

Melisa - A Distributed Multimedia System for Multi-Platform Interactive Sports Content Broadcasting

Emmanuel Papaioannou
Intracom S.A.
paem@intracom.gr

Kostantinos Karpouzis
National Technical University Athens
kkarpou@softlab.ece.ntua.gr

Philippe De Cuetos
ENST
philippe.de-cuetos@enst.fr

Vasilis karagianis
Intralot S.A.
karagiannis@intralot.com

Hervé Guillemot
ONDIM
guillemot@ondim.fr

Athanasios Demiris
Intracom S.A.
dema@intracom.gr

Nikolaos Ioannidis
Intracom S.A.
nioa@intracom.gr

Abstract

The new generation of television viewers is currently being confronted and becoming acquainted with a series of technological developments in the realm of consumer electronics and gaming that raise their expectations for similar advances in TV broadcasts. The MELISA platform aims at the cross-media broadcasting of sports events featuring interactive advertising and sports-related games over digital television and next generation mobile network infrastructures. The platform provides services for optimal presentation of complex interactive real time video content, for advertisement and an advanced real-time gaming (betting) engine. The MELISA server receives information from geographically separated Betting, Advertisement and Video Capturing Servers that enforces the notion of a Distributed System design for information exchange. The resulted Business model offers great flexibility in system configuration both in scalability and physical layout configuration.

1. Introduction

In general, a multi-platform content delivery system imposes a number of specific requirements on common media- and communications-related tasks. Perhaps, the most important of them is the ability to cater for multiple, diverse reception and playback platforms. Indeed, the same raw multimedia information that can be played back using high -end PCs must be adapted to become suitable for PDAs and set-top boxes or even multimedia-capable mobile phones.

A requirement that is inflicted mostly on the content adaptation and transmission components of the platform is that the authoring process must be as

automated as possible, while taking into account variable characteristics, such as network resources and protocol provisions and users experience with the end-user devices.

The Melisa platform is an end-to-end system that allows the creation, broadcasting and viewing of enhanced, multi-platform, video content in MPEG2 and MPEG4 format, extended with notions of interaction and e-commerce. The Server Platform provides a range of tools for real-time capturing, management and processing of XML information and video streams, from a number of input sources, while encoding different versions adapted to different terminal capabilities.

In addition to media preparation, MELISA provides the infrastructure necessary to process various transactions of the users related to sports broadcasts in real-time and the applications demonstrating the value of this service for personalized interactive advertisement, m-Commerce and wagering.

The MELISA clients, Set Top Box and PDAs provide the viewer with enhanced interactive content for real-time betting, advertising and in game information, enhancing the sports viewing experience.

In this paper, we describe a general overview of the MELISA system architecture and go into details about specific components of the integrated system. Section 2 describes the state of the art regarding some individual notions integrated in MELISA; these notions are presented in detail in Section 3.

2. State of the Art

2.1. Video enhancements

Since the raw video material that is used in the MELISA system comes from live sports broadcasts, an

important feature that end-users have come to consider as imperative is the provision of visual aids or enhancements. Usually, these enhancements provide numerical or, in general, quantitative information about specific measurements or events, such as the distance of ball placement to the goal line during a free kick, or a coarse indication of the distance during a long jump in athletics, and can go so far as to display a virtual racing track indicating the real-time positions of cars during a racing event. Another important aspect that is more closely related to the business model of such an integrated system is that of virtual advertising, where synthetic boards are placed in specific views of the sports venue, displaying content that can be fitted to suit different occasions or even broadcasters.

A significant number sport coverage enhancement application derives from the technology known under the generic term of Virtual signage. Virtual signage consists in inserting in real-time location into television broadcasts computer-generated objects at a pre-defined physical location (“tied-to-the field”). The insertion is performed so as to give the impression that the object actually exists. The virtual signage technology has been first commercialized for virtual advertising in the perspective to create sponsorship added value. Indeed insertion of virtual advertising allows geographical targeting (different messages according to area of broadcast), timesharing (several advertisers on the same space on a time basis) and enhanced visibility (utilization of new surfaces such as a field, 3D animated advertising).

