U-Create: Creative Authoring Tools for Edutainment Applications

Sebastian Sauer[1], Kerstin Osswald[1], Xavier Wielemans[2], and Matthias Stifter[3]

[1] ion2s – buero fuer interaction, Ober Ramstädter Straße 96b,
64367 Darmstadt – Mühltal, Germany
{sebastian.sauer, kerstin.osswald}@ion2s.com
[2] Alterface, 8 chemin des étoiles, 1348 Louvain-la-Neuve, Belgium
xavier.wielemans@alterface.com
[3] Vienna University of Technology, Karlsplatz 13, 1040 Wien, Austria
stifter@matthias.as

Abstract. The U-CREATE project aims at efficient content creation for new technologies, in particular interactive setups, Mixed Reality experiences and location-based services. A graphical authoring tool is developed to allow one to create elaborated contents in a fast and easy way. The present paper introduces the tool and elaborates on its main components, which allow the user to author an interactive story and also to manage the connected interaction devices.

Keywords: Authoring, Content Creation, Digital Storytelling, Edutainment, Mixed Reality, Mobile Learning, Location based Services, Editors.

1 Introduction

Not many content creation platforms are simultaneously targeting Augmented Reality (AR) and Mixed Reality (MR) applications, as well as conventional multimedia authoring. The predominant approach for authoring applications is time based (Dart [1]) and fewer are based on hierarchical (concurrent) state machines, like the prototype implementation of the U-CREATE[1] project, which allows the author to realize the applications with much less effort than conventional approaches.

Comparison and analysis of different approaches have shown, that the need for describing and creating interactive multimedia and AR/MR applications could not be covered be either the UML 2.0 standard [2,3] or others [4]. Especially the need for a general description language, which works at the base for the different target runtime platforms with their different scripting dialects, is a central issue of the described authoring and content creation system.

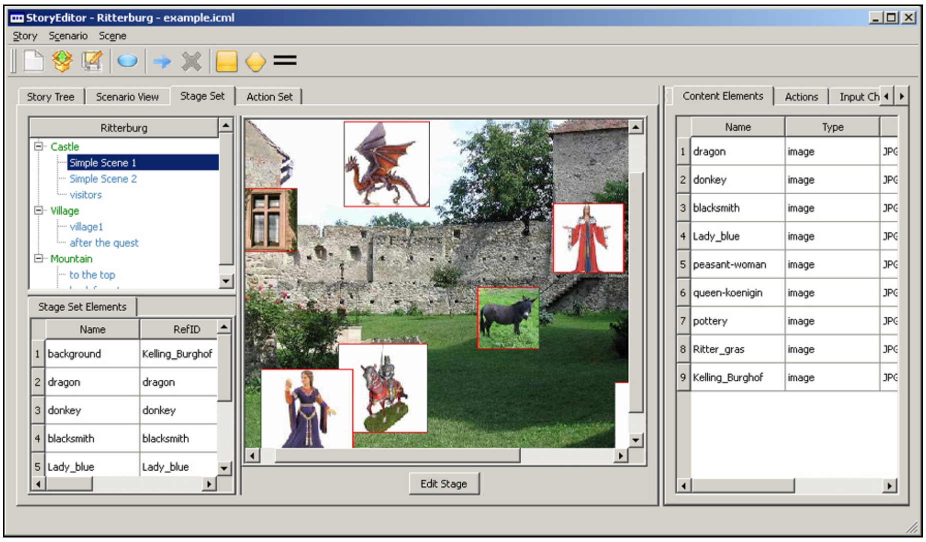
In the scope of the U-CREATE project, the goal is to provide an authoring tool for existing interactive technologies, which nowadays still require programming at

[1] U-CREATE is a CRAFT project funded by the European Commission under reference COOP-CT-2005-017683. For more details, one can consult http://www.u-create.org

S. Göbel, R. Malkewitz, and I. Iurgel (Eds.): TIDSE 2006, LNCS 4326, pp. 163 – 168, 2006.

code level. The authoring therefore pursues two goals: first, to increase competitiveness by significantly shortening production time (50% reduction of integration time) and effort (creation process affordable to non-specialists) for content development; and second to enable non-programmers to create contents for the intended systems.

