MPEG-4 based framework for game engines to handle virtual advertisements in games
Mohammed Chalil, Sreekumar K.P, Manoj Sankar
mdchalil@nestec.net, kpsree@nestec.net, msankar@amcomm.com
Network Systems and Technologies Ltd,A-3, Periyar, Technopark,, Trivandrum, Kerala, India.

ABSTRACT
Online gaming is a convergence industry segment with a proven revenue model and an existing user base. Inserting advertisements in games provides another revenue opportunity for the game designer and service providers. Virtual advertisement is the method of inserting advertisements into the game in a non-obstructive way. For online game service providers, changing these virtual advertisement scenes based on time/player-profile is required to attract local sponsors. This requires replacing objects in game scenes with dynamic advertisement objects of similar image attributes. Since there is no standardized framework for this, the service provider needs to know inside details of the game engine. This paper proposes an MPEG-4 based framework for virtual advertisements in online games. During online gaming, the game server will send the virtual advertisements (Dynamic Ad Objects) to the game machines. In the game machines the Dynamic Ad Objects will be used to replace the Static Ad Objects already present in the game. This will allow the game server to send different Dynamic Ad Objects to each player. Also the computation load for the game machine will be minimum because of FlexMux support of MPEG-4.

INTRODUCTION
Online game is a segment with proven revenue model and has an existing user base and interactive online games will definitely play a key role in futuristic game design. Inserting advertisements in games provides another revenue opportunity for the game designer. If the network and application interface. If the game engines are based on these standards, service providers can support virtual advertisement without knowing the exact internals of the games engine.

MPEG-4
MPEG-4 is an emerging international standard based on the encoding of audiovisual data using object based coding and scene description techniques. Three layer architecture of MPEG-4 as defined by ISO/IEC 14496 series is illustrated in Figure 1. MPEG-4 compression layer offers a very efficient data representation, MPEG-4 synch layer advertisements are inserted as if it is part of the game, it is called virtual advertisement. This ensures the advertisement scene will not intrude original game scene, at the same time getting attention of player by dynamically changing the scenes.

The concept of virtual studio is already been used in broadcasting [1]. There have been proposals to insert virtual advertisement scenes in broadcast images [2]. These methods employ raster-based approach, which requires complex pattern matching algorithms for identifying objects to be replaced. Also for the replaced objects image transformation has to be performed to match the original scene attributes, like tilt of background image, viewing angle, other parameters.

Modern game engines are based on vector graphics. Object identification and substitution will be an easier task compared to raster-based approaches. In the case of online games, synchronized scene changes and associated information are transmitted to game machines from a central server. The game machine will compose the scenes using the above information. If virtual advertisement has to be inserted in this game, game server needs to modify the scenes send to the game machine. But this requires processing depending upon the game logic, and the format used in the specific game. From the providers’ point of view, this is a handicap as it is impossible to modify virtual advertisements without specific software from each game developer.

MPEG-4, as defined by ISO/IEC 14496, specifies a series of standards for object-based 2D/3D encoding, address the synchronized transportation and storage of data. MPEG-4 uses Binary Information for Scenes (BIFS) for scene description. Flexmux and upstream capability of MPEG-4 enables implementation of highly interactive applications like network-based games.

Compression Layer
ISO/IEC 14496-2 specifies object-based hybrid natural and synthetic coding standard, which specifies the technologies enabling the functionalities such as content-based interactivity, efficient compression, error resilience, and object scalability. Standard addresses coding of both
natural and synthetic scenes and music. MPEG-4 based multimedia scene encoding sequence is illustrated in Figure 2. The scenes of the game are encoded as A/V Objects (synthetic or natural), Scene structure, and Object Descriptors. The game engine will decode and combine the A/V objects as per Scene Structure and Object Descriptors to produce the original scene.

![Figure 1 Architecture of MPEG-4](image)

**BIFS**

BIFS inherits its basic structure from Virtual Reality Modeling Language (VRML). The compressed objects on the scene are represented as a set of nodes and scene graphs to describe its synchronized behavior and interaction. When compared to VRML, BIFS has additional features like Timing Model, Protocols to modify/animate the scenes in time and 2D graphics support. Flexible Timing model of BIFS helps to maintain better synchronization between objects. Here, relative timing is used rather than absolute timing and it is possible to specify/change parameters like delay, end-time, stretch, etc. It is possible to change the time base of an object clock so that it can have time dilation or contraction with respect to others.

![Figure 2 MPEG-4 Compression Layer](image)

**DMIF**

The Delivery Multimedia Integration Framework (DMIF) is a general application and transport delivery framework specified as per ISO/IEC 14496-6. The main purpose of DMIF is to hide the details of the transport network from the user, as well as to ensure signaling and transport interoperability between end systems[3, 4]. This layer also has provision for frame error checking. This makes MPEG-4 different from its predecessors where presentation time stamps are in the bottom most layer.

