alVRed
Methods and Tools for Storytelling in Virtual Environments

S. Beckhaus

Fraunhofer IMK
Schloss Birlinghoven
D-53754 Sankt Augustin

A. Lechner

vertigo-systems GmbH
Engelbertstr. 30
D-50674 Cologne

S. Mostafawy

rmh - new media gmbh
Gilbachstr. 29a
D-50672 Cologne

G. Trogemann, R. Wages

Laboratory for Mixed Realities
Am Coloneum 1
D-50829 Cologne

Abstract

In this paper, we motivate and present methods and tools we have developed to support the efficient creation of interactive, complex, non-linear content for storytelling in immersive virtual environments. We propose a hierarchical and parallel composition of the abstract story structure by the authors themselves to ensure the realisation of their vision of the VR scenarios in an unaltered form. The presented tools are namely the VR Authoring Tool, the VR Previewer, the Avango™ StoryEngine, the ObjectFactory and the VR Tuner. With these tools, the formerly unstructured work-flow in the production process of virtual environment applications, including the contributions of customers, authors, modellers, programmers, and others, was significantly better organised by parallelising interdependent production tasks and providing a uniform platform to which all participants contribute and from which they receive information for their tasks. With the realisation of the demonstrator production 'Entropy' our concepts and implementation are validated. This production already showed that the tool set significantly improves the efficiency of the production process and the reliability of the implementation.

1. Introduction

Virtual environments (VE) have become more and more sophisticated. Current hardware and software enables virtual scenarios with high visual complexity as well as acoustic, haptic and even olfactory content. Applications range from scientific visualisation to marketing and entertainment. Beyond the mere fascination of 3D-immersive installations with a high visual quality, there is now a need to create *content* which is of attractive story, high degree of interaction, and which enables a new quality of experience in VEs.

The challenges to provide for this are twofold. Firstly, there is a need for methods and tools for storytelling in VEs, both to support the creative process of sketching a complex, interactive story as well as to identify suitable storytelling metaphors. Secondly, the efficient creation of content has to be supported. Here, authors and developers of applications require an environment which helps to improve their communication and provides for an iterative and

integrated application development cycle. The alVRed project aims to provide for tools and methods to meet these requirements and significantly improve the complexity and efficiency of VE productions.

In this paper we first reflect on storytelling in VEs and on authoring for traditional media productions like film. Then, we propose a system for authoring non-linear stories for VEs and explain the VE production process. In section 3 we present several tools we have developed to support the creation of interactive, non-linear content in VEs. These are namely the Authoring Tool, the Previewer, the Avango™ StoryEngine, the ObjectFactory and the Tuner. These tools were used to create the non-linear, interactive scenario 'Entropy', our experiences are discussed in section 4. We then conclude and indicate important future work in section 5.

1.1 Classical Authoring

In the area of classic image based media like television or movies it is still the exception that the creative person employs the aid of authoring systems. In creating TV series such as soaps or sitcoms, however, systems are already in use which greatly help the author with regard to the development and history of the individual characters. In the case that a number of authors share the writing for these screenplays, such tools help to prevent inconsistencies that otherwise would be hard to avoid in a story plot which often extends over many weeks and episodes. In most cases, however, authors create their screenplays in the many decades old traditional ways.

Disregarding the initial sketch of the idea, the author usually proceeds in three well defined steps. The first step is the exposé, the second step includes outline and treatment and the third step the final script on which the making of the film is based. This is called storyboarding.

The exposé describes the plot on a maximum of one typewritten page without information regarding cinematic realisation. This brief form gives all participants – usually the clients or producers – the chance to evaluate the presented material. If the discussion and evaluation leads to positive results the next step is taken and an outline created. This outline is a detailed description of the story. The characters, locations and timing of the scenes are established. The result is a sort of construction plan for the film script. The actions and reactions of the actors are described in detail. The subsequent treatment of the outline is an essential help in preparing for the third step, creating the narrative of the final script. In the case of documentaries this third step will often not be necessary. However, any dramatic plot will always call for an explicit film script. The essential parts of this script are the dialogs of the actors. It also contains any details that might be dramaturgically important such as specific camera shots, notes regarding props and set design or instructions regarding post-processing.

