ENGINEERING AUDIOVISUAL MEDIA PROCESSING SYSTEMS

Werner Haas, Werner Bailer, Gert Kienast

Institute of Information Systems & Information Management, JOANNEUM RESEARCH, Graz; Austria

{werner.haas, werner.bailer, gert.kienast}@joanneum.at

Abstract. *Efficient content production, processing, archiving and management can only be achieved by the usage of reliable, fast and cost-effective IT systems. These systems must be based on sound and proven principles, combining up-to-date software technologies with excellent understanding of the application area, standards orientation, and most prominently, with a user driven approach.*

This paper is not only concerned with media repositories, but will give a broader view on the general architecture of audiovisual media processing systems, starting with some examples from the applications and experience of the institute.

An overview is given on what are the building blocks of an "audiovisual media processing system", namely the processing components, the description infrastructure and the storage components. A systematic view is given how the interfaces linking these components need to behave in the fundamentally different cases of media analysis and media monitoring, as compared to media manipulation and restoration. The two approaches to processing, namely a temporally linear pipeline approach usually utilized for media analysis, is compared to an approach allowing random access within a time-window, almost always necessary for applications doing manipulation/editing/-restoration on very often even more than one stream of content.

Also the requirements for the description infrastructure, concerning the metadata model as well as the access tools are discussed. The resulting possibilities for essence and metadata storage and the consequences, advantages and drawbacks of centralized versus distributed models will be highlighted.

Finally, a general architecture for a content analysis / media monitoring system will be presented. The instantiation of this generic architecture for two concrete cases will be shown.

1. Introduction

Engineering audiovisual media processing system is a multi-faceted problem, requiring prioritization and optimization of many parameters. A few application areas are presented in order to illustrate the various practical requirements from a user's point of view.

In Figure 1, the user interface of a media monitoring application (Brand Detector) is shown. This application is geared to detect the occurrence of sponsor logos in video, e.g. for sports events. Accordingly, main emphasis is on close to real-time logo detection, recognition and tracking as well as on sophisticated statistics of results. Media search, content management and distributed access are of minor importance.

In Figure 2, an application for interactive TV production is shown. Here, main emphasis is laid on content management and particularly on highly powerful, combined text and visual search tools.

In Figure 3 and Figure 4, the user interfaces for an interactive TV authoring tool and for a film restoration system are shown. In these cases, displaying single key frames as well as giving an overview of the total film is of major importance. Fast access and display of single frames, shots and scenes are of paramount importance for the user acceptance of such systems.

Figure 1: Media monitoring application (Brand Detector).

