VSMM 2003
15-17 October 2003, Montreal, Canada

An Authoring System for Non-Linear VR Scenarios

Benno Grützmacher, Richard Wages, Georg Trogemann
Laboratory for Mixed Realities
Am Coloneum 1, D-50829 Cologne, Germany
{gruetzmacher, wages, trogemann}@lmr.khm.de

Abstract. Nobody would ever try to realize a movie without using a script. What is quite natural for movie productions seems to be the exception for VR productions. In VR we still observe a lack of approaches and tools that allow the description of an analogue to the movie script. The authoring approach proposed in this paper is based on *directed parallel graphs*, where the nodes represent story-relevant situations and the edges represent user interactions or self-triggered transitions (i.e. internal temporal, spatial and logical events). The basic functionalities of the authoring system are storyboarding, event scripting and logical testing. The storyboard function allows the user to define the overall structure of the story and to attach additional information (images, sound, movies, scribbles etc.), which is relevant for the production process. The logical interdependencies between the development of the story and the participating virtual objects are described by event scripts. The integrated internal story player finally allows for a logical traversing of the story.

1. Introduction

Beside the necessity to produce pictures, movies, sound etc. the creation of interactive VR scenarios requires big efforts in composing this 'raw data' into a real product (scripting). The latter usually is done from scratch and not by the authors themselves but rather by programmers. This inevitably leads to an altering of the author's ideas of the product outcome and to limited reusability capabilities resulting in a high consumption of time and manpower. Plenty of authoring tools for the creation of VR imagery and sound are available and used by artists who are no computer experts at all. The same is not true for a tool to compose the analogue to a (linear) movie script – a non-linear scenario script.

In this paper we describe our VR scenario authoring tool, which tries to close this particular gap in the VR production process. We propose an authoring approach based on *directed parallel graphs*. Graph structures in our view have two advantages which make them the ideal bridging elements between VR story engines and authors: Due to their unambiguous structure they absolutely qualify as an input for story engines from a technical viewpoint. Yet they are very intuitive and easy to grasp for non-computer-experts too when it comes to the modeling of processes. Hence we will utilize these very structures for authors of VR scenarios.

After a brief depiction of the underlying concept – that is a splitting of the scenario into several different objects with an own inner logic – we describe the different functionalities of the tool in detail. We point out the important future role of in advance scenario testing and finally give an example for a small implemented interactive story.

2. Related Work

The creation of interactive productions or non-linear VR scenarios is supported by several products on the market, usually strongly dependent on the desired output format. Typically these tools offer some assistance to the author to enable her or him to create content easily while restricting the author's possibilities, be it in output formats or the author's creative freedom.

Some powerful authoring tools are focusing on the creation of interactive multimedia content like the well-known tools Macromedia Director [5] and DVD Studio Pro [2]. But to be able to utilize Director's full opportunities for example an author has to be familiar with the embedded scripting language Lingo, which highly enhances the possibilities of the tool but demands advanced programming skills. To avoid the latter we equipped our tool with a very simple if-then-scripting – the only logical expression an author has to learn.

Many products have a stronger focus on the 3D design of these VR worlds, as for example the level editors for computer games [4,7,14]. These are used to build up 3D worlds directly which can easily be equipped with different objects and some user interaction possibilities. Other tools address special issues to enrich objects in VR scenarios with more complex behavior as for example the AI simulation tool Simbionic [12]. Beside the fact that most of the mentioned tools are game engine specific they all offer no support for dramaturgically oriented long term scenario behavior or a plot. In contrast the tool presented in this paper emphasizes on the modeling of story and scenario progress itself while deliberately sacrificing modeling opportunities for visual or audio objects.

Only a few available products support the development of mere story. For the development of linear stories the most prominent tool is Dramatica [11] which was used to write the scripts of numerous movies. This software assists the author in building up a plot and characters based on a sophisticated underlying theory of story construction and delivers an outline to be filled with text. A tool to create *non-linear* interactive stories is Storyspace [3] which enables authors to create hypertext novels with a graph defining the branching points. Surely the output of both tools is not suitable for the direct creation of VR scenarios. On the one hand the XML output of our tool always requires a suitable interpreter for a target platform. On the other hand XML is a very general standard which does not entail assumptions and restrictions about any targeted platform, hence allows for a maximum degree of freedom in the design of abstract scenario logic.