In order to give the impression that an inserted virtual object actually exists, (i) its position and shape in the video pictures must be adjusted according to the camera position and (ii) if a real object travels in front of a section of the virtual object, this section must be occluded. To achieve real-time insertion, each broadcast camera is instrumented with very precise encoders. For every field captured by the camera (i.e. 50 or 60 times per second), these encoders transmit the pan, tilt, zoom and focus positions of the camera. A computer at each camera reads the encoders and transmits these readings to the production truck. A simple 3D model of the field or of the surface where the virtual object is inserted must be conducted. This 3D model together with the camera position data allows at every instant to relate a physical point on the field with a pixel location in the video picture. This mapping tells how the virtual object should be reshaped and where it should be placed in every transmitted picture. Pixels that, according to the calculation, would be in an unobstructed view of a real advertising location are then classified between obstructed and unobstructed. Pixels classified as

obstructed are left unchanged while others are replaced by the color of the virtual object at the corresponding location. This makes the virtual object visible where the space is visible and hidden where the space is obscured, just as a real object would be. This sequence of operations occurs in real-time. Current real-time insertion systems require the cameras to be mounted on tripods.

2.2. Object tracking

Object tracking is of particular interest for sport coverage enhancements since it opens a large set of applications ranging from simple speed measurement to 3D virtual replay. Indeed, related issues are still very much open in the research community; thus, we will concentrate here on provisions and concepts in commercial systems, since they can provide for real-time and dependable data, which is fundamental in live or replay broadcasts.

An interesting application of object tracking is the display of the trajectory of a tennis ball or another small moving object, either superimposed on a video sequence or in a 3D mockup of the sports venue, with the choice of different playback speed or viewing angle. To achieve this, two dedicated non-moving cameras (usually referred to as “pointer cameras”) are positioned so as to cover the entire ball trajectory from two sufficiently different angles of view. The cameras take around 16 pictures of the ball along the way. On each picture the ball is automatically tracked and isolated from the background. The ball is then localized in the 3D space through triangulation at each of these 16 instants. Triangulation consists in computing the intersection of the two lines of position (LOP) corresponding to the position of the ball in each of the pictures. Indeed each pixel in the camera view corresponds to a single line in the 3D space. The claimed precision is of within one inch. The system operates near real-time and allows the virtual replay of a pitch one second after it occurred.

This concept can be extended to track larger objects, i.e. football players on the field. Systems built on this basis detect the position of every player on the field and build an interactive database that can be consulted at any time. Data such as average speed, player activity, ball possession or field coverage can be readily displayed at any instant of the game, aiming at helping broadcasters and coaches, besides catering for end-users. Besides football, this technique has also been applied to tennis matches, where the environment and object movement are much more controllable. Here, tracking is performed on the basis of video

streams recorded by 2 to 4 dedicated color cameras. The tracking algorithm first performs foreground/background segmentation by simple difference between successive frames. In the regions classified as foreground, local features are identified and matched across time. The trajectories of local features are then clustered so as to obtain a stable trajectory. This sequence of operation is repeated 20 times per second.

Besides image-based tracking applications, microwave or optical sensors and unobtrusive transmitters can be used to provide an object's or athlete's position over time. Such set-ups can be placed on football players' helmets, to provide their position and speed in the field, on racecars, or even sailing boats, intertwined with GPS data. The advantage of this technology over image-based tracking lies in the precision of the position data, which in turn can be used to derive precise measurement of instantaneous speed and acceleration. In football, for instance it allows you to know the instantaneous speed of a player or to quantify the force of a collision received by a player during an encounter with an opponent. A serious disadvantage of this sensor-based technology is the necessity to equip the object to track with sensors. This requires convincing the responsible sport federation and will not be applicable in the medium-term to track objects such as balls.

2.3. Set-top box technology

The first goal of digital TV has been the technical evolution of the basic existing TV services integrating new technologies. The first applications have been to offer new features like the availability of more channels for the same physical (satellite, cable or terrestrial) bandwidth or the introduction of access control for pay TV and pay per view. Another step currently under way is the integration of basic Internet access and complementary functions for web browsing and electronic commerce. Due to the size of the related market and thanks to many standardization efforts (for example in MPEG, DVB, ATSC or ATVEF, etc.), set-top boxes are evolving in terms of capacity and functionality. It is clear that set-top box architectures will strongly be modified in the 2-4 coming years. For example the addition of disk, of home network capability, of general or specialised processing (e.g. stream processing, image processing, supporting Java applications, etc...) go in this direction.