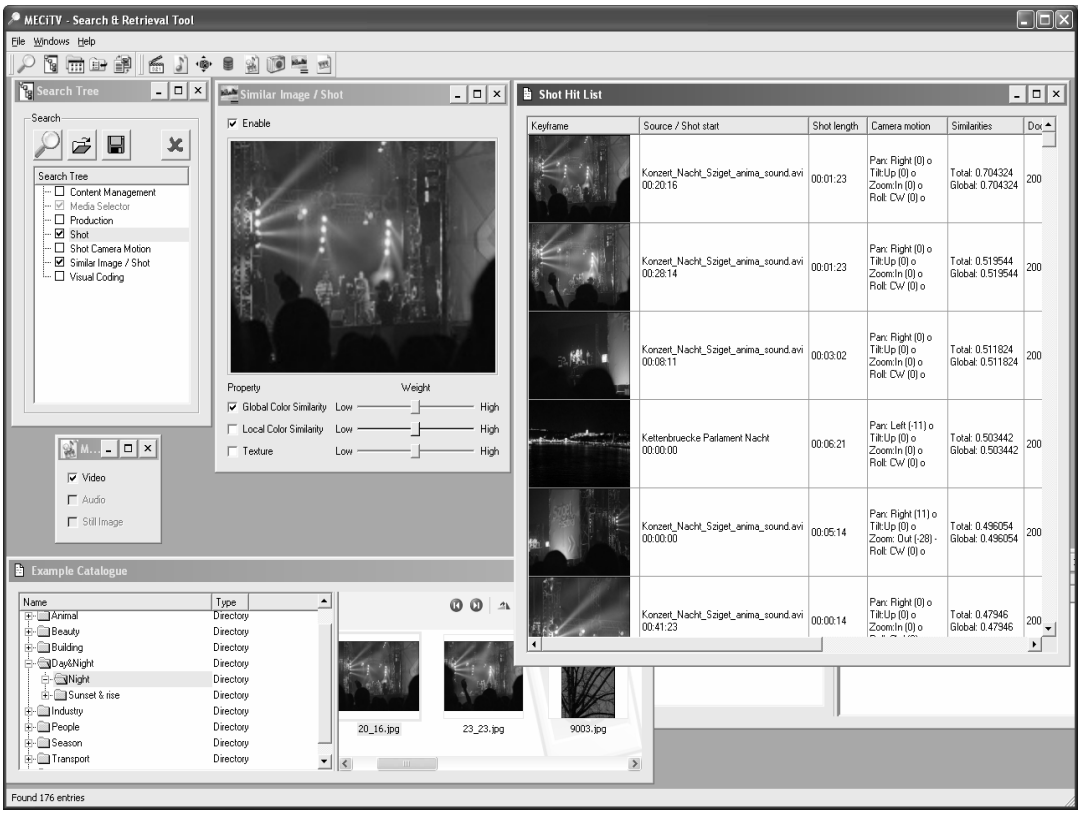

Figure 2: Search interface of an application in an interactive TV production environment.

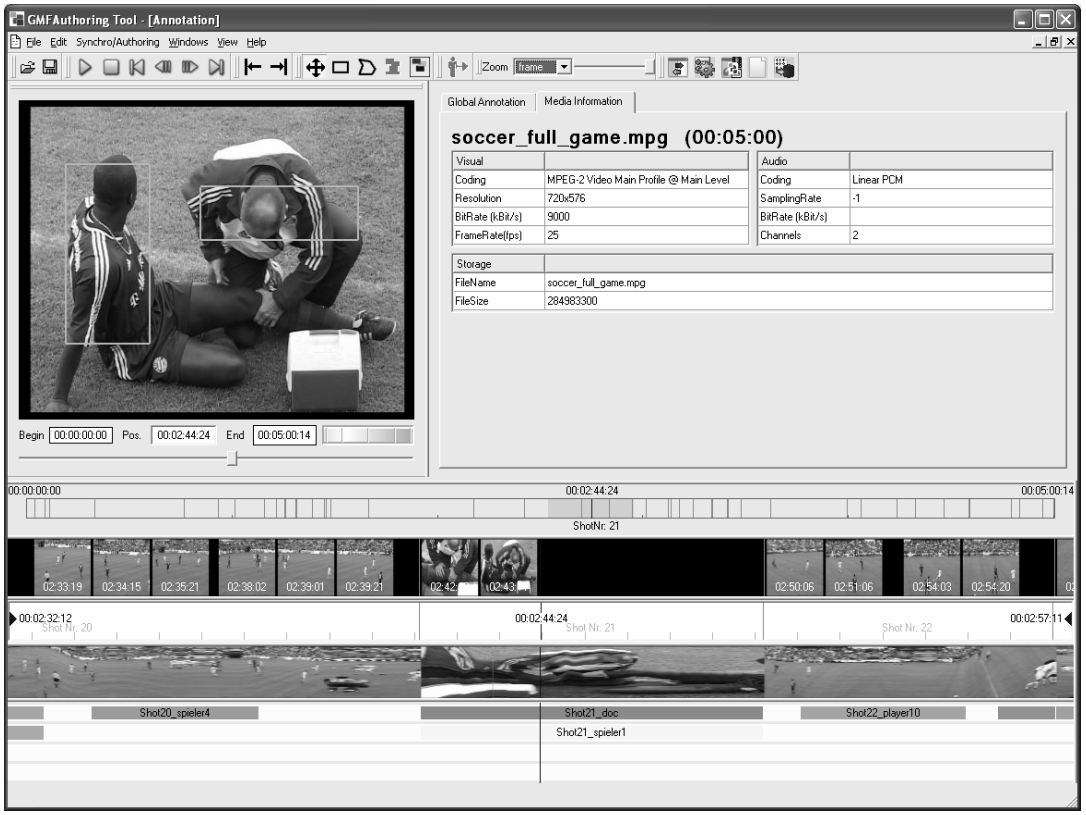

Figure 3: User interface of an authoring application for interactive TV productions.