Far from the latest considerations in interactive storytelling such as emergent simulated dialogues, the project aims at developing an authoring system that will allow anyone to create interactive stories that are experienced through mobile devices, VR trackers or gesture-based camera interfaces. In contradiction with [5], it intends to do so for anyone, not only people with "ability to program".



Ritterburg

- Castle
  - Simple Scene 1
  - Simple Scene 2
  - visitors
- Village
  - village1
  - after the quest
- Mountain
  - to the top

Stage Set Elements

	Name	RefID
1	background	Kelling_Burghof
2	dragon	dragon
3	donkey	donkey
4	blacksmith	blacksmith
5	Lady_blue	Lady_blue

	Name	Type	
1	dragon	image	JPG
2	donkey	image	JPG
3	blacksmith	image	JPG
4	Lady_blue	image	JPG
5	peasant-woman	image	JPG
6	queen-koenigin	image	JPG
7	pottery	image	JPG
8	Ritter_gras	image	JPG
9	Kelling_Burghof	image	JPG

Fig. 1. Story Editor (Preview)

For the present purpose, the need is to create elements (e.g. content, actors, artifacts, stages or objects), arrange them into relations and embed them together in the scenes of a user-controlled interactive story. The authoring process has therefore been organized around four editors, which are linked together in a unified interface:

- The interactive story itself is managed as a hierarchical stage graph edited in the S*tory Editor*.
- The various scenes, which constitute the story, are edited within the S*tage* E*ditor*, which provides a layout representation of these.
- The transition conditions between the story scenes and the actions that take place in the scene are defined in the A*ction Editor*.
- Finally, the *Input Wiring Editor* allows the user to define the exact use of the external devices to experience the story and how these are linked to the defined actions.

The following sections elaborate on the two major interfaces, the Story and Stage Editors while presenting the concepts on which the developed prototype relies.

2 Story Editor

The Story Editor is the authoring interface for the structural model of the story: it organizes its structural elements and presents them in different views.

2.1 Structural Elements

Story. It is the root node of each story space, including the whole installation that has to be realized. The entire story structure, all media elements, input channels (i.e. devices), and actions are parts of it. A Story includes one story tree, composed of all scenes, but can have more than one scenario.

Scene. It is logical part of the story, a state of the story graph, an entity that encapsulates certain media elements and actions and defines a chapter of the story. Complex scenes may contain other scenes, while simple scenes do not.

Scenario. A scenario defines a specific way to perform a story. It is an executable version of the story graph, typically defining a constrained path through the scene space. Examples could be “short version”, “guided tour for 1st floor”, or “directors cut”. Scenarios are a useful concept to design different variants of a story, and support concepts like shortcuts in the story flow.

Transition. This is a state change of the story, a switch from one scene to another. The transition is always related to some kind of condition, including an “empty” default condition that applies when the scene is at its end (all actions finished).

Action. The “atomic” performing primitive within a story is an action. It describes what a content element “does”, or where and how it is presented in the performance. If actions are combined with some media element, it can mean for instance to start or stop the element (e.g. play video clip XYZ). Additional non-media action types can be switches, bifurcations, synchronizers, and transition triggers.

Action Set. A group of actions and their connections (sequencing and/or conditional jumps) is called action set. It is enabled within the same state (scene), and working on the same set of media elements (stageSet) within a scene. It describes a set of performance primitives and their temporal and conditional relationships. Action sets include a description of possible (valid) user input events.