Nowadays, the trend is to integrate most of the blocks around the host processor (except the memory

and the RF reception). The different decoders are realized with dedicated hardware blocks. If we consider for instance the MPEG-2 video decoding process, the IDCT2 block will be used many times per frame and so it must be optimized to reduce the die size of the chip. The best way to optimize this part is to design a dedicated hardware block for this specific data processing. The statement can be made that nothing can be better than a dedicated circuit. This assertion is always true when considering a well identified application such as a simple MPEG-2 decoder with specific video coding on a well-known transmission support. When examining the future of Set-Top Boxes it is not possible to keep the same model. Because of the variety of the possible decoding processes, it is not realistic to design and integrate a complete decoding block for each of them. Another kind of processing management must be used.

In general, the evolution of the digital world leads to a convergence between the PC-world and the TV-world. For instance, Internet access will be available with next generation of TVs and Set-Top Boxes. The evolution of transmission data rates allows providers to propose new kinds of services via the satellite or the telephone line. Regarding interactivity, consumer behavior will probably change with the possibilities provided by MPEG-4, going from a passive position to an active one. The architecture of the Set-Top Box will have to manage the context change, and activate other decoders, requiring flexibility to be able to achieve this in parallel while keeping the same level of quality. Similarly, the concept is to be able to plug any kind of equipment in any system without any problem of compatibility. This requires some dedicated interfaces and some flexibility to manage each kind of transmission format for instance.

3. Distributed System Description

The Melisa Platform can be conceptually divided in four layers (Figure 1). There are four system modules that are responsible for providing the Melisa system with content for Broadcasting. The Lottery and Advertising systems are physically separated from the Melisa Application Server. The communication with the Melisa server is achieved using XML based messages through http connections.

The Capturing and Image Processing unit provide the Melisa system with MPEG-2 format Video Content and Visual Enhancements. The Visual Enhancements offered, range from multiple objects tracking [1][2] for Active Advertising and provision of statistical Information, to distance lines, offside lines, etc. The

modules communicative with the Work Flow Control Unit (Figure 2) using http connection and Java's Remote Method Invocation (RMI). The Melisa Application Server contains a collection of tools for off-line and online Interactive content authoring, and broadcasting. To maintain flexibility in system configuration each module has been designed as a separate application, communicating with the rest of the System using java RMI.

The persistent layer of the system, the Melisa Database, is clearly distinguished from the Melisa Application server. Database access is performed only through the Database manager providing complete database encapsulation separating the Database from the

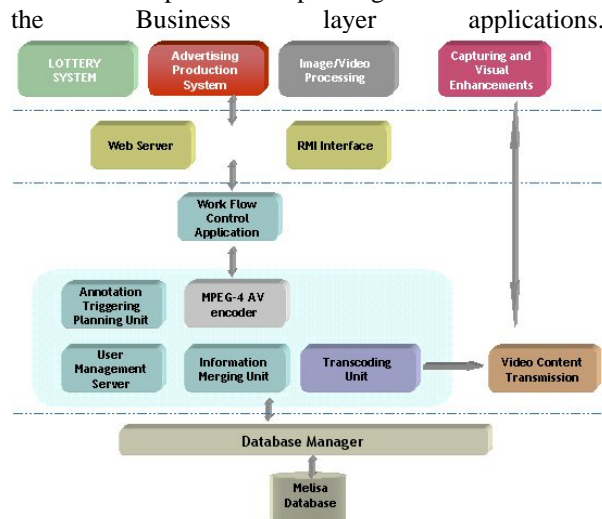


Figure 1. Melisa Platform Logical Architecture

3.1. System Architecture

The Workflow Control Application (Figure 2) is the control module of the MELISA Sender side, controlling data flow and checking on the progress of the different jobs. The Workflow Control Application handles all requests and events during the televised sports event originating or directed to the different service provider modules. Its major role is to orchestrate the overall production and service provisioning process at the MELISA sender. The Workflow control receives information from the Betting, Advertising and Capturing units, and initiates the process of converting the Information into MPEG4 [3] interactive content.