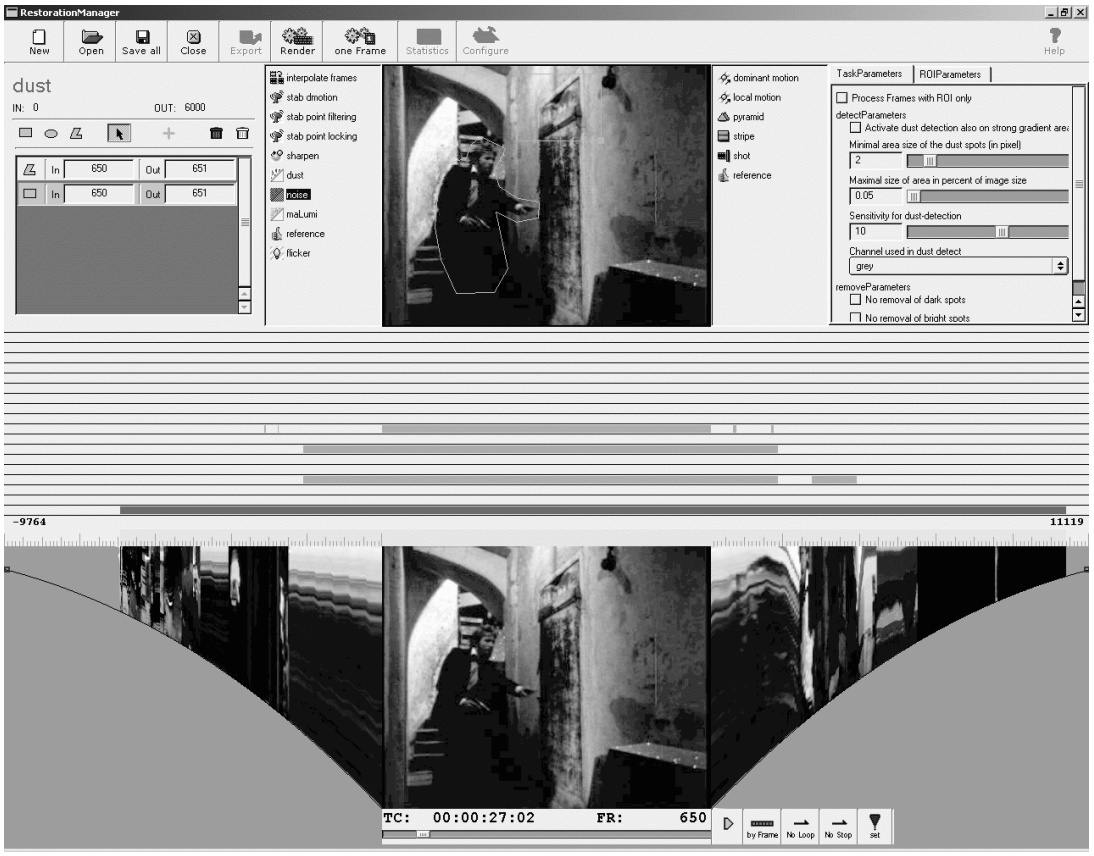

Figure 4: DIAMANT digital film restoration system.

2. Elements of audiovisual media processing systems

This section describes the elements we have identified to be shared by different types of media processing systems. The types of elements discussed below represent three main functionalities in a media processing system: processing, description and storage.

2.1. Processing components

Processing components are all components that work directly with the media data. All kinds of media processing systems need to perform analysis task on the media data, for example, to extract metadata or to determine the manipulation steps to be performed. Manipulation components are only required when the media data is modified, as it is for example the case in digital film restoration.

2.1.1. Analysis components

Analysis components take the media data (the essence) as input and produce other data. The result can be derived essence (e.g. transcoded essence or key frames), technical metadata (e.g. resolution, sampling rate) or a wide range of

descriptive metadata, such shot structure, camera motion, detected company logos, people appearing in the scene or genre classification.

In applications where a stream of essence is sequentially analysed, such as in media monitoring applications, a pipeline approach is a suitable concept for content-analysis. The samples of the essence streams are sequentially processed by a number of analysis modules, each performing specific tasks. The modules may be interdependent and require results of previous modules, which can be well modelled in the pipeline model.

If analysis is performed as a prerequisite for a manipulation of the essence, the pipeline approach is not always sufficient, as random access within a time window is required. This is for example the case when stabilizing an image sequence in film restoration, where it is necessary to estimate the instability over a longer time range, before a frame can be corrected.

2.1.2. Manipulation components

Manipulation components modify the media data and thus also need write access to the essence. As discussed above for analysis components, many manipulation components need random access to the essence within a certain time window.

Manipulation of the essence results may invalidate previously extracted metadata, which has to be taken account in the system design. If the manipulation components perform editing operations, they may also change the temporal structure of the essence.

2.2. Description infrastructure

The description infrastructure is based on the model which is used for representing the descriptions of media items, the metadata model. Furthermore, access components are required to enable the different components of the media processing system to work with the metadata model.

