

StoryML: Enabling Distributed Interfaces for Interactive Media

Jun Hu

Dept. of Industrial Design

Eindhoven University of Technology

Den Dolech 2, 5600MB Eindhoven, The Netherlands

+31 (40) 2475360

j.hu@tue.nl

Media Interaction Group

Philips Research Laboratories

Prof. Holstlaan 4, 5656AA Eindhoven The Netherlands

+31 (40) 2744846

hu@natlab.research.philips.com

ABSTRACT

This paper introduces an adaptive architecture for presenting interactive timed media onto distributed networked devices. The interactive presentation is documented in StoryML, an XML-based language, and presented to multiple interface devices organized in an agent-based architecture.

Categories and Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – *Artificial, augmented and virtual realities*;

H.5.1 [Artificial Intelligence]: Distributed Artificial Intelligence – *Multiaгент systems, intelligent agents*

General Terms

Design, Experimentation, Human Factors.

Keywords

Distributed media, distributed interfaces, architectures.

1. INTRODUCTION

New technologies, such as MPEG-4, SMIL, enable delivering interactive multimedia content to the consumers' homes. Using physical interface devices, a more natural environment in which real-life stimuli for all the human senses are used, will give people more feeling of engagement [4]. The work presented here focuses on how to structure the system and content to support distributed interfaces for timed media applications. The carrier for this work is an interactive storytelling application (TOONS) for children (age 8-12) in the NexTV project sponsored by the European Commission under the IST-programme.

Figure 1 shows the conceptual model of an interactive story. This model consists of storylines and dialogs. The storylines comprise the non-interactive parts in the video stream. The dialogs comprise the part in which the users make decisions to switch between storylines. The application plays the media objects on and gets user responses from several networked devices in the user's environment.

According to the user requirements, the system architecture should emphasize and support the following: (1) Distributed interfaces require that not only the content presentation, but also the user input and control are distributed over the networked environment. (2) The story will be presented to different user environments which configurations may change during the time.

(3) Not only the media, but also the user interactions are timed and should be synchronized.

We developed a framework for presenting interactive story onto distributed devices. The TOONS application was build based on this framework, presenting the story to a television screen, a toy robot, ambient lights and surrounding audio devices.

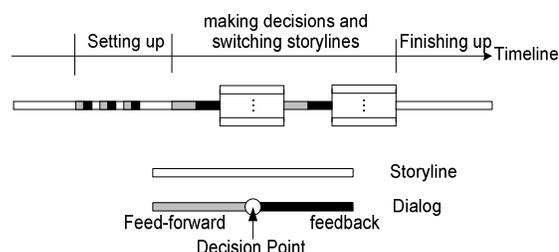


Figure 1. Conceptual model of TOONS

2. WHY NOT MPEG-4 OR SMIL

The TOONS application requires the documentation technology to deal with the distribution of the interaction and the media objects over multiple presentation devices. The Binary Format for Scenes (BIFS) based MPEG-4 [2] documentation emphasizes the composition of media objects on one rendering device, but not many. SMIL 2.0 [3] introduces the *MultiWindowLayout* module, which contains elements and attributes providing for creation and control of multiple top level windows. This is very promising and comes closer to the requirements of distributed presentation and interaction. Although these top level windows are supposed to be on the same rendering device, they can to some extent be recognized as interfacing components with the same capability, but not different ones.

3. STORYML

StoryML is developed for scripting the interactive story [1]. The unknown configuration of the environments requires the presentation to be described at a high level of abstraction.

We model the interactive media presentation as an interactive *Story* presented in a desired environment (called a *Theater*). The story consists of several *Storylines* and a definition of possible user *Interaction* during the story. User interaction can result in switching between storylines, or changes within a storyline. *Dialogues* make up the interaction. A dialogue is a linear conversation between the system and the user, which in turn consists of *Feed-forward* objects, and the *Feedback* objects depending on the user's *Response*. The environment may have several *Interactors*. The interactors render the media objects. And finally, the story is rendered in a Theater. Figure 2 shows the object-oriented model of the StoryML.

3.1 Interactors

An Interactor is a self-contained autonomous agent which has an expertise of data processing and user interaction. It is able to abstract the user inputs as events and to communicate with other interactors. An Interactor element in StoryML has a “type” attribute. It defines the basic requirement for the capabilities of the Interactor, at an abstract level.

3.2 Theater

A *Theater* is defined as a collection of several these cooperative Interactors. The Theater assigns different tasks to each of the Interactors according to the definition of the script, for example, rendering media objects, reporting the user responses during different periods of time.

As mentioned, the Theater is a *desired* environment in which the Story will play. The target environments might be different, therefore there will be a mapping problem, that is, how to satisfy the abstract requirement to the users’ physical environments.