The Information Merging Unit receives the information from the Workflow Control, and based on predefined Templates passes the data to Transcoding Unit. The Transcoding Unit generates an MPEG-4 stream that contains all the Interactive elements and

Visual enhancements ready to be broadcasted. Depending on the targeted platform, Set Top Box or PDA, the MPEG-4 stream is multiplexed into MPEG-2 SPTS streams or converted into RTP streams respectively.

The Annotation and planning tool is an authoring tool that allows the operator to schedule and plan an event in advance. Information concerning the Teams and Players involved in an event, Event scheduling, Statistical Information, and Broadcasting Information is entered in the system for use during broadcasting. The annotation tool allows the operator to monitor an event and record any significant event as it appears. The tool automatically notifies the Workflow control,

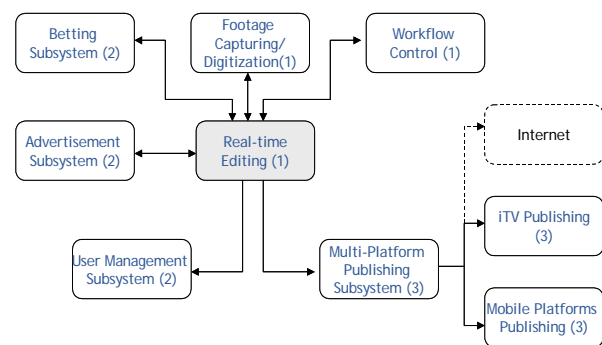


Figure 2 Overall Melisa Server Architecture

of the suitable visual enhancements that can be applied for a specific event.

The Transcoding Unit receives the visual enhancement information derived from pre-defined templates, and generates new MPEG-4 content. The MPEG-4 contains all the interactive elements and enhancements for supporting Melisa services; such as betting, advertising, etc. The Video Transmission unit is responsible for receiving both MPEG-4 and MPEG-2 video content and transmit them either multiplexed, MPEG-4 over MPEG-2 video, for satellite networks, or as RTP streams over IP networks.

The video streams are received either from the Melisa Set Top Box client, or from wireless clients such as PDAs. Both clients receive enhanced video content that allows the viewer to fully interact with the video content, to place on-line bets, view statistical information and perform e-commerce transactions. Each transaction is transmitted to the Melisa Server through IP connections for authorization and processing.

3.2. XML and Multimedia creation and representation

Information Merging is an important part of the MELISA Sender side since it integrates the offline and online processed content with the semantic annotation parts and prepares it for adaptation and delivery. Merged scene information is integrated with the video and other multimedia content to provide the final MPEG-4 content.

Successive annotation rounds provide scene description material, as the data traverses the different system components.

The process of dynamic generation of enhanced content can be significantly accelerated by the introduction of the notion of templates. Templates contain all generic aspects and can be instantiated for specific Event Actions and sports, to provide a uniform visual appearance and interaction capabilities. The template instantiation takes place after the completion of the authoring phase, namely the capturing, the visual enhancement of the content etc.

A template contains screen elements as distinct entities and is subject to a task order. Hence the screen elements are indirectly associated to the completion of specific tasks. A screen element can vary from simple logos to placeholders on the screen that can contain textual or visual information. A screen element is considered to cover a rectangle box regardless of its shape.

The output of the Information Merging unit essentially consists of real time of offline merged scene descriptions, encoded in textual BIFS format and metadata instances. The data describe the AV content that will be transcoded into MPEG-4 either for transmission as an RTP stream or over MPEG-2 for DVB-S transmission. The Information Merging Unit also generates MPEG-7 MDS-based metadata associated with the MPEG-4 content [4].

3.3. Lottery System/Advertising system

The Betting Subsystem handles the issues concerning offering of betting services through Melisa system. Under this scope, it implements the interface with an existing Lottery System in terms of data exchange and transaction handling.

The implemented interfaces as well as the data flows that occur during operation are presented in the relevant schema.

In order to support betting operations within Melisa system, it is necessary to interface a Lottery System. The Lottery System includes and provides all

necessary rules and mechanisms that are required for a Lottery Business to operate. This includes, among others definition on games, draw management, business configuration management, risk management for certain games, accounting, MIS applications, extensive reporting, etc.