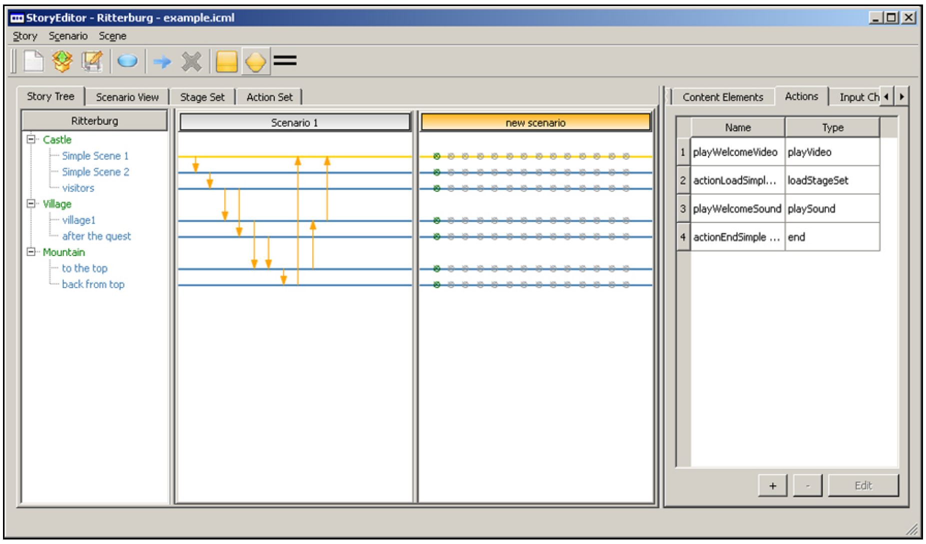
StageSet. A stageSet is the configuration of content elements for a specific scene. StageSets are defined hierarchical, inheriting all content elements from ancestor scenes. Size and dimension of stageSets (2D or 3D graphics, mono, stereo, 5+1, 3D-sound) depend on the media elements and on the physical input and output facilities available in the targeted performance setting.

Asset. The lowest level primitive of the story structure are assets. These are typically pictures, text, sound, animation, AV-clips, or virtual characters. Assets are referenced by the actions. Named assets may be pre-arranged into stageSets, which is very useful to generate a preview of the scene.

2.2 Views

Because of the various tasks involved when authoring an interactive story, different views of the same story are offered to the author.

The scene-tree view shows the static structure of the story, composed of scenes, which are grouped into so-called complex scenes, comparable to the act vs. scene structure of a stage play (drama-like structure). This view gives the most administrational view, comparable to the File-Explorer in Windows Systems.

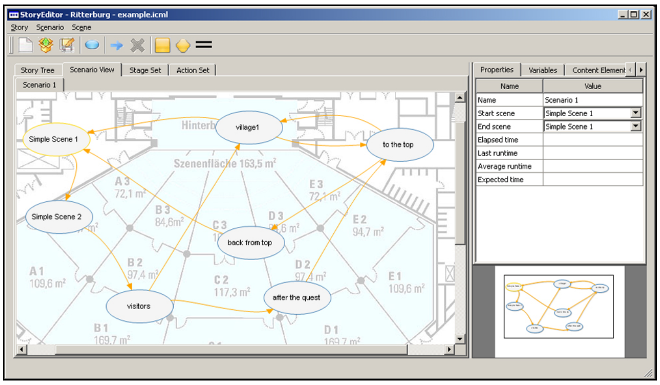


	Name	Type
1	playWelcomeVideo	playVideo
2	actionLoadSimpl...	loadStageSet
3	playWelcomeSound	playSound
4	actionEndSimple ...	end

Fig. 2. Story-Tree View (Comparing different Scenarios)

The scene graph view (see figure 3) shows the possible transitions between all scenes. Transitions are grouped into scenarios, where each scenario is one possible navigation structure for the story. This interface shows the same transition network as the scenario preview in the tree-view, but here with scenes represented as elliptical symbols instead of swim-lane lines.

Scene symbols can be freely positioned by drag-and-drop moves, and Transitions are built as dynamic connectors, which can be labeled with a text. The lower right area provides a magnifying rectangle. The content inside the rectangle is shown in the main view. A background image (wallpaper) can be loaded (e.g. a floor plan of an exhibition), and the size of the work area is automatically adjusted to the image size.



