An HTML like language supporting time-dependent transmission of hypermedia

C. Bouras¹, V. Kapoulas¹, P. Spirakis¹, A. Tatakis²

1. Computer Technology Institute, Greece
2. Expertnet SA, Greece
e-mail: bouras@cti.gr

Introduction

Multimedia information systems are characterised by the need to compose and represent data of different types and formats. In order to exchange multimedia objects in a distributed environment and to provide a multimedia application with the ability to reconstruct the original spatio-temporal presentation scenario of the object's component, there is the need for the use of a flexible model. This model must offer features such as the association between the media contents and their presentational characteristics, the spatio-temporal presentation of the involved media data, etc. Several models have been proposed for the integrated modeling of multimedia documents. HyTime and MHEG seem to address issues such as the above, yet they cannot be easily implemented. The Dexter Reference Model [2], defines three levels; the storage layer that describes how nodes of information are joined via hyperlinks to construct an information web; the run-time layer that describes the presentation of a hypermedia document and its behavior to real time user interactions; the within-component layer that refers to the document's components content and structure. The Amsterdam Hypermedia Model (AHM) [3], added to the Dexter model the notions of time, high level presentation and link context. CMIFed [4], was based on the AHM and aimed at providing authoring and presentation environments for hypermedia documents.

We propose a model for structuring multimedia documents that addresses the above mentioned features in a unified way. The core of the model is a
hypermedia presentation markup language, influenced by HTML. The
language supports several tags and keywords that can be used to define the
layout structure of a hypermedia document. Its most prominent characteristic
is that it supports embedded timing information that can be used for the
construction of a "playback" schedule for the various media contained in the
hypermedia document.

Description of the language

As it was mentioned above the language resembles HTML and offers the
ability to embed in it timing information that can be used for the proper
playback of video and audio data and the proper presentation of image data.
A short description of the corresponding language elements follows. The
description is in BNF, terminal symbols are in capitals and non-terminals in
brackets.

```plaintext
<Image>::= IMG<ImgOptions><Source><Id><Note>END_IMG
<Audio>::= AU<AuOptions><Source><Id><Note>END_AU
<Video>::= VI<ViOptions><Source><Id><Note>END_VI
<Audio_Video>::=
   AU_VI<Au_ViOptions><Au_ViSource><Au_Vi_Id><Note>END_AU_VI
<ImgOptions>::= <TimeOption>
   |<TimeOption><OtherImgOptions>
<AuOptions>::= <TimeOption>
   |<TimeOption><OtherAuOption>
<ViOptions>::= <TimeOption>
   |<TimeOption><OtherViOptions>
<Au_ViOptions>::= <SyncOption>
   |<SyncOption><OtherAu_ViOptions>
<TimeOption>::= STARTTIME STRING
<SyncOption>::= STARTTIME STRING STARTTIME STRING
<OtherImgOptions>::= HEIGHT STRING WIDTH STRING
<OtherAuOptions>::= /* empty for now */
<OtherViOptions>::= /* empty for now */
<OtherAu_ViOptions>::= /* empty for now */
```

The language has been developed in order to enable us to build a
client/server application that will allow the conduction of hypermedia tutorials.
In these tutorials the student connects to the server and requests a tutorial
which is structured as a hypermedia document. On the client's side the
description of the document in the above language is received and it is
processed in order to extract the embedded timing information. During the
processing every media stream with timing information is recognized by its
Corresponding language element and the timing information is utilized in order
to create a playout schedule by which the data will be presented to the user.
In order to ensure the proper playback of the various media a buffering
scheme is used. This scheme uses a number of multi-thread buffers whose
lengths are calculated with the use of several parameters such as network
delay, packet length, encoding standard, etc. When there are delays recovery
actions are taken such as duplication of frames, reduction of media
presentation rate, etc.

Conclusions

In order to test the validity and the performance characteristics of the above
language we have implemented a pilot application. The server of the
application, [1], has been implemented under Unix and the browser has been
implemented under both Unix and Windows 95. The network handling
routines have been implemented with the use of the Real Time transport
Protocol (RTP) [5], and its control protocol RTCP. Both protocols make use of
the functionality offered by the TCP/IP protocol suite. RTCP is used for the
provision of feedback information that allows the monitoring of the service
quality, while RTP is used for the transmission of the actual data. We are
currently conducting tests, over an FDDI ring, in order to measure the
performance of the above design and implementation and especially that of
the RTP and RTCP protocols. Also we are in the process of implementing
elements of the language that are currently unsupported.

References

[1] C. Bouras, V. Kapoula, D. Miras, V. Ouzounis, P. Spirakis, A. Tatakis,
"On-Demand Hypermedia Multimedia Service over Broadband Networks".
Fifth IEEE International Symposium on High Performance Distributed
Computing, Focus Workshop on Multimedia Collaborative Environments, 6-9 August 1996, Syracuse, New York, USA.