The Betting Protocol Editor module holds an administrative role concerning the events that are provided for betting. As show in the above figure, it interacts with the Lottery System and Betting Activation Engine in order to support Melisa betting services. Provided functionality includes the following:

- The administrator uses the Event Definition in order to retrieve information from the Lottery

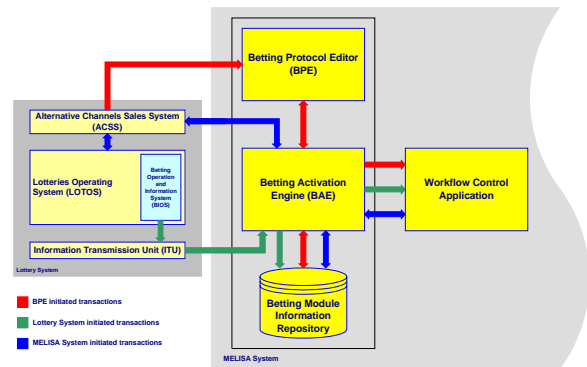


Figure 3 The Melisa Lottery Server

System concerning betting events and select which of them will be provided to the users. Information is communicated to Betting Activation Engine, stored on the local repository and also communicated to the Melisa Central Repository through the workflow control mechanism.

- Event Management is used in order to control the events published to the Melisa users. The administrator is able to enable/disable the events available for betting.

The Betting Activation Engine performs the critical role of interconnecting the Lottery System with the Melisa system. It has necessary set of rules to check and validate every message that is received and the necessary business logic to determine the actions to be performed. Provided functionality includes:

- Maintenance of a local repository to hold events related information and statuses that have been published to the Melisa core system. This information is updated according to the messages initiated from Betting Protocol Editor or Lottery System directly.
- Communication of bet placements to the Lottery System for bet validation and

registration. Respective responses are sent back the end user through workflow module.

- Maintenance of calculated events' odds, for usage by the Lottery's Traders. Every bet accepted by the Lottery System is provided as input to a real-time odds calculation algorithm. The purpose of such a mechanism is to propose odds for the events during bet placement in order to keep Lottery suggested payout to a certain range.

Communication between the modules as well as with the rest of the MELISA system is performed using XML. All messages are described using W3C XML Schemas (XSDs).

The concept of this communication is that the message originator creates XML content according to the specifications enforced by the XSD. Each message is validated prior transmission. After successful validation, the message is transferred using HTTP protocol and POST method. The receiver gets the message and performs validation according to the XSD. After that, the XML data are transformed to data structures in order to implement the requested function. In the same way and in synchronous mode, a reply is assembled, validated and returned to the originator from the same opened socket. The originator receives the reply, validates it, terminates communication and continues with processing of the data.

The advantage of this communication method is that data structure conformance and content integrity are performed during the validation of the message, eliminating the need to include such checks in the business logic of each component.

3.4. Real Time content Generation

In Melisa, we use MPEG-4 Rich Media, with audio-visual data of the sport event enriched with interactive BIFS graphics. The initial BIFS presentation is generated offline, prior to the start of the event, and encoded as an MP4 file. In the initial presentation, the broadcaster/program director embeds all audio-visual objects (images, sounds, etc.) and interactive functionalities (show clickable object, etc.) that will be used in the service. In order to maximize the user experience, the initial presentation look-and-feel is optimized for the target user terminal. Therefore, separate initial presentations are built for the Set Top Box, and PDA/mobile devices.

Figure 4 shows the initial presentation for a Football game scenario in the Set Top Box context. The interactive functionality that is common to all

scenarios is the **Bet Menu** depicted on the picture, which allows users to place a bet on a player during



Figure 4. Initial Football Game Scene

the event: players are listed with the associated odds; the user first selects a player, then the amount of money he wishes to play on the player; finally, a command containing all necessary information for the bet is sent to the betting engine.

When the event starts, the full initial scene together with the first images of the event is sent to the client platforms. During the event, online content is generated to enhance the initial scene with real-time data. The real-time interactive content is encoded as



Figure 5. Updated Football Game Scene

BIFS Updates for the initial MPEG-4 scene. This allows the program director to modify the parameters of the scene at any time, according to what happens in the event. A typical example of real-time content change is the modification of the Bet Menu. Typically

the odds for an available bet would vary during the event.

To simplify the task for the operator, a specific authoring tool is built for the Melisa service. This is an interface between the Information Merging Unit (IMU) and the MPEG-4 BIFS encoder (MP4Tool), which maps high-level interaction and graphics commands (such as *moveObject*, *replaceText*, *triggerAction*, *showObject*, etc.) to BIFS Updates. The operator is given a list of references on the scene Nodes that can be acted upon, i.e., for which Node BIFS updates can be generated.