2.2.1. Metadata model

The metadata model is the core element for the more metadata-centric types of media processing systems, such as for example media monitoring or retrieval systems. In the following, we describe the main properties of the metadata model of such a system.

Comprehensiveness. The metadata model must be capable of modelling a broad range of multimedia descriptions (e.g. descriptions of different kinds of modalities and descriptions produced with different tools).

Fine grained representation. The data model must allow describing arbitrary fragments of media items. The scope of a description may vary from whole media items to small spatial, temporal or spatiotemporal fragments of the media item.

Structured representation. The metadata model must be able to hierarchically structure descriptions with different scopes and descriptions assigned to fragments of different granularity.

Modularity. The metadata model should avoid interdependencies within the description, such as between content analysis results from different modalities (e.g. audio and visual). The metadata model shall also separate descriptions which are on different levels of abstraction (e.g. low-level feature descriptions and semantic descriptions). This is important, as descriptions on higher abstraction levels are usually based on multiple modalities and often use domain specific prior knowledge.

Extensibility. It must be possible to easily extend the metadata model to support types of descriptions not foreseen at design time or which are domain or application specific.

Interoperability. It shall be easily possible to import metadata descriptions from other systems or to export to other systems. This can be realised by basing the metadata model on an existing standard.

2.2.2. Access components

Access components are software tools that enable the components of the systems to use the metadata model. They shall abstract the technical details of the metadata description (e.g. storage format, database scheme) and provide an object-oriented access to the metadata model. The key properties of access components are described below.

Fine grained access. Because a comprehensive metadata description can become considerably large, while certain components just work on smaller fragments, the access components must allow accessing not only whole descriptions but also description fragments.

Independence of Storage Technology. The access components shall abstract the storage technology used towards the processing components and be independent of technology specific restrictions.

Distributed Architecture. As most media processing systems are distributed, while there is a central access point to the metadata descriptions, the access components must be capable to work in a distributed environment, with all consequences arising from possible concurrent access.

2.3. Storage

2.3.1. Essence

In a distributed media processing system, it may be useful to centralise essence storage, if many components need access to the same essence. Due to the size of essence data, the network may turn out to be the bottleneck in this case. Distributed storage can be more advantageous in some systems, although it may result in delays, when essence has to be transferred between clients.

In some media monitoring applications, which analyse a live media stream, it may not at all be necessary to store the original essence. If analysis is performed in real-time, it is often sufficient to store only relevant parts or a transcoded low bit rate version permanently.

2.3.2. Meta-essence

We define meta-essence as data that is derived from the original essence like metadata, but has essence characteristics. This includes for example low-resolution previews, and representational items such as key frames. In terms of storage, meta-essence can be treated similar to essence.

2.3.3. Metadata

Metadata can be stored attached to the essence, for example in a container format like MXF [3]. This is mainly useful for exchanging essence and the associated metadata, but not inside a processing system.

Inside a system, metadata should be stored centrally or at least indexed centrally. All components of the system will need access to the metadata repository and it is crucial for performance to have efficient access structures to metadata. A database system will be used in systems that need search capabilities. Many metadata standards are based on XML, and XML support is not on a sufficient level in many database systems for handling complex XML documents (cf. [4]).

3. General Architecture of Media Processing Systems

Based on the considerations stated in Section 2, we will identify the components, which are shared by the media processing systems that are in the scope of this work. The main components are the workflow controller, a set of processing modules, the metadata server and the essence/meta-data storage (file based). Metadata can be stored in files or in a database. The components are connected by workflow, metadata and meta-essence interfaces. Together they

form the generic architecture shown in Figure 5. The components and the interfaces between them are discussed below in section 3.1. Some selected common infrastructure components are described in more detail at the end of this chapter in section 3.2.