3. Graph Based Authoring of Non-Linear Scenarios

When using graph based methods (finite state model) the story structure or progress is modeled as a net of state transitions. The *nodes* of the graph represent particular situations (scene, camera shot, attributes of the environment), the *edges* represent user interaction or self-triggered transitions. When we talk of states we usually only consider *story relevant* states. In huge VR environment a single state might enclose options for impressive spatial navigation. Nevertheless this could be a completely irrelevant fact from a storytelling point of view.

A story modeled as a graph always only offers pseudo decisions to the user within the given structure. Different scenario progresses are merely different trajectories within the authored 'story space'. The total of the user's 'decisions' establishes a path through the graph in the form of a sequence of node transitions. Additionally the system itself can trigger such transitions. Typical conditions for the latter are temporal (e.g. time out), spatial (e.g. user navigates to certain place) or logical (e.g. indirect consequence of a selection done by the user). In the easiest case a single node just stands for a self-contained linear film sequence. In more complex scenarios each node in return might contain an entire graph, which represents a certain aspect of the story (sub-story).

With the below described authoring tool we want to enable creative authors to develop, organize and define complex non-linear scenarios. This process should start from sketching out first vague ideas and lead to the definition of the entire logical structure of the story in detail. Although a high level of complexity of those structures will be inherent in non-linearity itself, this tool should be easy to use for authors who are not familiar with abstract structures too. On the other hand we take into account that there are authors, who feel comfortable even with high levels of abstraction. They might appreciate having access to the deeper structures of the story and do not want to be limited by predefined structures. This tool should address both types of authors.

3.1 Storygraph and Object Graphs

The nodes of a graph represent all possible states a system can attain, whereas the edges describe the possible transitions between these states. Only one node can be active at a given time since an active node already defines the actual state of the graph. Figure 1 shows an example of several parallel *directed* graphs whose active nodes are marked. Possible state transitions are indicated by the arrowheads of the edges.

The theoretical background behind our graph based approach is the idea to approach the quandary every author of interactive scenarios is faced with: She or he might want to design the scenario in a way which allows for a lot of influence for a user. As a result it may be characterized by a great openness, but at the same time lack dramaturgic suspense or be unable to convey a story to the user at all. If the author on the other hand sticks to more classic forms of dramaturgy with a firm control over the progress the scenario might end up being rather linear and not taking advantage of the interactive elements of the environment.