Figure 5 shows the result of several generated real-time contents (i.e., BIFS Updates) on the initial presentation given in Figure 4. As we see on the top right corner of the Figure, the score has been changed, as well as the current game time. This corresponds to a call of method *replaceText* on the particular Text Nodes of the initial scene containing the score/time of the event. Also, following an important game action, a new image object has been displayed on top of player 9 which, when selected (by pressing 9 on the remote control) shows statistics for this player. Practically, this corresponds to successive calls of methods *triggerAction*, *moveObject*, *replaceText* and *showObject* on the appropriate Switch, Transform2D, Conditional and Text Nodes, respectively.

In addition to the BIFS Updates, we manage a BIFS Carousel, i.e., we periodically generate and send Random Access Points (RAPs), which consist in merging the initial presentation with the proceeding accumulated updates. The resulting new BIFS scene is useful for users that join the service after the start of the event, so that they provided with up-to-date information.

Note that although a different initial presentation was used for each target terminal (STB or PDA), our framework allows for real-time content scalability. Indeed, we take care of preparing both initial presentations so that the subsequent real-time updates are the same for both types of terminals. This is achieved by encoding both initial presentations with the same definition for the BIFS Nodes that are supposed to be updated during the event.

3.5. Enhanced Video Content Streaming

The transmission unit controls the transport of MPEG-4 over the DVB-S or UMTS/GPRS infrastructures, i.e. the transport of streamed MPEG-4 over MPEG-2, in the case of DVB-S broadcast, or its

transcoding and quality degradation so as to fit the lower transmission and playback capabilities of 3G mobile terminals.

Since the DVB and cellular-oriented architectures offer different degrees of media delivery, interactivity and responsiveness, it is essential to break down both the captured and synthesized material to match the relevant device. This task is appointed to the Control and Transmission Units: in this context, the Control Unit has to prepare different versions of the content and convey them to the Transmission Unit for delivery. After that, the Transmission Unit handles the connection with the satellite, in the case of a DVB broadcast, or the local UMTS/GPRS network, if the target device is a cellular one.

The transmission unit encapsulates the MPEG-4 content into the MPEG-2 transport stream (TS).

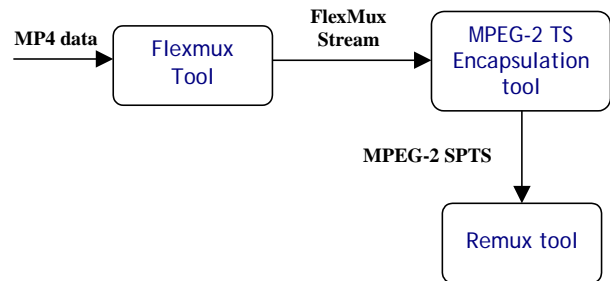


Figure 6. MPEG4 encapsulation into MPEG-2 Transport Streams

The ISO/MPEG took into account the transport of MPEG4 streams on a MPEG2 digital TV channel (**Amendment 7: Transport of ISO/IEC 14496 data over ISO/IEC 13818-1. January 2000**). The MPEG2 Transport stream syntax is relevant in a broadcasting environment. The PES packets may carry two types of payload, SL-Packetized streams, and FlexMux-Packetized streams. The latter is considered as a preferable solution and therefore adopted by the Melisa platform.

From the MPEG2 network point of view, the introduction of MPEG4 is seen as adding to the MPEG2 program a unique PID packet component. The adoption of the FlexMux scheme is greatly facilitating the evolution of a conventional MPEG2 Remux towards MPEG4 insertion since only one PID has to be reserved. It becomes more obvious if the bit rate of the FlexMux stream is regulated at a constant bit rate.

The FlexMux tool enables a stand-alone synchronization mechanism for the restitution of the MPEG4 audiovisual scene with the FlexMux Clock Reference (FCR). FCR is the common time base between elementary streams. FCR time stamp is the reference to be compared with the Composition Time

Stamps of the SL-packetized streams. As far as the synchronization of the elementary streams is concerned, FCRs are similar to the MPEG2 PCRs and CTS are similar to PTS.

