Audio Engineering Society

Convention Paper 6730

Presented at the 120th Convention
2006 May 20–23 Paris, France

Applying EAI technologies to bi-medial broadcast environments. Challenges, chances and risks.

Michael Zimmermann[1],

[1] Solution Manager Archives at VCS AG, Borgmannstr. 2, 44894 Bochum, Germany
michael.zimmermann@vcs.de

ABSTRACT

More and more broadcast companies try to optimize their production environments by enforcing bi-medial workflows. The recent applications and tools on the other hand only have poor integration interfaces to achieve this goal. EAI, originally focussing on the integration of legacy systems, has become a mature toolset to integrate various systems and offering tools and applications to ease integration. This lecture should show the possibilities and limits of EAI in bi-medial broadcast environments.

1. INTRODUCTION

Starting from the recent situation in many broadcast companies, we figure out how the workflows in radio and television production environments are, and continue to work out where bi-medial work takes place and makes sense. In each environment there are

- Different applications dealing with
- Different file formats and
- Different metadata models

Additionally there are different workflows between different broadcast companies, radio or television and program type. Considering all these manifolds it's easy to see that building an integrated, bi-medial production environment is a demanding task, which needs most modern technologies to achieve this goal. The core functional blocks of building a bi-medial production environment are:

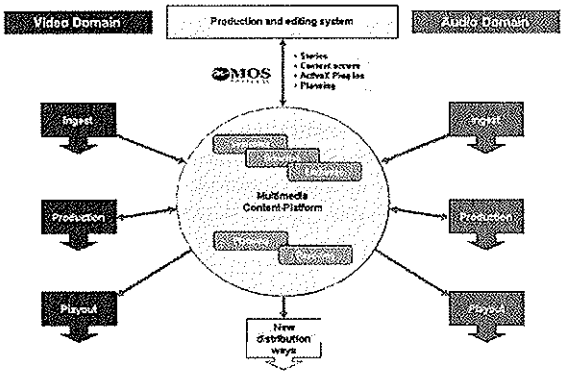
- Application integration
- Media management
- Metadata management
- Workflow management

- Enterprise planning
- Monitoring & Control for the overall architecture

2. THE BI-MEDIAL BROADCAST ENVIRONMENT

First of all, there is of course the question, whether or not we need bi-medial broadcast environments. If we reflect the former development of both environments independently, they end up in highly technical, optimized organisations which work well independent of each other, so why do we think about bi-medial work.

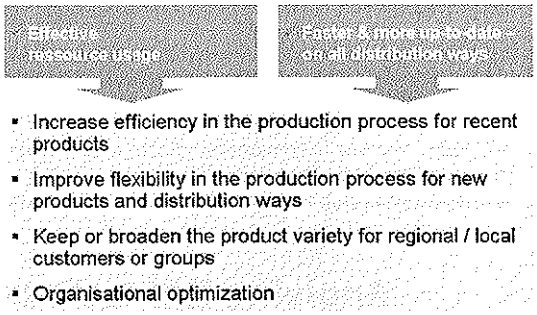
If we take a closer look , we see some similarities. In each domain there is an ingest, a production and a playout platform, which naturally results in completely different systems for each functional block.



Each user is used to his set of tools for his daily work, and has no information of what is available in all the other domains. And, if so, how to get the material from the other application to the own application.

If we don't think about one application for all, we have to accept that there will be a variety of specialised applications for the entire broadcast environment which work more or less independently from each other. If we want to take advantage of work that has already been done or plan and schedule work packages and productions independent from the underlying media (e.g. year of Mozarts birth), we need tools to support that. The first diagram shows the simplified view on the audio and video domain and their common subset of functionality.

Why Bi- and Tri-Mediality?



To come back to the initial question, we do not need bi-medial broadcast environments, but they have a lot of advantages and allow workflows which are impossible in separated environments.

3. PROBLEMS TO SOLVE

The centralized multimedia content platform is the market place for the essence of all applications in the domain. The functionality of this centralised domain comprises of the following elements:

- Enterprise Searching (see lecture 306 , Advanced cataloging and search techniques in audio archiving)
- Enterprise browsing (see lecture 306 , Advanced cataloging and search techniques in audio archiving)
- Exchange of essence and metadata
- Enterprise planning: When talking about bi-medial work there needs to be a sophisticated overall planning. The tools must be able to plan independently or with dependencies between the different production domains. It must provide long term to short term planning and it has to manage due dates for different stakeholders in the production process.
- Enterprise reporting: As bi-medial, integrated production workflows tend to get more

complicated it is necessary to have tools for analytical processing and report generation.