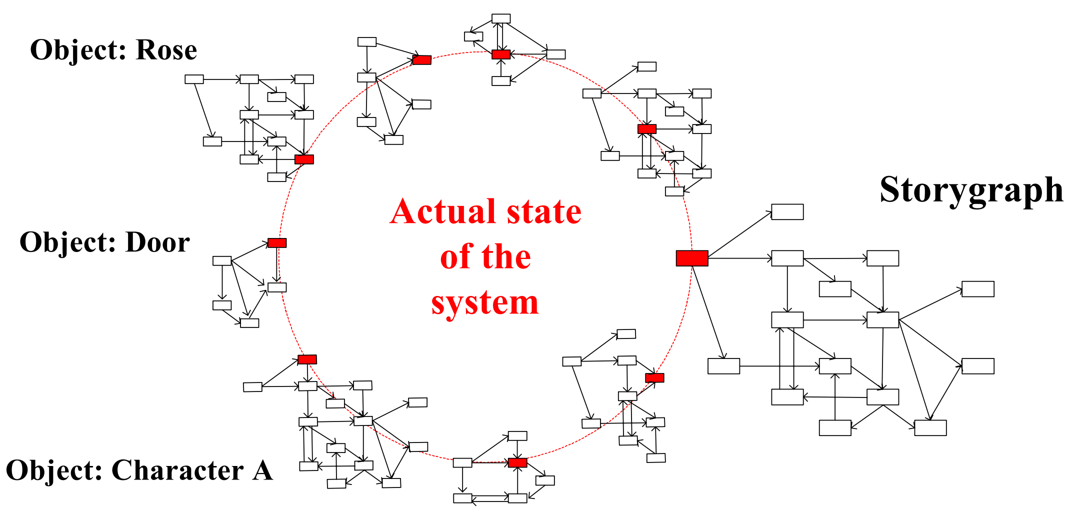

Figure 1: Scenario Graphs

We propose the following combined approach: Every story relevant entity (character, door, scenery, timeline etc.) is modeled as an object graph with its own inner logic. For storytelling the possible states of these objects usually do not have to be exhaustively accurate in the light of physics or psychology. Rather the author assigns a few story relevant states to them together with their respective dependencies, hence somewhat cedes control to the runtime system.

At the same time an author has the opportunity to form a special object – the *storygraph* – which more or less strongly enforces her or his intentions on the scenario progress during runtime. Depending on how strict this progress is formulated many different forms of scenario types are imaginable regarding time management and openness.

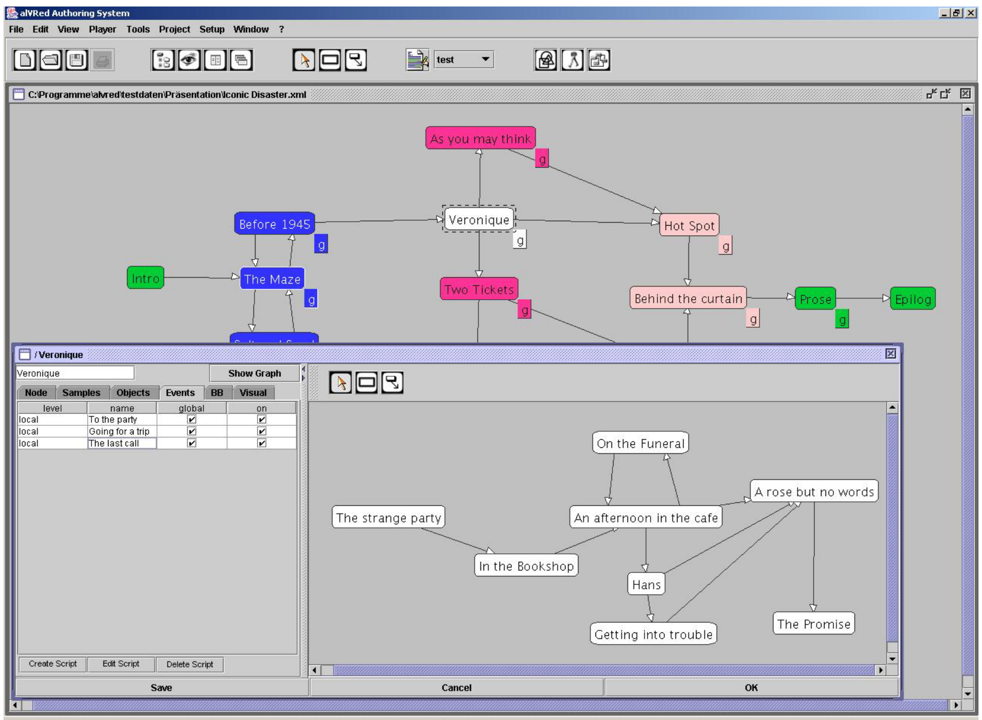

Figure 2: Storygraph hierarchy levels

Two further functionalities for the creation of objects remain to mention: Graphs are discrete structures while for many objects the introduction of continuous parameters is reasonable (e.g. brightness of a scenery). Hence the authoring tool offers the possibility to assign these *attributes* to objects. Furthermore the tool meets the leaning of authors to start out with a coarse draft and elaborate them successively by offering an optional number of hierarchy levels (Figure 2).