An Encapsulation module converts the produced FlexMux data into an MPEG-2 SPTS conformant stream. The data conversion is performed by executing four processes in parallel:

- Import the Flexmux Data
- Create PES Packets from Flexmux Stream
- Create TS Packets from PES Packet.
- Export the TS stream with PAT and PMT.

The Remux tool let us integrate the MPEG-4 Flexmuxed data into a special Audio/Video MPEG-2 TS file with a stuffing component. The special Remux tool replaces the stuffing data in the MPEG-2 TS by the MPEG-4 Flexmuxed data. This module allows us to test the complete processing chain.

This tool is based on a generic Mpeg2Tool. This generic tool is a generic template that can be used to modify an Mpeg-2 TS stream. Here is the static view of the implementation of this special Remux tool in the Mpeg-2 architecture.

A streaming server (Figure 7) handles MPEG-4 streaming over an IP network. The streaming server is divided into three modules.

The main module is the native streamer called “Basic StreamingServer”. It is responsible for establishing the streaming session and manages the client reception. The streamer adapts the packet size (e.g., fragmentation), builds the headers (e.g., RTP headers) and provides the streaming session configuration parameters. The streamer does not adapt

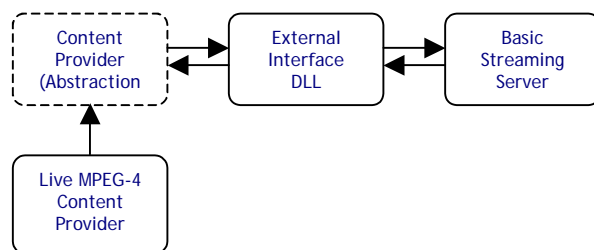


Figure 7. The Streaming Server

the conveyed streams (e.g., bit rate) according to QoS parameters.

The second module called “ExternalInterface” is a DLL that gives an interface of the content providers to the native streamer and can instantiate any content provider. It is responsible for the management of the Content Providers. According to a content URL or

URN, it contacts or creates the corresponding Content Providers.

Finally the “ContentProvider” module is a DLL that implements a specific content provider. A content provider is responsible for buffering the packets it retrieves and manages the packet management. It composes the contents and adapts them for delivery (e.g., transcoding schemes, bit rate modifications). The Content provider manages only one Scene containing Elementary Streams (ES). This schema offers the flexibility that different kinds of content providers can be defined and added to the streaming chain as new libraries.

4. Conclusions

The Melisa Platform offers a flexible system design for Multi-Platform Interactive Sports Content Broadcasting. Combination of computer vision techniques, mixed with MPEG-4 based interactive services, such as real-time Betting and Advertising on Set-Top-Box and Wireless clients, provides viewers of sports events a personalized, interactive experience.

5. Acknowledgments

MELISA is an IST project funded by the European commission in its 5th framework program. The project consortium consists of altogether 11 partners coming from industry and academia. This paper was possible only thanks to the excellent teamwork in the framework of this project. More details can be found under <http://melisa.intranet.gr>.

6. References

- [1] E. L. Andrade Neto, E. Khan, J. C. Woods and M. Ghanbari, “Player identification in interactive sport scenes using region space analysis prior information and number recognition”, *IEE Visual Information Engineering (VIE)*, University of Surrey, Guildford, UK, July 2003, pp57-61
- [2] E. L. Andrade Neto, E. Khan, J. C. Woods and M. Ghanbari, “Segmentation and tracking for interactive sport scenes using region adjacency graphs, picture trees and prior information”, *Picture Coding Symposium (PCS) 2003*, Saint Malo, France, April 2003, pp 353-358.
- [3] R. Koenen, “MPEG-4 - Multimedia for our time”, *IEEE Spectrum*, Vol. 36, No. 2, Feb. 1999, pp. 26-33.
- [4] J. M. Martinez, *Overview of the MPEG -7 Standard*, Dec. 2001, ISO/IEC JTC1/SC29/WG11 N4509.
- [5] F. Pereira, T. Ebrahimi, *The MPEG-4 Book*, IMSC Press, Prentice Hall, New Jersey, 2002
- [6] Aaron E. Walsh, Mikael Bourges-Sévenier, *MPEG-4 Jump-Start*, Prentice Hall, New Jersey, 2002