Even if all tools for bi-medial work have a different purpose, a decomposition of these applications will result in a set of common functionality, which is required for all the applications and normally not available in the existing applications from the audio or video domain.

Application integration: Naturally most applications are designed to have poor interfaces for integration. But this is exactly the necessity in this context, to have or make applications open for integration. This is one of the biggest challenges in an integrated environment.

Media management: Managing different file formats and providing the required format for the applications is one task that needs to be performed in an integrated environment. Consolidated storage and the separation or merging of audio and video are other functional aspects in bi-medial work. The media management must be usable independent from a concrete application, like a service that is provided where everyone who needs media management can access it.

Metadata management: Each application, database or metadata model normally has its own set of metadata, its own ID's and it's own structure and formats. The exchange of metadata between the applications is one issue of metadata management. The unique interpretation of the semantics is another aspect that needs to be considered when dealing with different metadata models. For example, if the production workflow requires a decision based on metadata content, this must be evaluated in a certain metadata schema. The semantics make it possible to interpret the contents properly.

Workflow management: Radio and television have their own workflows, but there possibilities to merge parts of these workflows or to integrate both domains on a workflow basis. The workflow management reflects the daily radio and television production logic and orchestrates the low-level functionalities to achieve this goal. Workflow engines allow the flexible adaption of customer-specific workflows on the basis of integration of services.

Monitoring & Control: Every application by itself normally has its complexity, integrating a variety of different applications forces that there needs to be a monitoring and control system to keep track of the procedures.

All mentioned functionality is largely compatible with recent Enterprise Application Integration (EAI) technologies, which builds the baseline for interconnected, coordinated enterprise working.

The next chapter shown the available technologies and their applicability to broadcast environments.

4. APPYLING EAI TO BROADCAST ENVIRONMENTS

4.1. Application integration

Application integration or Enterprise application integration[1] is defined as the use of software and computer systems architectural principles to bring together (integrate) a set of enterprise computer applications. It has continually gained wide recognition beginning around 2004.

Enterprise Application Integration has increased in importance because, traditionally, enterprise computing often takes the form of islands of automation. This occurs when the value of individual systems are not maximized due to partial or full isolation. If integration is applied without following a structured EAI approach, many point-to-point connections grow up across an organization. However, EAI is not just about sharing data between applications; it focuses on sharing both business data and business process.

Currently, it is thought that the best approach to EAI is to use an Enterprise service bus (ESB), which connects numerous, independent systems together. Although other approaches have been explored, including connecting at the database or user-interface level, generally, the ESB approach has been adopted as the strategic winner. Individual applications can publish messages to the bus and subscribe to receive certain messages from the bus. Each application only requires one connection, which is to the bus. The message bus approach can be extremely scalable and highly evolvable.

Enterprise Application Integration is related to middleware technologies such as message-oriented middleware (MOM), and data representation

technologies such as XML. Newer EAI technologies involve using web services as part of service-oriented architecture as a means of integration. Enterprise Application Integration tends to be data centric. In the near future, it will come to include content integration and business processes.

The common way to integrate with applications which are not directly able to support technologies like XML, Messaging or Web Services is to create connectors to the proprietary API of the application. We can distinguish four levels of enterprise integration readiness:

1. No interface: No chance.

2. Extract-Transfer-Load: Metadata or media files will be extracted from the application, transferred to the other system and loaded into the system. This is a very low-level integration with loose coupling and interaction between the systems. Remote control of these applications is barely possible.

3. Proprietary API: The application provides an API which allows the access to the system and partial or full control of the system behaviour. The way to integrate is to create an EAI enabled connector as a wrapper around the native API. This allows access to the peripheral application based on open standards. Possible standards are SOAP/Web Services or JCA[2], the Java Connector Architecture.

4. Direct support of Messaging, RMI or Web Services. This is the most elaborated way to integrate. The application can be integrated directly just by accessing their published interfaces.

The next step of integrating after accessing the peripheral applications is the logical interaction, or "broadcast production logic", which provides all required functionality for the multimedia content platform.

The appropriate framework to realise this functionality is base on middleware concepts.

