers that enrich the spoken text by adding

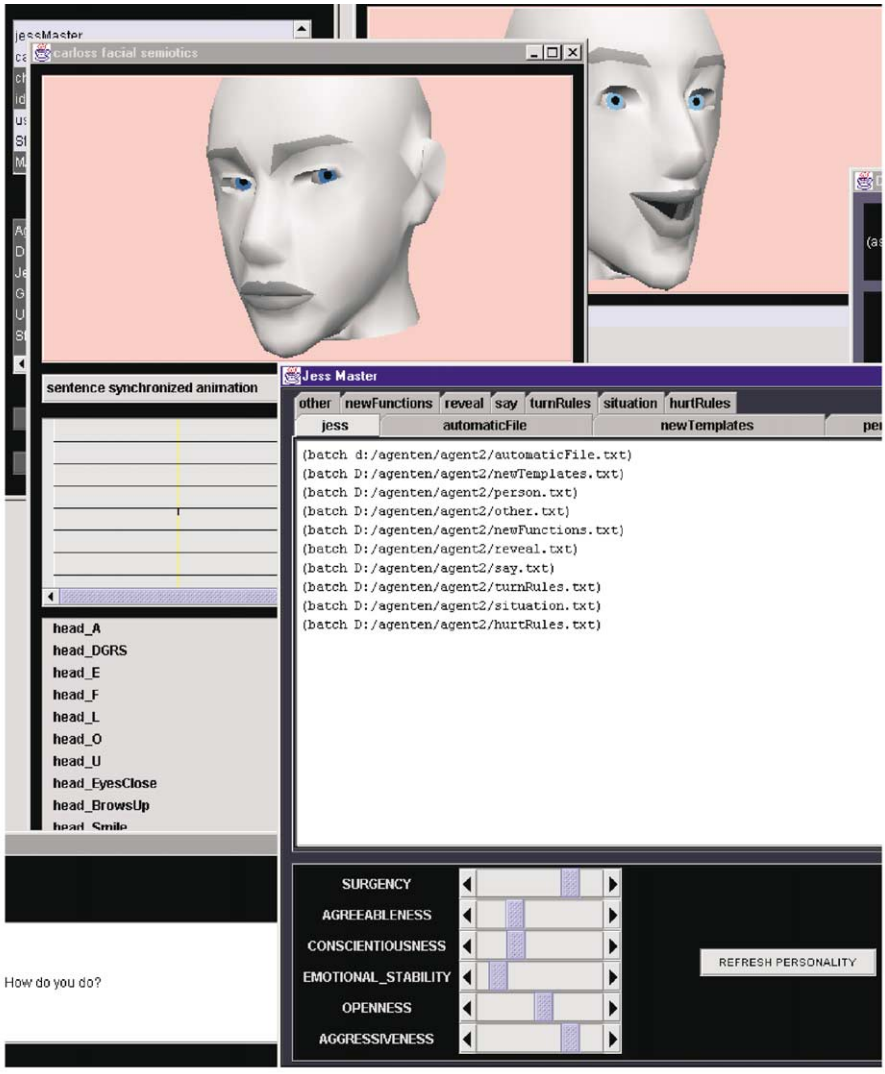

Fig. 3. Experimental stage for tuning parameters of the character model to control its avatar presentation.

more modalities (facial expressions, gestures, and emotions). In speech conversation systems, a higher level of abstraction is typically reached by defining the so-called dialogue acts and performatives [27]. Following this method, we describe more aspects of conversations that are suited to authors of narrative interactivity. They are especially related to the continuous character of a conversation, as well as to the social/emotional interactions.

Some examples for basic conversational aspects are:

- Conversation-participant (describes a participant of the conversation—the user or an automated actor)
- Conversation-element (a behavior that is carried out from a sender towards a recipient, e.g. Opener, Talk, Listen, PutTurn, GetTurn, GetAttention, Close)
- Content (Content element with its type, e.g. Question, Answer, Term)
- Story (The relation between content elements contained in a set, e.g. Sequence, Asynchronous)
- Time (Symbolic points in time of single beats, e.g. TimeslotList, ActualTime, NewTime. This allows us to model the conversation as a continuous process through a period of time)

The definition of these conversational aspects leads to the authoring of conversational rules that define how they change in relation to the following points.

- Objective of the story system (information, navigation, discussion, moderation, comments).

- Personality of individual conversation participants.
- Context and user knowledge.

More details are found in [28].