1.2 Authoring of Non-Linear Stories in Virtual Environments

The best approach to interactive stories or interactive drama – plot-driven and character-driven – has yet to be identified. From our experience, current VE installations for the most part are created as presentations in (science) museums or on industry fairs which strongly focus on facts to be communicated. Here, characters as an integral part of a classical drama still play a minor role. Yet, creating an interactive VE with a non-linear plot addresses the difficulty of guaranteeing a proper dramaturgical timing along the various possible plots as well as keeping the complex story structure manageable in the first place. Our approach to stories and VE productions can be regarded as plot oriented.

Authors who have never worked on developing stories in an interactive, non-linear dramaturgy will quickly find that the usual strategies and tools fail in the task of developing VE productions. Instinctively, an author will try to shape the story line with a classic linear dramaturgical flow. The fact that the user is always in a way the co-author at the author's side is at first quite irritating. How can one anticipate all possible ways of interacting with the story a user comes up with? Nevertheless, for all of these cases the occurrence of for example illogical short-circuits should be prevented.

Authors being VE novices can apply a simple trick to keep their grip on the story, but the result is a somewhat fake interactivity. The principle is that certain junctions are built into the actually linear structure of the story. At these junctions the user can interact with the story. The path through the story is, thereby, clearly depicted which greatly simplifies the tasks of the modelling and programming crew right from the start. The price paid is that the possibilities for interactions are limited – this is, of course, a direct contradiction to the basic idea of VEs.

We use Avango™ in the alVRed project as VE development platform. Avango™ is a software framework for complex and distributed VEs. It is available for SGI Irix and Linux. With its underlying scene graph and field-container design, acoustic, haptic and visual rendering capabilities, with embedded scripting support (via the Elk/Scheme language), and its many extensions, Avango™ has evolved to a powerful system. It is built to efficiently support the development of interactive immersive VE applications for scientific visualisation, data exploration, entertainment and many more areas. For complex non-linear storytelling, the disadvantage of current VE frameworks like Avango™ is that they do not include a high level authoring tool for the story. The story logic has to be implemented by a programmer. The more complex the story the more tedious, time consuming, and error-prone is the implementation. Extending a VE framework like Avango™ with a powerful hierarchical authoring tool for the story logic greatly enhances the process of creating interactive non-linear stories in VEs.

1.3 Support for Virtual Environment Productions

Besides supporting the author in the process of non-linear storytelling, there is also a need to improve the production process of VE applications. A typical production scenario is subject to significant time constraints, as larger groups of individuals are cooperating. Among them are customers, authors, producers and directors as well as designers, programmers and technicians. Often this cooperation is merely based on diffuse arrangements between all participants and between the customer and the producer in particular. There are no proper means of communication available to unambiguously describe the anticipated results from scratch or the stage of completion during the production process other than through traditional sketches, abstract verbal descriptions or technical excerpts. Furthermore, since different tasks are interdependent – e.g. application programmers waiting for modellers to deliver functional geometry, while modellers waiting for the author to complete the list of required geometry – any kind of intelligent interweaving of interdependent production processes during development immediately transfers into time-saving and an increased cost efficiency. Therefore, there is a need to facilitate the production workflow of complex VE presentations with storytelling content.

2. The Knowledge Game *Entropy*: A Sample VR Production Process

As an example to illustrate the development process of interactive, non-linear stories in VEs, we chose to create a knowledge game – Entropy – for a VR theatre. The audience will experience in an emotional and exciting way this per se abstract term, how information is collected, stored, transmitted and, in some circumstances, lost – that 'information is a volatile commodity'. The elaborate exposé envisions an alternation between interactive and non-interactive (movie like) acts. The audience will be guided by two characters, namely the come alive cursors of the prominent video game *Pong* from the early seventies. The Pong cursors explain how to use the interactive tools and the rules of the game, before wishing the players luck and making way for part one of the game. In the following the audience will have to achieve different tasks like for example collecting symbols, sounds and colours during a high-speed flight through an abstract world of information. Some of the tasks will only be solvable collaboratively. Dependent on the dexterity of the audience and the resulting score at certain levels the progress of the game will change.

To sketch a complex, multi layer non-linear story and behaviour like this in a traditional way requires an extensive description. It would be helpful if the author is provided with an appropriate authoring tool that can describe this scene's behaviour inside the overall story. Rather than developing some sort of automatic story or dialogue generating apparatus the purpose is to facilitate the scripting of the author's vision of a well composed largely anticipated scenario. Rather than to describe parallel plots in the story only, an authoring tool should allow for describing the behaviour of a scene and its objects under certain conditions. This would give the user full control over the timing in certain sub parts of the story and allows the maximum of possible not pre-defined interactions.

