A UNIFIED SOLUTION FOR THE INTEGRATION OF MEDIA APPLICATIONS AND PRODUCTS IN BROADCASTER ENVIRONMENTS – THE ASSET ARCHITECTURE

PAULA VIANA[1, 2], **MARIO CORDEIRO**[1], **VITOR RODRIGUES**[1], [1]INESC Porto ([2]ISEP), Portugal
DAMIEN BOMMART, Compaq-HP, France
GIULIA FERRARI, MASSIMO STRAMBINI, SHS, Italy
INGO HOENTSCH, TOBIAS MARX, IRT, Germany
WALTER BERNET, EDGAR MÜLLER, Dalet a.n.n, Germany
BERNARD COUSIN, MATHURIN BODY, INRIA, France
SERGE DAULARD, THOMSON, France
BERNARD ALGAYRES, MARC LAURENTIN, FPDI, France

ABSTRACT

The ASSET project, an European funded project, is defining and developing a software architecture that will be made available to manufacturers for allowing easy interfacing between digital television equipment and applications. The project partners (Compaq-HP Group, THOMSON, Dalet a.n.n, INESC Porto, INRIA, Institut fuer Rundfunktechnik, Front Porch Digital International, SHS Multimedia) are creating an universal and unified system architecture that shall make the integration of different systems easier. This paper gives an overview of the project approach, describes the system architecture and the prototype under development.

INTRODUCTION

The introduction of IT concepts and technologies has been opening the possibility for new approaches on the implementation of digital television facilities covering the whole workflow: acquisition, creation, editing, control, storage, broadcasting, publishing and archiving of digital TV content.

This approach hasn't however yet solved the problems that Broadcasters and System Integrators face due to lack of connectivity and interoperability between equipment and applications: solutions available in the market are still vertically integrated or proprietary, requiring specific and costly development, relying typically on a single manufacturer or system integrator.

The ASSET project [1] (IST-2001-37379 Architectural Solutions for Services Enhancing digital Television) main goal is to overcome these problems by creating an universal and unified system architecture that will provide a set of software tools and APIs that shall make the integration of different systems independent from the device manufacturer, programming language and underlying middleware platform.

The project is exploiting opened standards and emerging technologies (like MXF [2], standard data models for describing essence, XML [3] and distributed systems technologies) for defining the concept of an Asset Middleware that will wrap the standard software layered architecture into a software middleware which proposes:

- The abstraction of broadcast software and hardware devices as logical resources
- Generic, openly defined and simple interfaces to control devices and data distribution
- Added value for system logic: decisions to configure the devices and to convert/move data

SYSTEM ARCHITECTURE

The ASSET architecture is based upon a software Framework - the *ASSET Framework* - composed by a set of standard interfaces and protocols for applications and products working together in an integrated environment.

Once applications are connected to the Framework, they can communicate with any other application.

Products from different manufacturers can be natively connected to the ASSET Framework or through a software adaptor – the ASSET agent or ASSET proxy – enabling their control and management by any ASSET compliant application connected to the Framework.

Adding a new application or a new product to the system simply requires the writing and integration of a small software adaptor.

Figure 1 illustrates the different components of the ASSET architecture.

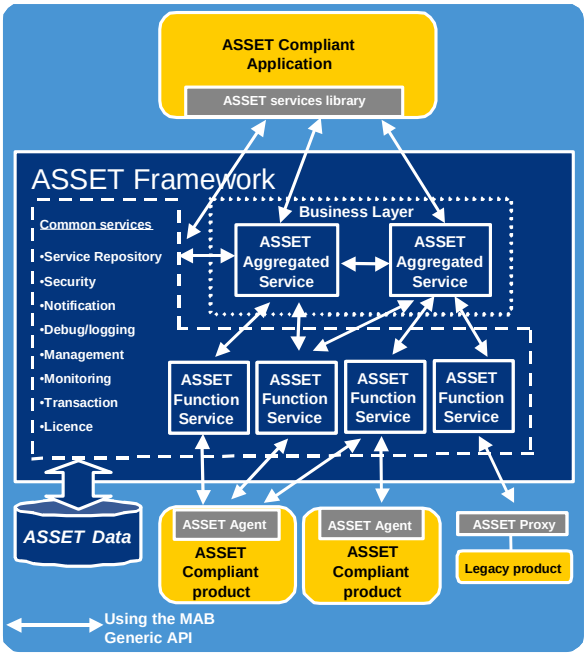

Figure 1 – ASSET Architecture

ASSET Architecture components

The ASSET Architecture defines a number of concepts and functions that enable the implementation of an ASSET Compliant Framework:

- The *ASSET Common Services* provide implementation of key infrastructure requirements such as security, logging, notification, resource management, etc. This allows a uniform and single implementation of these services throughout the solution.

- The *ASSET Function Services* provide an abstraction layer at the function level (encoder, recorder, player, etc.) hiding the specificities of products connected through agents and proxies (eg: at that level a VTR output or a Video Server output are just considered as 2 system-wide logical output ports).

- ASSET *Aggregated Service*s implement additional logic on top of lower layer services (provided by ASSET function services, ASSET common services or even by other aggregated services). An example of an ASSET aggregated service can be an *ingest service* that is built on top of a recorder function service, a transcoder function service and the ASSET notification common service (to notify applications connected to the Framework that a new asset has been acquired).

- An *ASSET compliant application* is a top level ASSET software component. It uses the services provided by common services and aggregated services registered to the Framework.

- The ASSET framework software adaptor enables the connection of an application or a product to the ASSET framework. It allows also sending and receiving messages to/from other software components connected to the ASSET framework.

- The *ASSET services library* is a software component that must be included into (or linked with) the Application software code in order to use the services provided by the ASSET framework (an ASSET framework software adaptor is one of the components of the ASSET services library).

- A *product* is a manageable hardware or software component implementing one or several common functions: most of the Video Server products implement a Recorder function (handling the Audio/Video input ports), a Player function (handling the Audio/Video output ports) and a Storage function (managing the storage repository where Media assets are recorded to or played from).

- An *ASSET compliant product* is a product that is managed by the framework through a built-in *ASSET agent* (which includes an ASSET framework software adaptor).

- A *legacy product* is a product that has no built-in ASSET agent. Nevertheless, this product can still be managed by the ASSET framework via an external software module called an *ASSET proxy* (which includes also an ASSET framework software adaptor).

THE MEDIA ASSET BUS CONCEPT

The ASSET architecture is implemented through a software bus called the Media ASSET Bus or MAB.

The goal of the MAB is to provide support for the widest range of applications and products integration within the ASSET framework, independently from the underlying protocols (RS-422, TCP, UDP, HTTP or SOAP) or the operating system environments.

The MAB is a set of standard interfaces and protocols, for applications and products, working together in an integrated environment. It defines:

- a message based **Transport Abstraction Layer (TAL)**, with **XML** format message exchange over the transport layers. The abstraction provides both synchronous as well as asynchronous call semantics.
- an interface that allows any third party media application or product to integrate into the Media ASSET Bus by developing a simple software adapter. Application developers or Agent or Proxy developers use the MAB Software Development Kit (SDK) in order to connect to the MAB and exchange messages with other MAB components.

The concept of MAB is illustrated in Figure 2 - different Media ASSET Bus components are connected via software adaptors built on top of the MAB SDK.

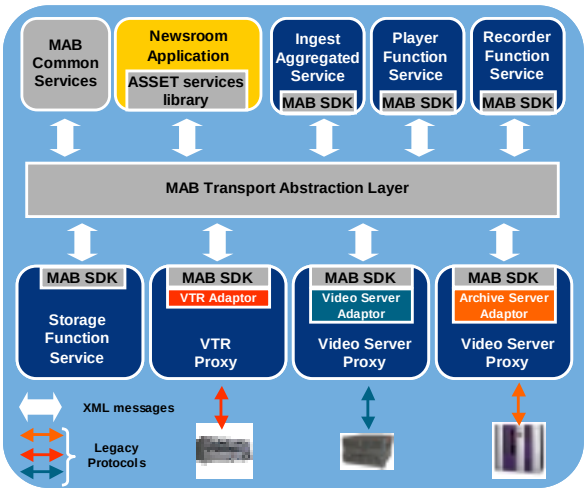

Figure 2 – The Media ASSET Bus

ASSET DATA MODEL

Within the ASSET framework, three data models, independent of each other and serving different purposes, were defined:

- A Structural and Control Data Model for the control and management of products and services available through the ASSET framework
- A Descriptive Data Model for the description of essence
- A Content Management Data Model for locating media assets

Only the first data model is maintained internally by the ASSET framework. The other two categories reside outside the MAB and are accessed through an abstraction layer on top of the corresponding metadata service. This approach guarantees that existing Media Asset Management systems can be integrated in the ASSET framework.

Structural and Control Data Model

The Structural and Control Data Model is the repository for the information related to all the services available through products in the ASSET framework. This data model is strictly tailored to the ASSET framework and represents the topology of the system and resources.

Descriptive Data Model

Almost every broadcaster or production company may have its own workflow and its own proprietary set of metadata describing essence, workflows and processes.

Hence, it is impossible to introduce a descriptive data model within ASSET that would meet all the different requirements. As a result, ASSET does not prescribe a fixed and universal data model, but provides facilities that allow users to implement their own descriptive data model. The idea of the descriptive data model within ASSET is not the storage of descriptive metadata, but the capability of exchanging and understanding descriptive metadata.

In order to offer an added value and to guide users implementations, ASSET provides a default data model. DMS-1 [7][8] is not compulsory and may be replaced by any other data model. However, only a single data model can be used within one ASSET system.

Content Management Data Model

Neither the Structural and Control Data Model nor the Descriptive Data Model provides a mean to:

- Define the structure of a Media asset (is it made of 1 physical file?, a collection of files?). For one existing Video Server, a Media Asset can be made of 1 file for storing the video essence, 4 files for storing the audio essence, 2 time-code files and 1 extra metadata file while for another video server a Media Asset will be stored in a single file (e.g. MPEG-2 Transport Stream file).
- Find the physical location of a Media Asset in the different storage repositories present in the system.
- Link derived Media Assets (same clip encoded at a different bit rate or using different compression techniques) from the origin Media asset.

The goal of the Content Management Data Model is to address this functional need.

CONTENT EXCHANGE

The requirement to share/exchange content files between broadcast facilities, using non-proprietary formats, has been emphasised by user organisations like EBU. Aspects like, multiple users to simultaneously and independently access the same content; various and adaptable speed transfer across Local and/or Wide Area Network; common "container" for data, metadata and essences organised in data models; simple and direct access to the content through standardised network protocols and interfaces and unified formats for manipulating, managing, sharing, storing and distributing essences and metadata are priorities that must be satisfied by emerging content exchange technologies.

The Pro-MPEG Forum [4] and the Advanced Authoring Format (AAF) Association [5] developed an open standard that ensures the interoperability among the different vendor systems involved in production environments.

MXF - Material eXchange Format

MXF [6][2] has been designed to meet user demands. It is being put forward as an *Open Standard* that is *not compression-scheme-specific*, simplifying the integration of systems using MPEG and DV as well as future, yet unspecified, compression strategies. Transportation of these different files is then independent of content, and does *not dictate the use of specific manufacturers' equipment*. Any required processing can simply be achieved by automatically invoking the appropriate hardware or software codec.

MXF in ASSET Framework

According to the ASSET Framework, content exchange (audio-visual material and associated metadata) shall rely on a file format that provides full interoperability between different equipment and different applications. The exchange format should be open and standardised, compression-independent, cross-platform and support streaming/transfer bridging.

Due to its characteristics, MXF is being used as the ASSET solution for Content Exchange. Two different scenarios were defined:

- the native interchange file within the same ASSET framework
- the import/export of both Essence and Metadata between different ASSET architecture based systems or simply external ASSET MXF compliant products.