Name	Value
Name	Scenario 1
Start scene	Simple Scene 1
End scene	Simple Scene 1
Elapsed time	
Last runtime	
Average runtime	
Expected time	

Fig. 3. The Scene Graph View shows all the scenes and transitions of one scenario, freely arranged on the canvas, in front of an optional background image (in this example, the floor plan of some building)

3 Stage Editor

The stage editor has the purpose to do the layout or setting of each scene of the story. It supports 2D and 3D scenes and also a mix of both. The latter are scenes where a 2D overlay is shown on top of a 3D scene. When navigating through the 3D scene the overlay stays unchanged, it is fixed to the virtual camera.

When editing a 2D scene the 3D space stays empty and all media objects are fixed to the overlay plain. For the author this will look and feel like a conventional 2D editor. With standard graphic accelerators it is more efficient to render everything in 3D.

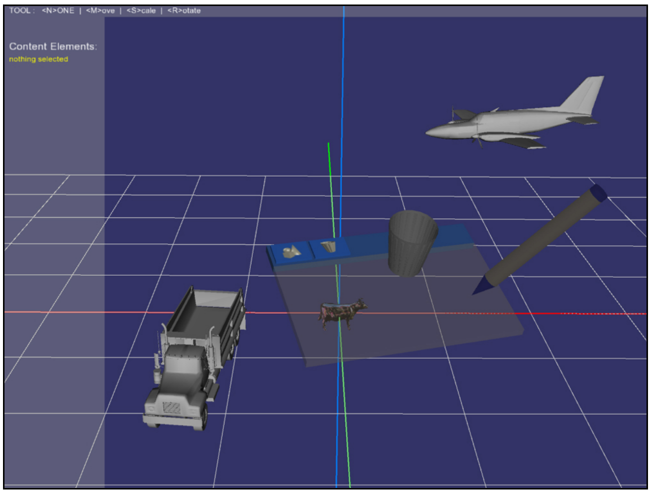


Fig. 4. 2D and 3D Rendering is available in the Stage Editor

The goal is that all supported media types (assets) can be viewed within the stage editor. Since it is based on OSG this means that each media type has to be rendered in 3D using methods provided by this real time graphics framework. While 3D objects are naturally supported, 2D media types need special solutions. Supported 2D media types are: texts, HTML data, images, vector graphics, sounds and movies.

Each imported media element is put into a container class, which provides methods to react on **user interaction**, which will be linked in the *Input Wiring Editor* with the interaction that are possible thanks to the various devices available.

4 Conclusion

Contrarily to many interactive authoring tools, which build the authoring part first and then progressively extend its experiencing capabilities, the U-CREATE prototype is aiming at building a tool to assist the creation of interactive contents for advanced experiencing systems that do already exist. Doing so, its challenge is to cope with all the already existing features of these systems and to come up with a solution that conceptualizes them all into one unique framework.

References

1. Maribeth Gandy, Steven Dow and Blair MacIntyre. "Prototyping Applications with Tangible User Interfaces in DART,The Designer's Augmented Reality Toolkit." Positional paper at: Toolkit Support for Interaction in the Physical World Workshop, held at: IEEE Pervasive Computing 2004, April 20, 2004.
2. S. Sauer and G. Engels, Extending UML for modeling of multimedia applications, Proc. IEEE Symposium on Visual Languages, Tokyo, September 13-16, 1999.
3. S. Sauer and G. Engels. UML-based behavior specification of interactive multimedia applications. In: Proc. IEEE Symposia on Human-Centric Computing Languages and Environments (HCC'01), pages 248--255, 2001.
4. F. Ledermann, An Authoring Framework for Augmented Reality Presentations, TU Wien, 2004
5. Mateas, M. & Stern, A.: Procedural Authorship: A Case-Study Of the Interactive Drama Façade. In: Proceedings of Digital Arts and Culture (DAC), Copenhagen (2005)