After that the author's work has to be transformed into a working VE story. Ideally, the logic of the story as defined and hopefully tested by the author can be imported into the VE system. If not, the intention of the author has to be guessed and programmed by a programmer. This will most likely be error-prone regarding both, the proper translation of the author's intention into story behaviour as well as the robustness of the implementation.

Parallel to this, the modeller has to get a detailed impression of how the geometric objects in the scene should look like. He has to rely on the descriptions in the authored story. In turn, the modellers will have to be able to show their modelling results to the author in an early stage to verify their work. For example, if an object misses certain attributes the author regards as important but forgot to specify, then this should become visible as early as possible.

If then, the modelled and programmed story will be displayed in its final location, the virtual environment, details in the scene and the timing will inevitably have to be adjusted. This is to adapt to specific display systems attributes but also to tune the story based on real experiences of the story in an immersive surrounding. Ideally, these adaptations can be done directly by the author while experiencing the story. Otherwise the programmer will have to reprogram the story for each change which is a time consuming process and interrupts the experience.

In the following chapter, we will describe our approach to supporting these processes.

3. The alVRed Tools

To support the creation of interactive, non-linear content in VEs, we developed the alVRed tool set. It consists of the VR Authoring Tool, the VR Previewer, the StoryEngine, the

ObjectFactory and the VR Tuner. The latter three are an extension of Avango™ as they are integrated into its framework. Figure 1 shows the relation of all five tools in the development process of a VE production. With the Avango™ StoryEngine connecting to the Authoring Tool, the Previewer linking the geometry to the RenderEngine of Avango™, and the Tuner accessing the parameters within the ObjectFactory, individual updates to the content immediately become available to all sub-systems during runtime. Updates can be changes in the story logic by the author, modifications of the geometry by the modellers or presentation, and timing optimisation by the director. We describe these separate tools in the following sections and, at the same time, describe their interfaces.

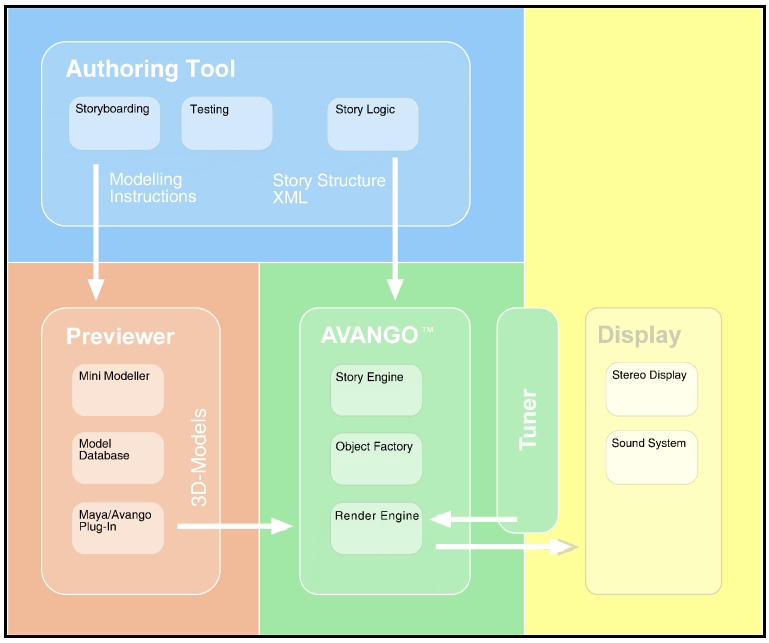

Figure 1: The alVRed Tools for Storytelling in the Production Process:
Authoring Tool, Previewer, Avango™ StoryEngine, ObjectFactory, and Tuner

3.1 VR Authoring Tool: Creating the Story

The core requirement for the VR Authoring Tool is to provide story authors – in general not familiar with programming – with an instrument to transfer their vision for an interactive story into a computer. While there are numerous tools available for the creation of 3D images and sound, there is no tool for the VE story structure creation, in particular for non-linear stories with a hierarchical structure. Our developed easy-to-use authoring tool closes this communication gap between authors, 3D world modellers and programmers.