3.2 Special Objects and Building Block Kit

Beside the possibility to model objects by building an object graph as described above, the authoring tool also offers some special objects to the author which are part of a *Building Block Kit*. The collection of these special objects (containing scenery, aperture, switches, timelines, camera etc.) is not considered as complete but to be extended from time to time. The relevance of these objects is twofold. On the one hand some of them are very common and therefore often used but maybe extensive to build. So it is manifest to provide them in a kind of library. On the other hand specific objects can be used to represent some peculiarities of a particular target platform. Furthermore they can directly represent some objects of a platform as counterparts.

3.3 Scripting

Event Scripts are used to define the logical interdependencies between the storygraph and the objects. Since *any* scripting language will offer the known assets and drawbacks [9] we focused on the straightforwardness of our syntax. An event script consists of a list of conditions, which must be validated to activate this script and a list of results which will then be executed. These scripts can be assembled inside any node.

Event Script: Conditions (c_1, c_2, c_3,..) → Results (r_1, r_2,..)

Within a node we can refer to (call up) any other object and thus link these objects. In an event script of the above form a state change of a called up object might now serve as a triggering condition. As well an event script can lead to a state change in a called up object or even in a state change of the particular object itself (e.g. storygraph progress). It is in the form of event scripts too how an author can model interaction possibilities.

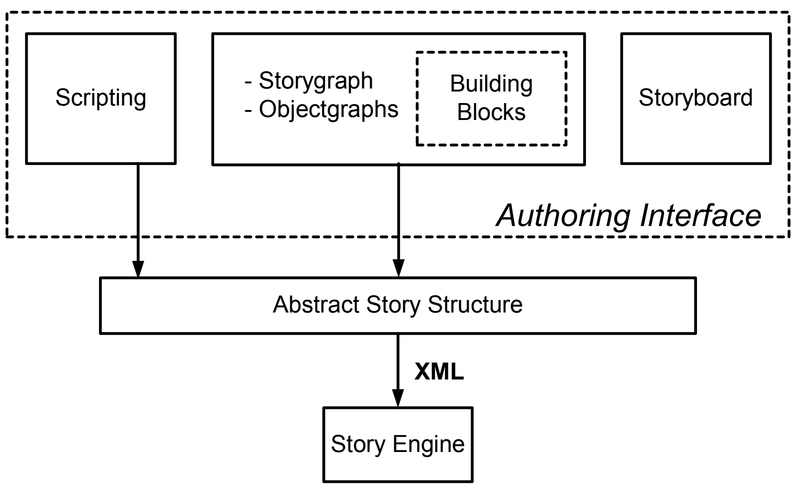

Figure 3: Functionalities of the VR Authoring Tool

The entire logic and hence abstract description of a scenario is stored by the authoring tool as an XML file. To utilize this 'non-linear script' a story or game engine will need an appropriate importer. Within our ongoing project this XML story file in a test was successfully imported into the Avango™ StoryEngine[13].

4. Internal Story Player

The integrated internal story player allows for a logical traversing of the story. Thus the author not only can detect possible logical errors (debugging) but also is able to try out and tune

(diverse) scenario progresses to a certain extent. The running story player dynamically displays all elements referred to in the active storygraph node. In particular the author can observe active timelines, *story time* (time spent in the scenario), *node time* (time spent in the current node) as well as a list of all event scripts and objects referred to in this node. In addition all events are recorded in an *event log*.
With a growing sophistication of interactive scenarios such a testing opportunity becomes increasingly important since their (long time) behavior is getting less predictable for authors. The interplay of even a set of simple parallel entities is capable of generating complex patterns [15]. Roughly the procedure rules for the parallel graphs are as follows (Figure 4):

- When activating a node actualize the states of the called up objects according to the authored specification.
- Find matching event script conditions and add them to the list of matched scripts.
- Continue executing the scripts in the list of matched scripts.