Middleware[3] is now used to describe web servers, application servers, content management systems, and similar tools that support the application development and delivery process. Middleware is especially integral to modern information based on XML, SOAP, Web services, and service-oriented architecture.

Middleware is the enabling technology of Enterprise application integration.

There are several approaches available in the market like Microsoft Biztalk Server[4], Bea WebLogic[6], JBoss[5] and others.

They all have in common that the required functionality is decomposed into components and published via standardised interfaces like SOAP, DCOM, RMI or CORBA. Applications are then able to use this functionality by integrating the client stubs of the published functionality. The following chapters describe the components or functional blocks that are common and required in broadcast and bi-medial environments.

4.2. Media management

The most obvious items to manage are the digital media files. The main functionality comprises of

- Import and export of the media between the applications
- Generate browsing copies for federated search capabilities
- Storage of the media
- Transcoding, normalization
- Cutting, joining
- Watermarking, digital rights management (DRM)
- Audio and video analysis (quality checks, speaker (change) detection, key frame generation)

All functionality should be solely addressable to achieve a high degree of flexibility and adaptability.

4.3. Metadata management

Metadata (Greek: meta- + Latin: data "information"), literally "data about data", are information about another set of data. A common example is a library catalogue card, which contains data about the contents and location of a book: They are data about the data in the book referred to by the card. Other common contents of metadata include the source or author of the described dataset, how it should be accessed, and its limitations. Another important type of data about data is the link or relationship between data. Some metadata schemes attempt to embrace this concept, such as the Dublin Core element link.

Since metadata are also data, it is possible to have metadata of metadata–"meta-metadata." Machine-generated meta-metadata, such as the reversed index created by a free-text search engine, is generally not considered metadata, though.

→ See tutorial on metadata here at the AES.

In the broadcast context metadata is the data accompanying the essence. Of course there are standards to unify the metadata model in the broadcast environment[7][8][9][10], but the reality looks different and follows the statement: **"The good thing about standards is that everyone has its own.".**

Especially in an environment where various applications are integrated and data needs to be exchanged, we can expect the data to look different in each domain. If we are required to exchange metadata between systems there are some general considerations independent from the concrete underlying model that need to be considered.

1. Availability of data elements. Which fields are available in each system?
2. Format of fields. This is a wide area of possibilities and problems:
    a. Length of text fields
    b. Date formats (22.08.2006, 2006-08-22, 08/22/2006)
    c. Duration (150, 2'30'')
    d. Keyword mappings (Production type, Stereo/Mono/Surround)
    e. Structural differences. E.g. a name of a person can be Wolfgang Amadeus Mozart; Mozart, Wolfgang Amadeus; W.A.Mozart or divided up into first name, (middle name), last name.
3. Exchange format. This has also different possibilities like comma separated files, KLV files, XML. Luckily most vendors agree upon XML as the exchange format of choice.

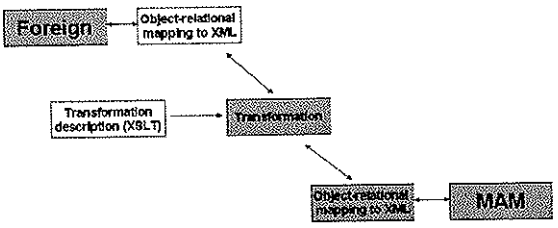
All mentioned problem areas are on the syntactical level. But there are as well problems on the semantic analysis of metadata, which addresses the interpretation of the metadata. For example let's take the field "creator", this can be the creator of the original recording or the creator of the digital media or the creator of the metadata record. If we now want to take metadata from one application and transfer it to another application, which has also a creator, this doesn't necessarily guarantee that it has the same meaning, it requires a deeper knowledge on the meaning of the field. Another usage scenario is a federated search over heterogeneous databases, where the search phrase needs schema dependant interpretation. The framework for describing these relationships is based on ontologies[11].

As XML gets the de-facto standard for representation of metadata, we can neglect to discuss various formats, but what's left open is the management of the metadata.

EAI technologies support the XML-based metadata management. Therefore there are a lot of standards based on XML. For the description of a metadata model, which means the structure and the contents, there is the XML Schema[12] definition. Metadata that will be exchanged can be validated against those schema definitions to programmatically proof its consistency.