Story Logic

The basic idea behind the authoring tool is that multiple and hierarchical graph structures are easy to grasp and particularly suitable for the creation of non-linear stories and worlds. All elements that are relevant for an envisioned scenario are built up on a panel as directed parallel graphs by mouse-click. The nodes of a particular graph represent the possible states of the generated objects which in turn are representations of visible (e.g. scenery, doors, characters,...) and invisible (e.g. camera, timelines, apertures, plots,..) objects in the virtual world to be modelled. Interactions with the story are defined within objects and by scripting. Due to the very universality of our notion of an object there is no conceptual need to distinguish a graph modelled object which for example represents a character in the VE from

an object which represents an abstract (and invisible) entity like a timeline. Figure 2 shows one node inside the overall graph of the story plot (the *storygraph*) specifying the different possible sub states of act 2 ('Info-Hub') of our Entropy scenario.

In the running scenario, the current state of act 2 will in turn depend on the states of numerous other objects like for example the memo game's timeline, earlier scores and user actions.

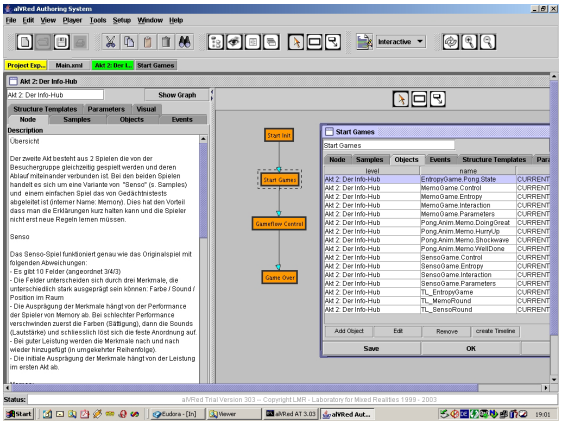

Figure 2: Storygraph Fragment, Scripting

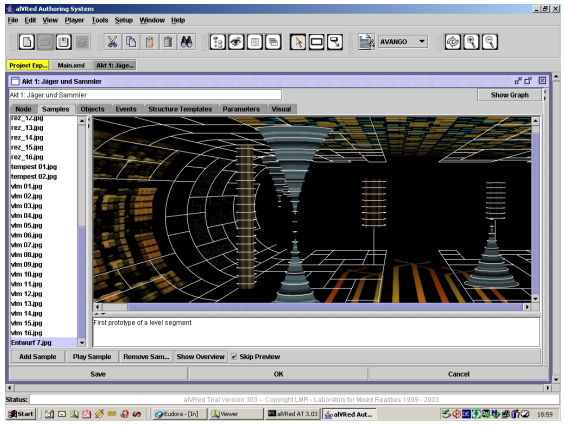

Figure 3: Storyboarding

The conditions for a node transition – i.e. change of an object state or story progress – are scripted within the nodes. To support the author, who might neither be aware of technical constraints of a certain VE target platform as well as of its full potential, a specific building block system can be established from which he or she can choose from.

Storyboarding

Within each node, event specific information describing the scene or the object appearance can be deposited by the author. This information – pure descriptive text and attached sample sounds, pictures or movies – serves as illustrative material for the modellers of the virtual world. The internal media player of the Authoring Tool enables the author to access the attached files for most of the current media file formats. In Figure 3 the author chose to add an image of the envisioned VR world to illustrate its desired appearance in the story for the modeller. Like the states of the objects the author designs the 'states of the story' with the help of a storygraph, which is on the bottom line just another but *central* object. Since there are basically no restrictions on how complex a story can be, the tool also offers the possibility to encapsulate entire parts of the story (e.g. self-contained story parts) within a – in that case – hierarchical higher node. Thus modelling of complex scenarios can be arranged more clearly by the author while the scenario structure itself gains stability and manageability for the associated story engine.

Testing

An internal story player allows the author a *walk through* of the designed story long before any elements of the virtual world are modelled. This testing functionality is of great importance to check the scenario for consistency, timing and suspense. For more complex stories their (long time) behaviour may not be predictable anymore and an 'on scene' testing with a costly target platform will often be unaffordable or come too late. Since the Authoring Tool is not based or dependent on concrete 3D models (which may not even exist by the time of testing) the progress of the story is presented to the author in purely textual form, similar to a program in debug mode.