The difference between these two approaches does not reside in the file format being used (both cases use MXF) but in the content of the MXF file:

- for the **Exchange between ASSET compliant products connected to the same ASSET framework**, only the essence and the structural metadata will be included in the file as the correspondent descriptive metadata is stored inside the ASSET metadata function services, shared by both products.
- For **Importing/exporting to/from outside ASSET**, descriptive metadata can be included in the MXF file. The import process must extract the descriptive metadata and update the ASSET metadata function services. The export process must build MXF files with both structural and descriptive metadata (available from the ASSET metadata common services) allowing external applications and systems to access essence and metadata (structural and descriptive) in just one file transfer.

DEMONSTRATION PLATFORM

Figure 3 shows the currently planned demonstrator for IBC 2003. This demonstrator will be a deployment example of the ASSET framework in the common situation of TV production, where devices and applications of different manufacturers are integrated into one system. The demonstrator does not focus on the particular workflow that has been chosen, but on how the functionalities of the system are supported by the ASSET middleware and how integration of the system is simplified by the ASSET technology.

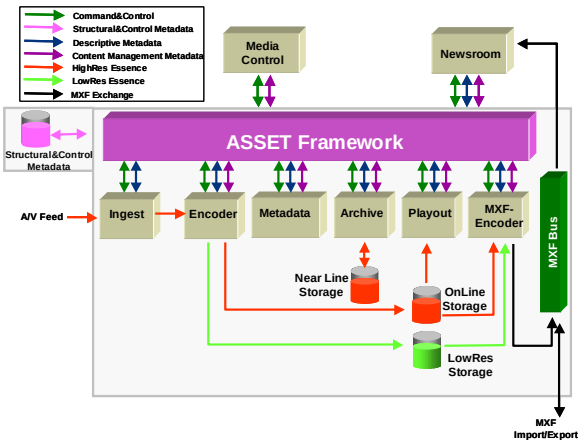

Figure 3 – ASSET demonstration scenario

The demonstrator uses two clients and several providers of different manufacturers to emulate a simple workflow of a News Platform. The clients are a media control application and a news system. The media control application is used to control the ingest

processes, to delete content, to copy content or to move content between devices. The newsroom application provides the creation of a programme (rundown) referencing essence located on the online storage, offers a preview of the material and the metadata, and triggers the playout of the videos.

The providers are an ingest server, a transcoding device, an archive system (near-line storage), a online storage system, a playout server, a MXF encoder and a database containing metadata describing all content stored in the system.

In the demonstration system, content can be previewed and played out via the newsroom application. The media control application can initiate content transfers between the archive server, the online storage or the playout server. Content can be ingested into the system via A/V- feed to the ingest server or via MXF file transfer. In both cases, descriptive metadata is created by the system or, if it already exists, extracted from the ingested content. A low-resolution browse stream is generated to support the pre-view functionality of the newsroom system.

Using the ASSET framework within the above scenario allows both easy integration and collaboration of various proprietary applications and devices within a broadcast environment and transparent exchange of essence through the aid of MXF.

The ASSET demonstrator can also serve as a reference system where end-users, administrators and integrators are able to test and to verify the implemented functionality against the requirements.

CONCLUSIONS

This paper presents the work under development in the IST ASSET Project. The project defined an architecture and APIs that will be made available to manufacturers, so that interconnection between equipment in a digital television environment is easier. The architecture, concepts and demonstration scenario, have already been defined by the partners. The project is currently working towards the development of a newsroom platform that will demonstrate the ASSET concepts.

REFERENCES

[1] ASSET web site – http://www.ist-asset.com

[2] Pro-MPEG Forum, Material eXchange Format (MXF) 10b, http://www.pro-mpeg.org/mxf.htm, October 16, 2002

[3] Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation. World Wide Web Consortium, http://www.w3c.org/TR/REC-xml, October 6, 2000

[4] Pro-MPEG Forum - http://www.pro-mpeg.org

[5] AAF Association – http://www.aafassociation.org

[6] Bruce Devlin, MXF – the Material exchange Format, EBU Technical Review, July 2002

[7] Pro-MPEG Forum, A guide to MXF Descriptive Metadata Schemes and their application, August 2002

[8] Pro-MPEG Forum, Material Exchange Format (MXF) - Descriptive Metadata Scheme (DMS-1), October 2002