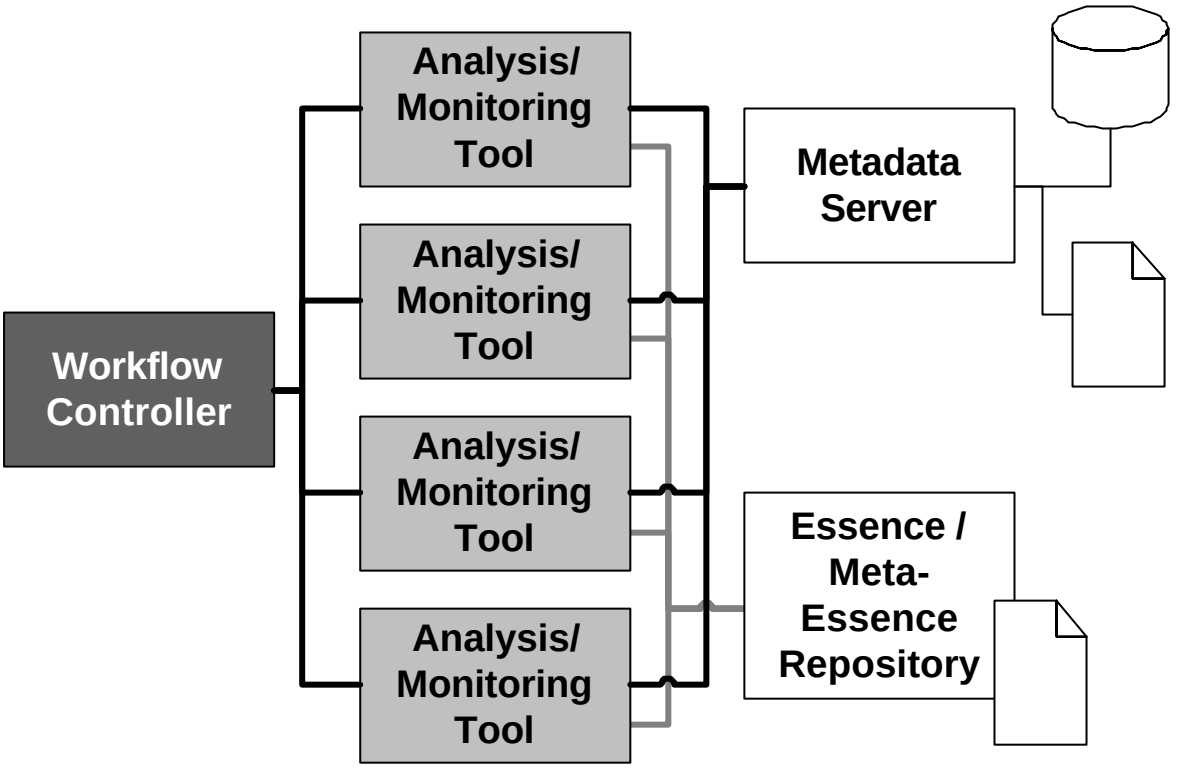

Figure 5: Generic architecture of an audiovisual content-analysis / media monitoring system.

The architecture of a media processing system is very often distributed. This has two main reasons: Firstly, systems need to be scalable and secondly, systems are often heterogeneous, e.g. different components require different platforms. The consequences are the need for shared access to metadata and essence/meta-essence.

3.1. Components and Interfaces

3.1.1. Components

Workflow controller. This is the component that manages the whole processing system either based on a pre-defined configuration (as is the case in a media monitoring system) or based on user interaction. It distributes tasks to processing components and monitors the status of running tasks.

Set of processing tools. A set (or chain, if the pipeline approach is used for analysis tools) of components that perform the actual analysis and/or manipulation tasks. They receive tasks from the workflow controller, access the

storage to read/modify essence and use the metadata server to read or update metadata descriptions of the essence being processed.

Metadata server. It is a central access point to metadata, which serves as an interface to the metadata storage and abstracts it towards the other components. It provides functionality to get and update metadata descriptions and fragments thereof.

Essence/meta-essence storage. This is the storage for essence and metadata files, which can be accessed using some standard protocol (local file system, SMB, NFS, FTP, etc.).

3.1.2. Interfaces between the Components

The components described above are connected by three different types of interfaces:

Workflow. Between the workflow controller and the analysis and manipulation components, processing instructions as well as status and progress reports are exchanged.

Essence/meta-essence. The processing components read and modify essence and/or meta-essence.

Metadata. The processing components access and modify metadata descriptions using the metadata server.

3.2. Selected Infrastructure Components

We have implemented a set of generic and versatile components for common infrastructure tasks in a media processing system. Some of them are discussed in the following.

3.2.1. Content-Analysis Framework

The content-analysis framework consists of a framework controller and a plug-in architecture for the audiovisual content analysis modules. An analysis graph is formed from the modules and models data flow and dependencies of the modules. The framework supports multi-threading, so that parallelisation of processing is possible. Content-analysis modules to be executed within the framework can be easily written by fulfilling a simple interface.

There are generic modules for access to video and audio data, so that access to the essence is done only once for the graph. All modules that need access to essence can connect to the input module.