Implementation

The Authoring Tool is programmed as a stand alone application in Java 1.4 and runs on Windows, Linux and Macintosh platforms. The developed story structure and its complete inner logic is stored by the Authoring Tool as an XML file. It will later be used by the story engine of the target platform.

3.2 VR Previewer: Visualising whilst Production

The VR Previewer module provides the Avango™ runtime system with geometry for the visual rendering. This module supports an iterative way of visual content generation, offers real-time monitoring of first sketches and further results, and enables parallel development of 3D models and application programming.

The author describes objects and animations with the Authoring Tool. In a first step, a required set of requisites has to be determined, organised and translated into an abstract structure of geometry, audio, light or interaction for immediate use within the alVRed tool set.

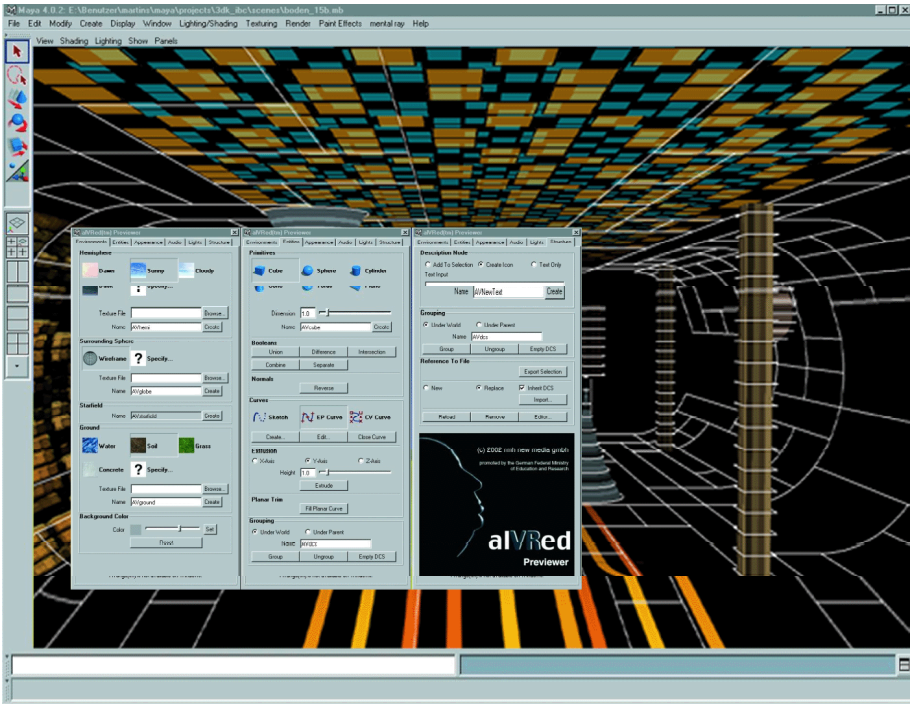

Figure 4: The VR Previewer with GUI Elements

With the 'Mini-Modeler' segment of the Previewer, a structured scene graph can easily be established, providing the foundation for all further refinements. Defining elements which are already available (i.e. sketches, written descriptions, template geometry, audio files) can be included in the 3D world and subsequently substituted with proper modelled geometry during an iterative production process. All nodes receive a unique ID, thus, being accessible by the Avango™ RenderEngine. Individual geometry files can be externally referenced supporting facilitated decentralised modelling. Furthermore, the Previewer provides editing functionality and support of advanced rendering features like multi-texturing or real-time shadowing.

The Previewer is implemented as a plug-in inside Alias|Wavefront's 3D-Modelling Framework Maya utilizing MayaAPI and MEL (Maya Embedded Language). A graphical user interface (GUI) provides easy handling of basic modelling tasks ('Mini-Modeler') as well

as advanced functionality for scene graph editing and organisation without previous knowledge of Maya itself. All powerful import/export, modelling and animation features of Maya remain available for the expert user.

With the click of a button the Previewer can launch the Avango™ runtime environment, putting the most current visual state of the project on display. The Maya scene graph is interpreted from within memory and transferred to Avango™. Alternatively, it can be exported as a file, compatible to OpenGL Performer with Avango™ specific extensions.