3.3 Media Objects

Storylines, feed-forward and feedback components in a Story can be recognized as time-based media objects. A media object can be rather abstract, for example, expressions, behaviors, and even emotions, can be defined as media objects as long as they can be recognized and rendered by any Interactors.

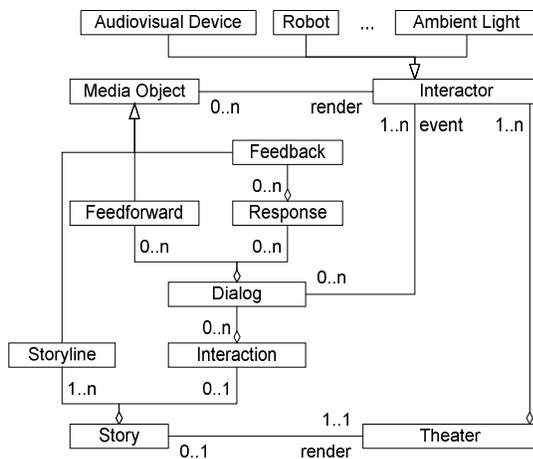


Figure 2. Object oriented model of a presentation

4. THE STORYML PLAYER

Figure 3 shows the hierarchical structure of the StoryML player. The content portal establishes the connection to content servers and provides the system with timed content. The content pre-fetcher overcomes the start latency by pre-fetching a certain amount of data and ensures that the media objects are prepared to start at specified moments. An XML parser first parses the StoryML document into Document Object Model (DOM) objects and then the StoryML parser translates the DOM objects into internal representations. The Timeline controller acts as a timing engine that plays an important role in media and interaction synchronization.

The bottom level agents indicate different physical interactors. For each physical agent, there is an intermediate virtual interactor connected as its software counterpart. All virtual interactors are managed by an Interactor manager.

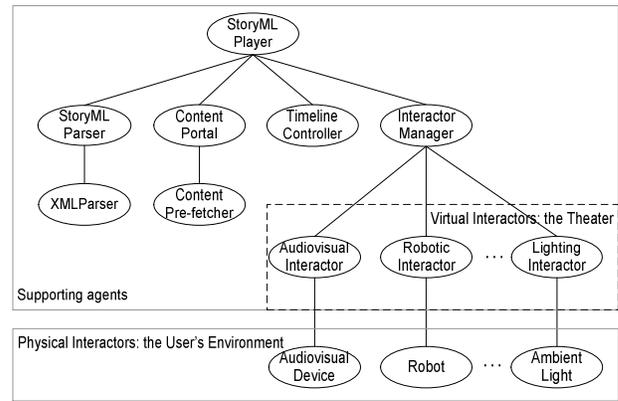


Figure 3. the StoryML player

As the hierarchy shown in Figure 3, it is the responsibility of a virtual interactor to find out if in the user’s environment there is a physical device available that can be a replacement. This is done by broadcasting a query to the network and every networked device will return a token together with a description of its capability. If such a device is found, the media presentation and interaction tasks are then transferred to the device identified by its token.

The virtual interactors maintain the connection with the physical ones by sending handshaking messages on a regular basis. Once no response is from the physical device, the virtual interactor will try to find another one in the environment. If failed to find any corresponding physical devices, the virtual interactor will present itself onto an audiovisual device in order to make the media presentation and the user interaction possible again.

Therefore the mapping from the desired Theater to the user’s physical environment is done autonomously and dynamically by the virtual interactors without the upper layer manager intervention.

5. CONCLUSIONS

We presented a framework which allows the presentation of multimedia contents in different Environments. StoryML plays an important role in the framework: (1) StoryML allows the separation of the content from the concrete physical devices. (2) A StoryML document specifies abstract media objects at a high level and leaves the complexity to the implementation of the rendering interactors. (3) The StoryML player supports the automatic mapping of the same document to different environments with dynamic configurations.

6. REFERENCES

- [1] Hu, J. (2001). *Distributed Interfaces for a Time-based Media Application*. Post-master thesis, Eindhoven University of Technology, Eindhoven.
- [2] Koenen, R. (1999). MPEG-4: Multimedia for Our Time. *IEEE Spectrum*, 36(2), 26-33.
- [3] Slowinski, M., Kennedy, T. & Kennedy, T. *SMIL: Adding Multimedia to the Web*. Sams, 2001.
- [4] Strommen, E. (2000). Interactive Toy characters as Interfaces for Children. In E. Bergman (Ed.), *Information Appliances and Beyond: Interactive design for consumer products* (pp. 257-298). New York: Morgan Kaufmann Publishers.