To exchange metadata between different schemas we need transformation of metadata, defined as XSL Transformations.[13] This allows application independent transformation based on Schemas and Transformations. This is no longer something strongly related to broadcast environment or tools, it's simply an

abstraction of the real world to make it more universally usable.



The diagram above shows the relationship and usage of the XML based standards. The MAM and the foreign system have their own database with their own metadata model. The object-relational mapping maps the data model to an XML Schema. Data that will be transferred during the transformation process can be validated against the schema. The transformation process transforms the data from schema to schema based on the rules defined in the XSLT definition. We will pick this subject up again when talking about the Enterprise Service Bus approach.

This technology is used for searching and browsing as well.

4.4. Summary

The described components for the integration of audio and video domains to build a bi-medial production environment covering the following aspects:

1. Integration components for peripheral applications
2. Media management components
3. Metadata management components
4. Middleware serving as a framework for the upper components

This is the basic infrastructure for building applications. All enterprise tools like searching, browsing, exchange, planning or reporting can use the components and orchestrate them in the required manner. But how can we achieve automated processing without human intervention, just because the required processes are always the same. This requires an additional component in the integration environment, the so-called workflow engine.

5. WORKFLOW MANAGEMENT

Workflow management or business process management tools refer to activities performed by businesses to optimize and adapt their processes. [14]

Although it can be said that organizations have always been using BPM, a new impetus based on the advent of software tools (business process management systems or BPMS) which allow for the direct execution of the business processes without a costly and time intensive development of the required software. In addition, these tools can also monitor the execution of the business processes, providing managers of an organization with the means to analyze their performance and make changes to the original processes in real-time. Using a BPMS the modified process can then be merged into the current business process atmosphere.

The recent implementations are designed to flexibly change processes as required without changing program code. The business process management contains three activities:

5.1. Process design

This encompasses either **the design or capture of existing processes.** In addition the processes may be simulated in order to test them. The software support for these activities consists of graphical editors to document the processes and repositories to store the process models.

5.2. Process execution

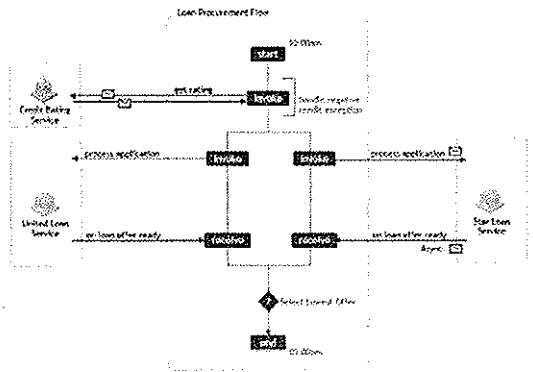
The traditional way to achieve the automatic execution of processes is that an application is developed or purchased which executes the steps required. However, in practice, these applications only execute a portion of the overall process. Execution of a complete business process can also achieved by using a patchwork of interfacing software with human intervention needed where applications are not able to automatically interface. In addition, certain process steps can only be accomplished with human intervention. Due to the complexity that this approach engenders, changing a process is costly and an overview of the processes and their state is difficult to obtain.

As a response to these problems, the Business Process Management System (BPMS) category of software has evolved. BPMS allows the full business process (as developed in the process design activity) to be defined in a computer language which can be directly executed by the computer. The BPMS will either use services in connected applications to perform business operations or will send messages to human workers requesting they perform certain tasks which necessitate a human attribute such as intuition as opposed to automated processes. As the process definition is directly executable, changes in the process can be (in comparison to the traditional approach of application development or maintenance) relatively quickly moved into operation. In order to work effectively a BPMS often requires that the underlying software is constructed according to the principles of a **service-oriented architecture.** Thus, it is often difficult to make a suite of existing legacy systems fit with a BPMS.

5.3. Process monitoring

This monitoring encompasses the tracking of individual process so that information on their state can be easily seen and the provision of statistics on the performance of one or more processes.

The Oracle BPEL engine[15] is such a tool that provides business process management functionality.



The diagram above shows an example of a business workflow which invokes services to fulfil the defined workflow.

If we think about bi-medial workflows we will find, from the perspective of the BPM engine, the following components:

- Services for integration with the applications from the audio and video domains
- Services for media and metadata management

This can be seen as a service-oriented architecture, which is the required foundation for business process management.