3.3 Avango™ StoryEngine: Executing 'Story Logic'

The Avango™ StoryEngine implemented in Elk/Scheme, represents the runtime environment of the story logic which was designed by the author with the help of the Authoring Tool. An XML-based file format allows for simple data exchange between these. The core technology of the StoryEngine are communicating hierarchical finite state machines (HFSM) which are based on the well-known UML specification. At runtime, the logic of every story object (namely nodes, event-scripts, attributes), which was created in the VR Authoring Tool, is translated into a corresponding HFSM. All HFSMs are executed in parallel and communicate with each other through events. Typical events are creation, deletion or state change. As the StoryEngine is a layer of its own, the story logic can be tested, changed or reloaded at runtime.

3.4 ObjectFactory: VE Representation of 'Story Logic'

As the StoryEngine solely represents the story logic of all story objects, the task of the ObjectFactory is to link all of them to their actual VE counterparts (e.g. graphics, sounds, interaction), the VE objects. The *state* of an object is managed by the StoryEngine. The reaction to state changes, however, affects the rendering (visual, acoustic, haptic etc.) of the VE objects. On the other hand, the VE objects can also send messages to the story objects, which then can react accordingly.

The ObjectFactory consists of a class library, implemented in Elk/Scheme. A common interface serves as base for the VR Tuner. The ObjectFactory provides a wide range of standard and special effects building blocks. Some of them operate directly on the Avango™ scene graph; others implement a specific functionality and can be combined to provide a high-level abstraction of a certain task. The current feature set includes among other things

- geometric transformation, deformation and animation
- texture and image manipulation, movies
- physical simulation and particle systems
- spatial audio and olfactory rendering
- tactile feedback and haptics
- interpolation (numbers, position, time, curves etc.)
- visibility, proximity, time and other sensors
- input devices and interaction abstractions

In addition, there are many compound, high level objects, for example door and vehicle abstractions, cameras, and timelines. Most of them are already available to the author within the Authoring Tool.

3.5 VR Tuner: Fine-Tuning the VE Representation 'In Situ'

With the VR Tuner details of a production can be modified at runtime while experiencing the story inside the VE. The Tuner consists of a 2D GUI for viewing and direct manipulation of object parameters. The GUI, which is presented on a Pen PC is supported by a set of hardware motor faders to change values in a way as it is known from light and video mixing consoles. Both components are mounted in a flight case (Figure 5). As an option, the Pen PC can be worn independently from the rest of the hardware, and used together with a 6DOF interaction device, the stylus. The stylus is used for the selection of objects inside the 3D scene.

Figure 5: VR Tuner Hardware inside the Virtual Environment at Runtime

The 2D GUI is displayed on the wearable computer and makes the Tuner available anywhere inside the VE. In practice, the user holds the Pen PC in one hand, and in the other hand the stylus for 3D interaction. With this setup, the story objects can be interactively tuned while the user is immersed in the VE. This guarantees full WYSIWYG (What You See Is What You Get). The selection of objects can be done directly with the stylus in 3D or by clicking on the 2D tree view representation of the scene graph on the Pen PC. The scene graph oriented 2D interaction is especially suitable for complex scene graphs and for the selection of objects without a selectable 3D representation. This can be, for example, invisible objects, light, or sound. No matter which selection method was used, the Pen PC will show the corresponding set of parameters for the selected object or environment. Certain values of objects are assigned to specific hardware faders on the mixing console. If an object is selected, the faders are moving to the positions, according to the assigned values.

This allows the user to inspect and tune the modifiable parameters of the objects. Custom dialogs and mixer setups are available for light, colour, sound, and other objects which are closely related to the final representation of the story in the actual display system. A generic dialogue handles other object types.

To avoid irritation of the user when switching attention between the 3D environment and the pen PC surface, the 2D and 3D interaction is always synchronised. This is the case for the Tuner, as, after a selection in 3D, the user will find the Pen PC interface and the motor faders in the expected states. At the end of a tuning session, all tuned values can be saved and stored

for later presentation of the story. The tuning and storage of modified parameters is separated from the main design and production process. Thus, it is possible to perform last minute adjustments of the story and its visual representation to adapt to different display systems or specific presentation needs.