3.2.2. MPEG-7 Library

We have implemented an API for parts 3, 4 and 5 (visual, audio, MDS) of MPEG-7 [1], which is used by the components of our systems to work with the metadata model. The MPEG-7 library provides an object-oriented and typed representation of the entities in the MPEG-7 standard is more efficient and has some other advantages over using a generic concept like DOM, such as type safety, and the possibility to add type specific implementation (e.g. specific access structures for collections of elements). Because of the amount of types in the standard we have implemented a code generator that produces C++ classes from the MPEG-7 XSD files. This also makes the library future proof, as new code can be simply generated whenever new versions of the standard are released. The library is freely available [2].

The library supports XML as serialised representation of MPEG-7 descriptions, which has several advantages over a binary representation, such as human readability, the number of available tools and the extensibility by way of schema extensions.

3.2.3. Client/Server Infrastructure for Access to MPEG-7 Documents

Because of the distributed nature of media processing systems, we have designed a client/server infrastructure for access to metadata documents, with the focus of functionality on the server side. The document server provides read/write access to MPEG-7 documents for a number of clients. It allows the exchange of whole documents or any fragments thereof, which are addressed by XPath statements. As MPEG-7 XML documents of larger media items tend to have considerable size, direct access to document fragments is crucial for efficiency. The document server provides mechanisms for concurrent access to documents by multiple clients.

The document server also abstracts the infrastructure used to persistently store the document (e.g. file or database), which is an advantage over direct access to a database, as partial access and especially update on a fine grained level is not supported by many XML enabled databases (cf. [4]).

4. Application Examples

4.1. DIRECT-INFO

The goal of the DIRECT-INFO project is to create a system for semi-automatic sponsorship tracking in the area of media monitoring. It will offer an integrated system combining the output of different media analysis modules to

semantically meaningful trend analysis results, which give executive managers and policy makers a solid basis for strategic decisions.

The DIRECT-INFO system architecture is based on the following basic design criteria: to take a proven, feasible, affordable and cost effective approach, to build on reliable technology, to define an open architecture in terms of extensibility, modularity and exchangeability of components and to be scalable of computing power and data throughput. Other important requirements that are more technical are: independency of the main system components, ease of integration and system robustness.

The DIRECT-INFO system is able to do 24/7 monitoring. Since not all analysis components of the DIRECT-INFO work in real-time there is a need to pre-filter the incoming media stream(s) in order to detect so called "relevant semantic blocks", which refer to programs that are relevant for analysis. The Content Analysis Controller is responsible for the analysis workflow and decides on the relevancy of these blocks by taking into account EPG information and results from a heuristic genre classification module.

This approach is based on the assumption, that on average all the subsystems can perform the analysis steps in between the time the next semantic block is detected. Considering the use case "sponsorship tracking" this assumption is a realistic one and was agreed with the operators of the system. Further up-scaling can easily be achieved by replicating the number of subsystems.

For realisation of such a workflow the system architecture as visualized in Figure 6 was defined.

Acquisition and Media Repository provide the Content Analysis Controller (CAC) with the media to be analysed. The CAC triggers the analysis subsystems according to the analysis workflow. The subsystems store their results in an MPEG 7 document which is evaluated by the Fusion component to detect relevant appearances of a sponsor's brand in related the sponsored sports team.

In order to easily enable further integration of new analysis subsystems all communication between components is based on well defined interfaces implemented via web service technology. Metadata are centrally stored in the standardized MPEG-7 format within the MPEG-7 Document Server. For the storage of essence data there is a central media repository and content is transferred to the analysis subsystems on demand.