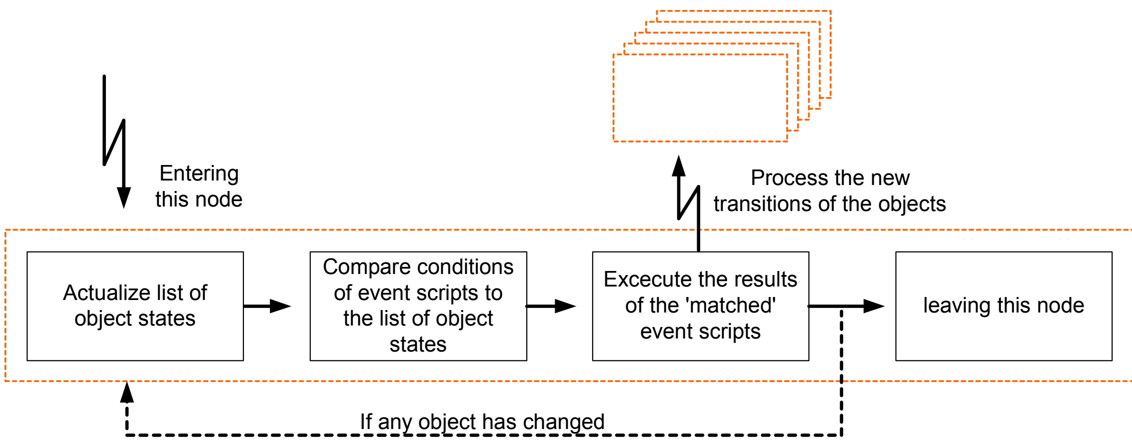

Figure 4: Script Processing

It is an important aspect too that authors now are able to 'run their stories' long before a single image or sound is modeled.

5. Storyboarding

Our authoring tool focuses on the modeling of the logical structure of a story and does not provide any functionality to model images of the virtual world, characters etc. The author should have the possibility to lay down and illustrate his or her conception of the way the virtual environment should look like and other information which is useful for the further production process. Hence the authoring tool provides the possibility to write down text description for every node, which describes the content in the author's words and can as well contain unambiguous instructions for modelers and programmers. In addition the author can attach (pictures, movies, sound) of several media formats to give a more concrete illustration of her or his ideas. The Authoring Tool provides an integrated media player to present the different media.