4. Results and Discussion

To evaluate the alVRed tool set, we chose to produce the interactive, non-linear knowledge game 'Entropy' as described in the second section. The story was authored by a professional author. All previously described tools were and are still used in the development process. Visually, the story among other things includes the scenes whose geometric representation are shown in Figures 6 and 7.

The VR Authoring Tool developed within the framework of the alVRed project proved to be an efficient and easy to use tool to the authors developing the story. The on-screen interface was felt as intuitive and easily understood. The available spatial distribution of texts in the story graph even developed its own dramaturgical significance. Herein lies the basic difference to the linear sequence of a traditional manuscript. The creative task can be compared to using a flip chart with an almost infinite depth. However, the author is not required to physically flip the pages – any depth can be reached by a simple click of the mouse. In contrast to the traditional method using paper and felt tip pen, this digital authoring system made it possible to include all other media like images, videos and sound files within the texts. Since the links between the scenes were installed by the author himself, he was also able to perform simple visual checks of the correct dramaturgical flow. And the author arranged the individual scenes in a clearly structured form which was not possible before in written manuscripts.

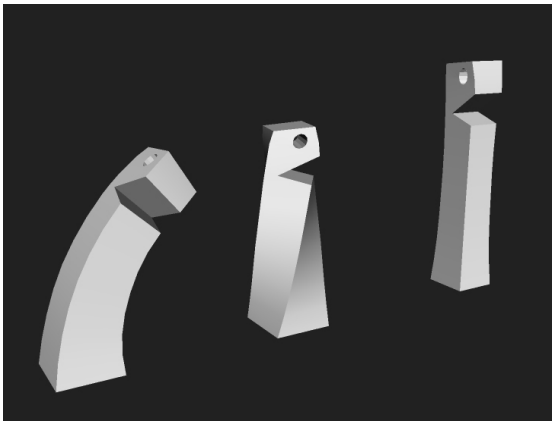

Figure 6: Entropy: The Pongs

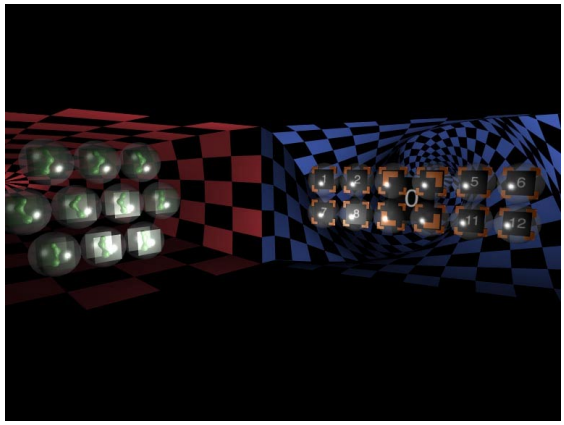

Figure 7: Entropy: The info hub

The flow and logic of the story was checked by the author himself using the internal story player of the VR Authoring Tool. This proved to be useful in testing the logic of the story which is imported as such into the runtime environment. Subsequently required tasks, like programming, were greatly simplified by the data files generated with the authoring system. The story logic was already tested and therefore known to work in the final application. Thus it was possible to introduce changes in the story logic at a late production stage, as the logic is stored independent from the programming. This makes last minute changes of the story progress possible and allowed for parallel story authoring and programming. These changes used to be extremely time absorbing in the past and could only be done by a computer

specialist. This approach also allows the author to store different storygraphs for different purposes (versioning).

Due to its completely abstract concept and very general output in form of an XML file the VR Authoring Tool is not confined to a certain target platform. It is currently tested by several other authors and research institutions in Europe, North-America and Asia as an input device for other storytelling systems and game engines. On account of the novelty of this kind of story modelling it took time for some of the authors to familiarise with the authoring concept. Generally it was found that authors who worked with the tool only a short amount of time tended to create completely classical, linear stories, which of course is valid too.

The proposed systematic workflow has resulted in a much more efficient structuring of the overall production process. Collaboration between the author, the modellers and the programmers has been facilitated as interdependent production tasks were performed in parallel, while working on the same scene graph and on the same corresponding story objects. Thus, all participants were able to contribute their content in a continuous way rather than on isolated sessions dedicated to system integration. Communication during development was supported by taking preliminary 'snapshots' at different production stages, which served as examples for further discussion. As the underlying data could easily be retrieved from within the alVRed framework any time during the story build-up, the production team was able to take frequent 'snapshots' that served as a visual aid for decision-making and motivation. Concrete time saving figures can only be guessed as no comparative data is available. However, due to the fact that there was virtually no idle time during production compared to unorganised approaches, a factor of two seems to be a realistic estimation. In this example, we needed approximately only half the normal time for the production because of parallelisation and iteration.