In the Digital Trade Show system, a basic set of rules is implemented that describe discourse-related behavior within rules for *passive opening and closure of a discourse*, *pro-active opening/closing of a discourse*, as well as a *change of discourse*.

The prototype of the conversation engine is written as a Java program using a Jess [29] module as inference engine for conversational reasoning. Jess has the advantage of being suited for prototyping due to its flexibility and ability to be integrated in Java. The conversation modeling is therefore performed as an expert (x) system that works like a round-based finite state automaton (FSA). The finite state automata approach are standard in game industry [30]. The advantage of FSA is in leaving out the expensive search trees with regard to making the system real-time capable.

User inputs are parsed by the user input interpreter engine for conversational aspects (e.g. Opening/closing, Turn-taking). These aspects are passed to the respective conversation engine to allow further conversational processing. A sketch of the user–actor cycle in this architecture is shown in Fig. 4.

The conversational intentions of the actors are driven by the amodal output of the related conversation engine. By giving stage directions to the actor avatar engine, the conversation engine takes care of choosing an appropriate avatar behavior. The avatar visualizes the general conversational aspects such as turn-taking, discourse changing, or individual ways to show 'talking' and 'listening' by interpreting them as stage directions.

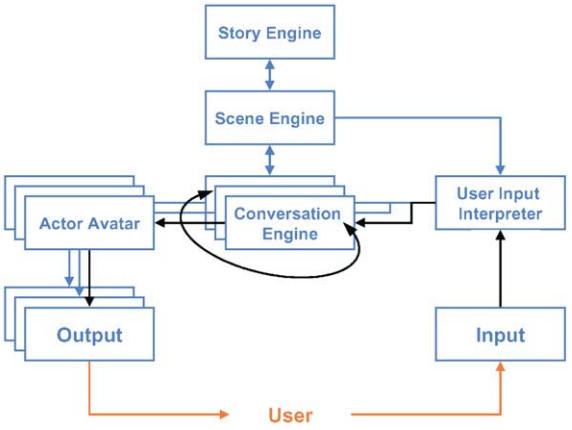

Fig. 4. The relations between story, scene, conversation, actor and user.

6.3. Actor avatar engine and motion model

For both projects, an autonomous avatar engine is being developed that can act by performing animations with a personal scope of interpretation. Currently, automatic animation generation is implemented for facial expressions and deictic hand gestures. The facial animations are driven by scripts with dialogue text and stage directions produced by either an author (as in GEIST) or a conversation engine (as for the Digital Trade Show Booth). A stage direction can be an annotation of the dialogue concerning the mood of expression, or a conversational aspect as described in Section 6.2.1.

In GEIST, a script invokes the display of predefined animation sequences. These are adapted to reflect the current position of the user, and some smoothing is done when concatenating different sequences.

In the Digital Trade Show Booth application, the actor avatar engine receives a much more fine-grained input than in the other cases. The face of the avatar depicts the simulated affective processes of the agent. Coordination of movements and the position of head and eyes are managed here, as well as every visual–speech-related task like lip synchronization and the use of batons. Conversational regulators and emblems are adapted "on the fly", following instructions and state reports of the agent. (See [31] for more details.)

In both cases, development of the platform and the underlying motion model was done in Java and Java3D, using morphing techniques for the face and combined methods for the hands. In the near future, we will also make use of OpenSG [32]. See the left side of Fig. 3 to identify morph targets of the underlying motion model for the face, and Fig. 5 for an example of a textured animated result.

7. Conclusion and future work

We present a modular system approach for interactive storytelling in order to be able to manage complexity at selected levels for authoring and user interaction. On top, we strive for an experimental stage that gives authors access to those levels and provides them a start with linear traditional methods, and increase the amount of the agency step-by-step at advanced stages. In almost the same manner, we use the same concept to successively develop software platforms at each level that provide more and more agency and adaptability to user interactions. In general, we follow an authoring-centric view of storytelling.