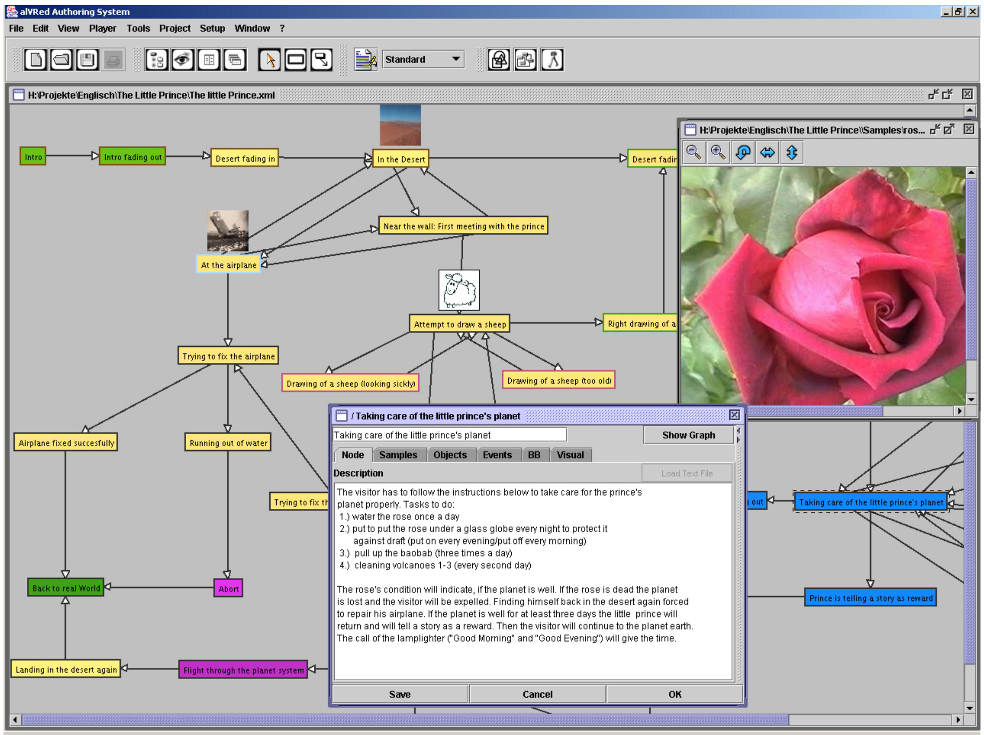

Figure 5: Storyboarding Functionality

6. Conclusions

The tool we described in this paper enables creative authors to define, organize and test non-linear VR scenarios. Using this tool not only leads to more efficient VR productions, but also allows for the realization of more complex story structures. The proposed approach based on directed parallel graphs is very powerful and easy to grasp for non-computer-specialists. In a first test implementation – an interactive adaptation of a fraction of the story 'The little Prince' by Antoine de Saint Exupéry [10] – we were able to make on-site changes to the story progress on the Avango™ Cave platform within a minute (Figure 5). In addition storing sole story logic as a relatively small XML file offers the option of versioning, i.e. using the same imagery and sound for entirely different presentations.

7. Future Work

One of the great challenges in the field of digital storytelling and VR is the design of more open scenarios. To overcome the restrictions of navigating through predetermined stories, in the future we will investigate two different principles: Artificial Intelligence (AI) techniques and *heterarchical* structures [6]. One feature of future authoring tools will be the description of AI-based character behavior and story development. At the moment AI-elements in VR productions have to be programmed by AI-specialists. The other principle that is very promising in the context of interactive VR Scenarios is that of heterarchy as opponent to hierarchy. In computer games heterarchy in its simplest form is already used to achieve game balance. The use of non-transitive rules is recognized as a effective mechanism to avoid dominant game strategies [8].

The described concept of parallel graphs – the splitting of the scenario into several different objects with its own inner logic – already allows for the description of complex heterarchical structures. At the moment we guarantee that the author keeps control by having one special graph – the storygraph – that determines the overall development of the story. Giving up the existence of the dominating storygraph leads to open-ended systems whose behavior can hardly be predicted. To use such heterarchical structures as basic design principle opens a Pandora box of questions for digital storytelling. What are the basic design principles for open systems? Are there general rules that can be applied? How can we create reliable, robust and consistent stories that are on the other hand surprising and unpredictable? How can the author keep some control over the development of the story? How can these open systems be tested?

8. Acknowledgements

This work is part of the ongoing 'alVRed' project [1] and was supported by the German Ministry of Education and Research (BMBF) under grant No. 524-40001-01IRA06A within the 'Virtual and Augmented Reality' project framework.

References

[1] Project alVRed, http://www.alvred.de
[2] Apple, http://www.apple.com/dvdstudiopro/
[3] Eastgate Systems Inc, http://www.eastgate.com/Storyspace.html
[4] Planet Halflife, http://www.planethalflife.com/workshop/styleguide/editor.asp
[5] Macromedia, http://www.macromedia.com/software/director/productinfo/productoverview/
[6] W. McCulloch, A Heterachy of Values Determined by the Topology of Nervous Nets. Bulletin of Mathematical Biophysics, Vol.7, University of Chicago Press, Chicago, 1945
[7] RPG Planet, http://www.rpgplanet.com/morrowind/mods/editorfaq.asp
[8] A. Rollings and D. Morris, Game Architecture and Design, The Coriolis Group, LLC, Scottsdale, 2000.
[9] Rouse III, R.: Game Design: Theory & Practice, Worldwide Publishing, Inc., Plano, 2001
[10] A. de Saint-Exupéry, The little prince, Heinemann, London, 1954
[11] Screenplay Systems, http://www.dramatica.com
[12] Stottler Henke, http://www.shai.com/products/simbionic/simbionic_product_overview.pdf
[13] H. Tramberend, A Distributed Virtual Reality Framework, in L. Rosenblum, P. Astheimer, D. Teichmann (eds.), Proceedings Virtual Reality 99, IEEE Computer Society Press, Conference, Houston, Texas, 1999
[14] Unreal Developer Network, http://www.planetunreal.com/architectonic/first_level.html
[15] S. Wolfram, A New Kind of Science, Wolfram Media Inc., Champaign, IL, 2002