5.4. Service-oriented architecture

Unlike traditional point-to-point architectures, SOAs comprise loosely coupled, highly interoperable application services. These services interoperate based on a formal definition independent of the underlying platform and programming language (e.g., WSDL). The interface definition encapsulates (hides) the vendor and language-specific implementation. A SOA is independent of development technology (such as Java and .NET). The software components become very reusable because the interface is defined in a standards-compliant manner. So, for example, a C# (C Sharp) service could be used by a Java application.

SOA provides a methodology and framework for documenting enterprise capabilities and can support integration and consolidation activities.

High-level languages such as BPEL or WS-Coordination take the service concept one step further by providing a method of defining and supporting workflows and business processes.

5.5. Summary

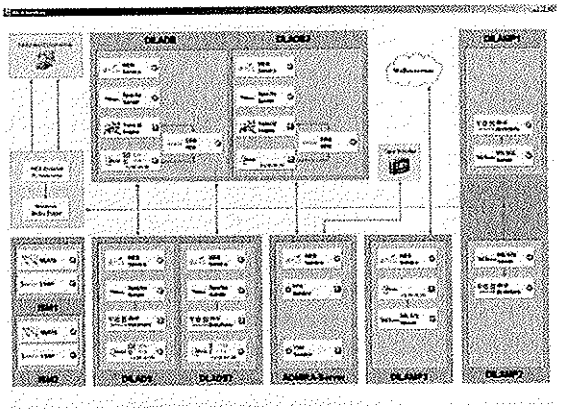
On top of the functions and services we have the tools to build applications that operate enterprise-wide, we are able to define processes that run automatically and that can be defined in an independent notation without the change of program code.

For the attempt to apply EAI to broadcast, bi-medial environments it shows that the infrastructure is already there as long as the existing, broadcast-related applications allow the integration in a broader context.

With the coupling, either loose or tight, between the applications, the overall system gets more and more complex and the dependencies between the applications can cause problems which are not transparent to the users or administrators.

6. MONTORING & CONTROL

Due to the growing complexity of the integrated system monitoring & control is necessary to keep track of the recent state of the system.



The diagram above shows a M&C screen with the systems and applications and the current state of the system. If there are complex workflows involving several applications there need to be different views on the system. Where the user is interested in which state his jobs recently are, there the administrator is more interested in the state of all the servers. Both can e provided with the M&C tools. SNMP and JMX are recent standards for monitoring & control functionality.

7. CONCLUSION

Coming back to the original idea of integrating the audio and video production domains by utilizing EAI standards, we have identified the required elements for an integrated environment,

- Application integration
- Metadata management
- Media management
- Workflow / business process management

These elements must fulfil certain requirements to make them usable for enterprise browsing, searching, planning, reporting and the exchange of metadata and media. The related technologies from the EAI design approaches are

- Middleware / application server
- Web Services / SOAP
- XML with XSD and XSLT
- Service-oriented architecture (SOA)
- Business Process management (BPEL)
- Enterprise Service Bus (ESB)
- Monitoring & Control

The paper shows that there are great opportunities one the one hand, challenging new problems to solve on the other hand. Each aspect is worth making a separate paper of it and because of this the paper can simply give a first idea on possibilities and approaches to get bi-medial work to run. The biggest obstacle is mostly not the technical challenge but the integration with applications which are not open for such integration scenarios. You can build sophisticated integration components which are worthless in the moment you are not able to deliver the value to the user.

8. REFERENCES

[1] http://en.wikipedia.org/wiki/Enterprise_application_integration

[2] http://java.sun.com/j2ee/connector

[3] http://en.wikipedia.org/wiki/Middleware

[4] http://www.microsoft.com/biztalk/2006/prodinfo/what-is-biztalk-server.mspx

[5] http://labs.jboss.com/portal/jbossas/index.html

[6] http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/weblogic/

[7] http://dublincore.org/

[8] http://www.bbc.co.uk/guidelines/smef/

[9] http://www.irt.de/IRT/bmf/metadaten_ERD-17.pdf

[10] http://www.ebu.ch/en/technical/metadata/specifications/index.php

[11] http://en.wikipedia.org/wiki/Ontology_%28computer_science%29

[12] http://www.w3.org/XML/Schema

[13] http://www.w3.org/TR/xslt

[14] http://en.wikipedia.org/wiki/Business_Process_Management

[15] http://www.oracle.com/appserver/bpel_home.html

[16] http://en.wikipedia.org/wiki/Service-oriented_architecture

[17]