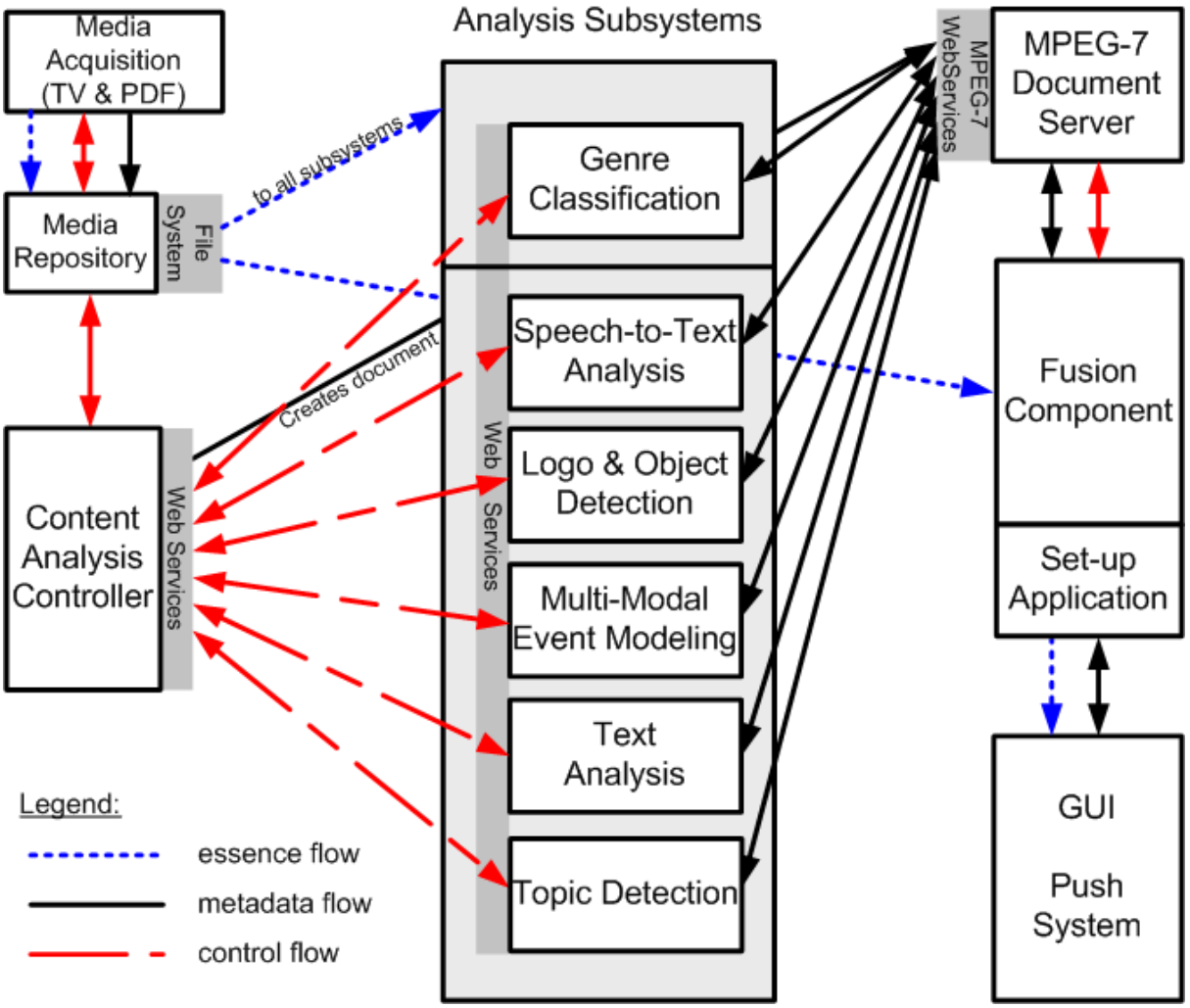

Figure 6: System architecture of DIRECT-INFO system.

There are three different kinds of flows within the DIRECT-INFO system. The essence flow describes where the content (i.e. video, audio, images, text) has to be distributed within the system. The metadata flow describes the distribution of all descriptive data belonging to the content, e.g. acquisition time of MPEG-2 essence, or the results generated by the logo detection module. The control flow denotes which components of the system control (start, stop, etc.) other components.

After a TV programme is captured the CAC triggers the creation of the MPEG-7 document and the analysis subsystems in a pre-defined order. The subsystems perform their analysis tasks individually and independently and store their results in one MPEG-7 document per semantic block. The Fusion component parses this document to find out which appearances of a sponsoring brand are related to the sponsored company.

4.2. PrestoSpace

The objective of PrestoSpace is to develop an integrated approach to the preservation of and access to audiovisual archives, the so-called “Presto-Space Factory”. It covers the complete workflow from digitisation, preservation, restoration and storage to access and delivery. Figure 7 shows the architecture of the PrestoSpace content-analysis and restoration system.