Compared to prior experiences with the VE production process, with the support of the alVRed tools we could achieve a very fast implementation of the story. The interfaces could be approved and the introduction of creative processes in technical environment worked well. In earlier smaller test scenarios we already successfully integrated changes of story and dramaturgy at runtime. In addition we could now modify and add 3D objects with specific behaviour during story execution.

To make this demonstrator production a real field test we did not only want to create a compelling scenario which successfully runs on our own display system and consequently has a very limited number of users. Rather we looked for a possibility to make it accessible to a greater public, where the scenario has to prove itself in terms of stability and suspense. Hence we are glad that the Technical Museum in Vienna (Technisches Museum Wien) agreed to have this production installed in its VR theatre where it will be shown from April 2004 on.

5. Conclusions and Future Work

Interactive narratives and VEs of the past were based on crude abstractions of reality. While in the field of 3D-rendering and visual complexity dramatic improvements have been achieved in recent years, we are still waiting for more complex and compelling story structures. Usually, the logical structure of narratives and games is based on simple branching events. The designer of the system would set up a number of alternative decisions and interactions and allows the user to directly or indirectly choose one. The tools and methods we described in this paper significantly improve not only the efficiency but also the complexity of VE productions.

The VR Authoring Tool enables creative authors to digitally develop, organise and define complex non-linear stories. It supports the sketching of first vague ideas up to the definition of the whole logical structure of a story in detail. Due to the very universality of our notion of an object there is no conceptual need to distinguish between characters and the behaviour of other 3D objects. The logical interdependencies between the behaviour of objects are described by event scripts and the story logic can be tested using the internal player. The VR Previewer is a fast preview module included into the Maya modelling package. It provides for an iterative way of content production and offers real-time monitoring of first sketches and further results. As the StoryEngine solely represents the story logic of all objects, the task of the ObjectFactory is to link all of them to their actual VE counterparts. The VR Tuner provides a combined 2D/3D interface to tune parameters of objects from the ObjectFactory during runtime and enables the author to realise last minute changes in the final installation. Although this project is still ongoing, we could already demonstrate the strength of most of the developed tools for efficiently creating complex non-linear VE content by means of a small test production (interactive version of 'The Little Prince', 2002). The much more complex scenario 'Entropy' will be installed publicly as from April 2004.

The great challenge for digital storytelling is to overcome the problem of completely predictable and anticipated systems and to allow for the design of more open narratives that nevertheless follow dramaturgical principles. Hence, telling highly individualised stories means to deal with the old conflict of control and autonomy. The basic research question for interactive storytelling is, how can we create dramaturgical robust and narratively consistent systems that on the other hand produce interesting and surprising behaviour? Our concept of multiple hierarchical graph structures already allows for the description of complex scenarios with simultaneously active objects. At the moment the author keeps control by having a central storygraph at the top of the hierarchy that is able to determine the overall development of the story. In the future we will investigate other possibilities to assure dramaturgical interesting story development. Giving up the concept of the central storygraph will lead to open-ended narratives but rises the question of how to distribute dramaturgical control onto simultaneously active objects.

Acknowledgements

The ongoing 'alVRed' project is funded in part by the German Federal Ministry of Education and Research (BMBF) under grant No. 524-40001-01IRA06 within the 'Virtual and Augmented Reality' project framework.

The research and the development results presented in this paper are the collaborative work of the following persons:

S. Beckhaus[2], S. Conrad[2], M. Goebel[2], B. Gruetzmacher[3], F. Hasenbrink[5], T. Holtkaemper[2], R. Jain[1], A. Lechner[5], S. Mostafawy[4], R. Mueller[1], M. Suttrop[4], J. Thoma[1,] G. Trogemann[3], R. Wages[3]

[1] facts+fiction GmbH, Cologne, [2] Fraunhofer IMK, Sankt Augustin, [3] Institute Laboratory for Mixed Realities, Cologne, [4] rmh - new media gmbh, Cologne, [5] Vertigo Systems GmbH, Cologne, Germany