We have shown the extent to which we made first implementations for procedural storytelling tools in two example projects. Their main emphasis is on the levels of the story engine, the actor avatar engine and the

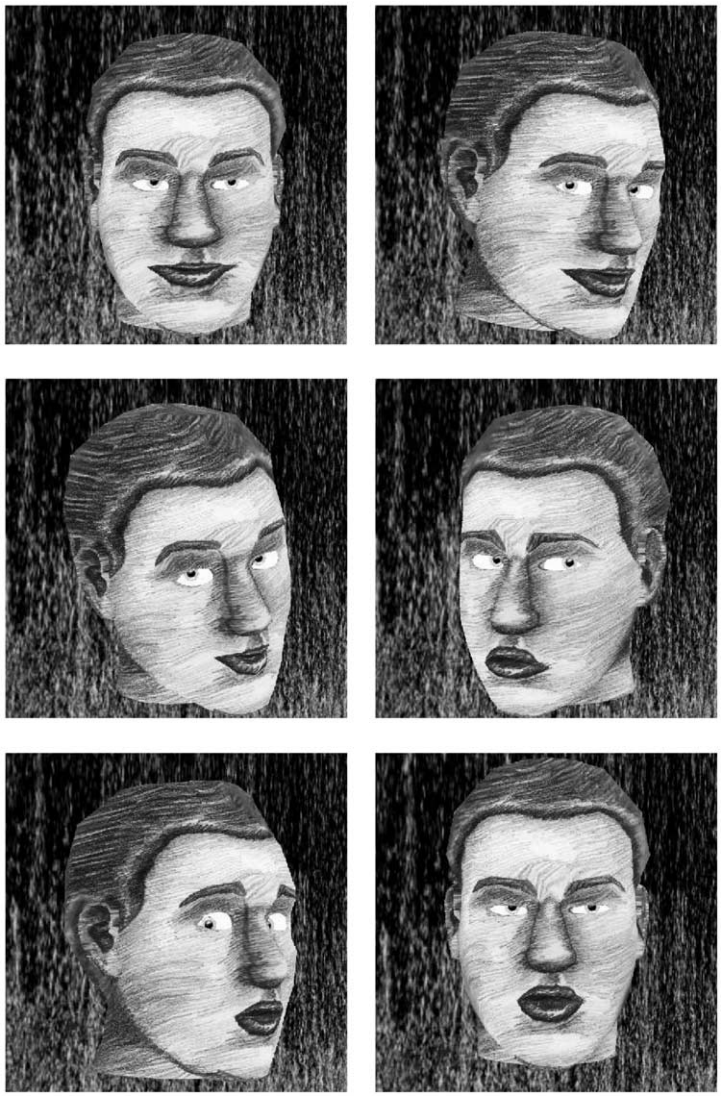

Fig. 5. Resulting image sequence according to the presentation model, including texturing and morphing of facial expressions.

character agent with a conversational behavior. Authoring experiences with these prototypes show that methods to address autonomy have to be developed for non-programmers, and that our prototypes can serve as experimental platforms. The distinction of several layers is an advantage in order to be able to focus on specific problems and reduce complexity.

Currently, we are testing and refining our concepts in the projects described above. Our next goal will be to design a scene action engine that is able to dynamically generate scripts based on a drama model that allows for rich user interaction and non-linear storytelling. In regard to the character conversation engine, we would like to support dialogues between several characters, as well as the user. In the long term, we will draw conclusions from our experiences and build accessible tools by developing authoring environments that provide intuitive access for authors who have artistic backgrounds.

References

[1] http://www.extempo.com/
[2] http://www.dramatica.com/
[3] Mateas M. An Oz-Centric review of interactive drama and believable agents. In: Wooldridge M, Veloso M, editors. AI Today: Recent Trends and Developments, Lecture Notes in AI 1600. Berlin and New York: Springer, 1999.

[4] Mateas M, Stern A. Towards integrating plot and character for interactive drama. In: Dautenhahn K, editor, Proceedings of the AAAI Fall Symposium: Socially Intelligent Agents: The Human in the Loop, Technical Report FS-00-04, Menlo Park, CA: AAAI Press, 2000. p. 113–8.

[5] Mateas M, Sengers P. Narrative intelligence, In: Mateas M, Sengers P, editors. Proceedings of the AAAI Fall Symposium: Narrative Intelligence, Technical Report FS-99-01, Menlo Park, CA: AAAI Press, 1999. p. 1–10.

[6] Crawford C. Assumptions underlying the Erasmatron interactive storytelling engine, In: Mateas M, Sengers P, editors. Proceedings of the AAAI Fall Symposium: Narrative Intelligence, Technical Report FS-99-01, Menlo Park, CA: AAAI Press, 1999. p. 112–4.

[7] Sgouros NM. Dynamic generation, management and resolution of interactive plots. Artificial Intelligence 1999;107(1):29–62.

[8] Szilas N. Interactive drama on computer: beyond linear narrative, In: Mateas M, Sengers P, editors. Proceedings of the AAAI Fall Symposium: Narrative Intelligence, Technical Report FS-99-01, Menlo Park, CA: AAAI Press, 1999. p. 150–6.