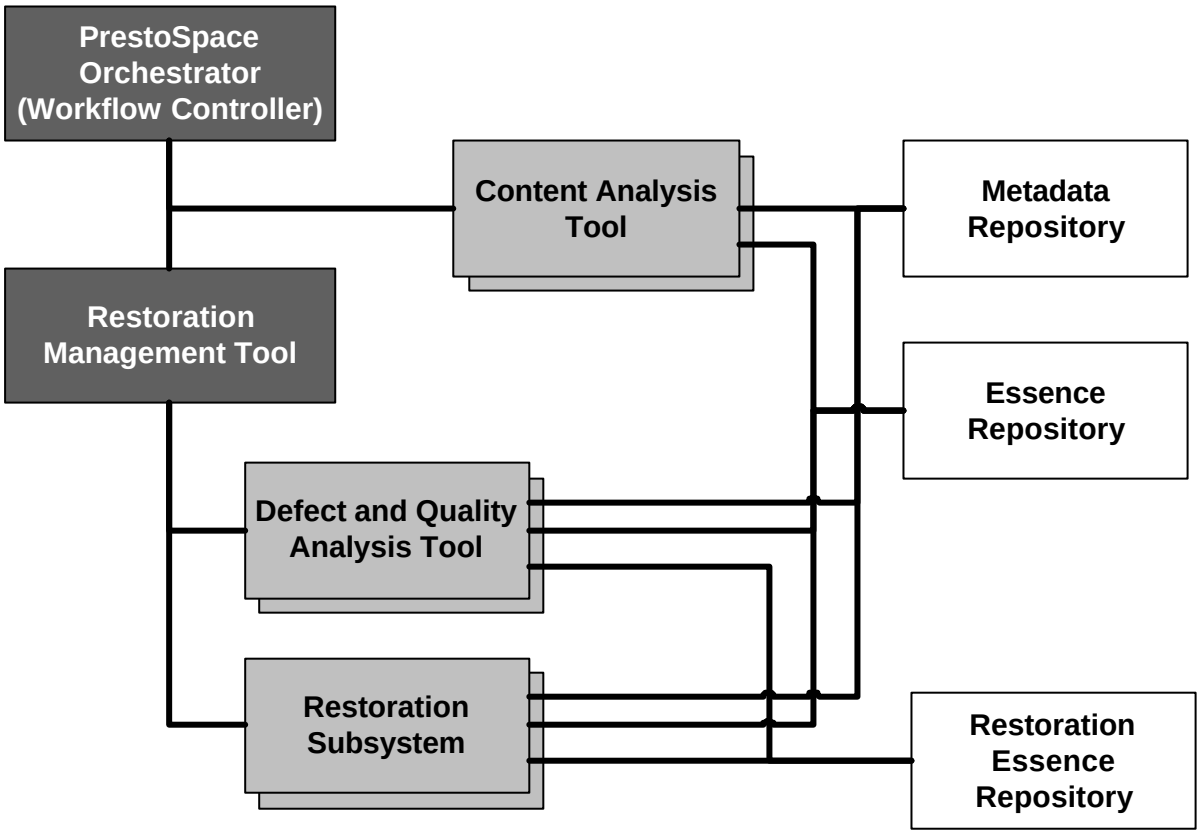

Figure 7: Architecture of PrestoSpace content-analysis and restoration system.

The PrestoSpace Orchestrator is the central workflow controller for audiovisual content analysis and restoration.

Automated defect and quality analysis is performed on the material to be preserved. A set of restoration subsystems is available for performing audio and visual restoration tasks. The defect and quality analysis workflow and the assignment of restoration tasks to the appropriate restoration subsystems are handled by the Restoration Management Tool.

In order to increase the accessibility of the archived material, additional metadata is annotated. This is done by applying a number of automatic content-analysis tools as well as by manual documentation.

5. Conclusions

A generic architecture has been developed and was implemented for applications in three different areas of audiovisual media processing: digitization/restoration, interactive TV and media monitoring. The concept of creating a modular toolbox of independent components with well-defined interfaces allows providing integrated customized solutions for different use cases.

The architecture builds on accepted standards:

- MPEG-7 is used for metadata storage and exchange. This XML-based standard also enables the future use of ontologies for annotation of audiovisual content
- Web service technologies for communication (control and metadata flow) between components.

This allows easy integration and re-use of components in multiple applications.

6. Acknowledgements

The ideas and results presented in this paper have been developed at the Institute of Information Systems & Information Management of JOANNEUM RESEARCH during the last few years in various projects performed and supported on national and European level. All this support is gratefully acknowledged.

Particularly, the above mentioned projects DIRECT-INFO (IST-FP6-506898) and PrestoSpace (IST-FP6-507366) have been partly funded by the European Commission in the Sixth Framework Programme.

7. References

[1] ISO/IEC 15938:2001, Multimedia Content Description Interface.

[2] JOANNEUM RESEARCH MPEG-7 Library. URL: http://iis.joanneum.at/mpeg-7.

[3] SMPTE 377M-2004, Material Exchange Format (MXF) File Format Specification.

[4] Westermann U., Klas W. 2003: An Analysis of XML Database Solutions for the Management of MPEG-7 Media Descriptions. ACM Computing Surveys 35(4):331-373.