**DAI**

In order to keep the application unaware of underlying transport details, MPEG-4 defined an interface between applications and DMIF called the DMIF application interface (DAI). The DAI provides the required functionality for realizing multimedia applications with QoS support irrespective of transport protocol.

**PROPOSED FRAMEWORK**

This framework exploits some of the MPEG-4 features for synchronized A/V object replacement enabling seamless modification of virtual advertisements in online games.

The game designer shall identify the portion of scenes where the virtual advertisements can be inserted. These scene portions shall be encoded as a separate object called Static Ad Object (SAO) and multiplexed along with the game scene objects. When this game is played, the advertisement scenes will be SAO and will be same each time the game is played.

![Figure 3 The System architecture](image)

The designer shall describe all the SAOs present in the game and their associated information; this is called Game Ad Profile (GAP). The GAP shall be included in the
The replacement scheme for game engine is illustrated in Figure 4. MPEG-4 engine will decode the streams and give the decoded information to the game engine. In addition to this, the game engine can take other game control file directly from local resources.

The MPEG-4 engine interacts with network and local storage through respective DMIF to get the MPEG-4 stream. The MPEG-4 engine can change the video footage dynamically by modifying the elementary streams and associated object descriptors. The MPEG-4 engine can also modify the BIFS to produce composite MPEG-4 stream. This is then fed to game engine as the multimedia content of the game. An example to illustrate this is shown in Figure 5.

Features of FlexMux are explored to replace an SAO with DAO. Here, the stream from the local storage is de-multiplexed and identified SAO (based on ES_ID) is replaced with DAO that is taken from the stream coming from network. Then the elementary streams are re-multiplexed to another MPEG-4 stream, which produces the desired video footage.

**ISSUES AND POSSIBLE SOLUTIONS**

Since the DAO is send through network in real time, excess of delivery jitter of related DAOs will degrade alignment and positioning of ad scenes due to non-availability of objects at the required instance. Since timestamp interpretation is based on the local clock, time-base synchronization between the server and the game machine needs to be ensured. Substantial difference in the time-base will lead to object misalignment and buffer overflow.
Flexible Multiplex (FluxMux) Layer of MPEG-4 gives provision for grouping Access Units (AU) with similar QoS or other resource requirements. By grouping of related DAOs and BIFS, delivery jitter of related objects can be minimized. Synchronization of interaction between the AU is achieved with the help of Object Clock Reference (OCR) and time stamps (DTS and CTS) associated with each access unit. This relative timing will help the synchronization of related objects.

The strategy adopted to keep the timing and synchronization intact is illustrated in Figure 6 Local system clock is gen-locked to the PCR of the MPEG-2 TS[8]. This can also be gen-locked to OCR of the incoming DAO, if PCR is not available/accessible. In this way the Timing and Synchronization Logic (TSL) will adjust its time notion (Local time base) to that of the server using PCR ticks in the DAO stream. The locally stored game streams including SAO will be pulled from the file based on the local time base generated by the TSL.

The TSL will modify the OCR based on the changed time base. The TSL also will modify the FlexTime parameters based on FlexTime parameters of SAO and DAO so that small drift in time does not affect the synchronization. The TSL, which is running in tandem with DAO source, is used to generate pull request for accessing local storage. This strategy will minimize the buffer requirement at the game machine.

**CONCLUSION**

In this paper we presented a generic framework for online games to support virtual advertisements. The support provided in MPEG-4 for object based coding and relative timing is used to develop the framework. This framework replaces static Ad objects in game with dynamic Ad objects send from online game server. It will enable the service provider to change or insert virtual advertisement scenes onto any game engine supporting this framework. The basic advantages of this framework are (i) it is independent of the actual game software (ii) it is simple compared to existing techniques to implement in the game machine.

One of the major challenges with the online replacement of Ad objects has been the synchronization between the related Ad scenes. The issue has been addressed by exploiting the provision for grouping and relative timing of MPEG-4 AUs. The requirement of time-base synchronization of the game engine with that of the server is also addressed. Gen-locking the game-engine with that of server and using this time base to pull the static streams from the local storage will ensure seamless functioning of the system. The TransMux feature of MPEG-4 will ensure the compatibility with any suitable existing protocol stack such as (RTP)/UDP/IP, (AAL5)/ATM, or MPEG-2 TS over suitable link layer, due

**REFERENCES**

1. Marc Price, Graham Thomas, “3D virtual production and delivery using MPEG-4”, BBC Research and Development, UK
8. Jae-Gon Kim and Jinwoong Kim, "Design of a Jitter-free Transport Stream Multiplexer for DTV/HDTV Simulcast", ETRI, KOREA.