[9] Murray J. Hamlet on the holodeck: the future of narrative in cyberspace. Cambridge, MA: MIT Press, 1998.

[10] Propp V. Morphology of the folktale, International Journal of American Linguistics, Part III. 1958;24(4): 1–106.

[11] Cassell J, Prevost S, Sullivan J, Churchill E. Embodied conversational agents. Cambridge, MA: MIT Press, 2000.

[12] Perlin K, Goldberg A. Improv: a system for scripting interactive actors in virtual worlds. Computer Graphics 1996;24(3):205–16.

[13] Sengers P. Narrative intelligence, In: Dautenhahn K, editor. Human Cognition and Social Agent Technology, Advances in Consciousness Series, Philadelphia, PA: John Benjamins Publishing Company, 2000.

[14] Arijon D. Grammar of the film language. Boston: Focal Press, 1976.

[15] Grasbon D, Braun N. A Morphological approach to interactive storytelling, In: Fleischmann M, Strauss W, editors. Proceedings: cast01//living in mixed realities. Special issue of netzspannung.org/journal, The magazine for media production and inter-media research, 2001.

[16] Grasbon D. Konzeption und prototypische Implementation einer Storyengine: Dynamisch-reaktives System zum Erzählen nichtlinear-interaktiver Geschichten bei größtmöglicher Spannung, gedanklicher Immersion, Identifikation und Motivation des Spielers. Diploma thesis, Technical University of Darmstadt, 2001.

[17] Linane-mallon B, Webb B. Evaluating Narrative in Multimedia. DSV-IS97, Fourth International Eurographics Workshop, Granada, Spain, 4–6 June, 1997. p. 83–98.

[18] Berger AA. Arthurs Computer- (Geschichten-) Abenteuer, in Televizion 13/2000/1, Internationales Zentralinstitut für das Jugend- und Bildungsfernsehen, München, 2000. p. 39–47.

[19] Rao AS, Georgeff MP. BDI Agents—from theory to practice, In: Vic Lesser, editor. Proceedings ICMAS95, Cambridge, MA: MIT Press, 1995. p. 312–9.

[20] Maes P. Artificial Life Meets Entertainment: lifelike autonomous agents. Communications of the ACM 1995;38(11):108–14.

[21] Rousseau D. Personality in computer characters, in working notes of the AAAI-96 workshop on AI/ALife. Menlo Park, CA: AAAI Press, 1996.

[22] André E, Rist T, Van Mulken S, Klesen M, Baldes S. The automated design of believable dialogue for animated presentation teams. In: Cassell J, Prevost S, Sullivan J, Churchill E, editors. Embodied Conversational Agents. Cambridge, MA: The MIT Press, 2000. p. 220–55.

[23] Elliot C. The Affective reasoner: a process model of emotions in a multi-agent system. Ph.D. thesis, Northwestern University, The Institute for Learning Sciences, Technical Report No. 32, May 1992.

[24] Bates J, Loyall AB, Reilly WS. An architecture for action, emotion, and social behavior, Technical Report CMU-CS-92-144, School of Computer Science, Carnegie Mellon University, 1992.

[25] Ortony A, Clore G, Collins A. The cognitive structure of emotions. Cambridge: Cambridge University Press, 1988.

[26] Pervin LA, editor. Handbook of personality: theory and research, New York: Guilford, 1990. p. 609–37.

[27] Mengel A, Dybkjaer L, Garrido JM, Heid U, Klein M, Pirrelli V, Poesio M, Quazza S, Schiffrin A, Soria C. Mate Dialogue Annotation Guidelines. Report, in Telematics Project LE4-8370, Germany, January 2000.

[28] Braun N, Schneider O. Conversation modeling as an abstract user interface component. In: Proceedings of GI Workshop Synergien zwischen Virtueller Realität und Computerspielen: Anforderungen, Design, Technologien, Vienna, September 2001.

[29] Friedman-Hill EJ. Jess, The Java Expert System Shell, in SAND98-8206 (revised), Livermore, CA: Sandia National Laboratories, 2001.

[30] Woodcock S. Game AI: The State of the Industry, in Game Developer Magazine, August 2001.

[31] Müller W, Spierling U, Alexa M, Iurgel I. Design Issues for Conversational User Interfaces: Animating and Controlling 3D Faces. In: Proceedings of Avatars 2000, Lausanne, 2000.

[32] http://www.opensg.org/