Managing Metadata

Matthias Schnöll, Roman Holzhause and Tristan Riedel
Anhalt University of Applied Sciences – Department of Electrical, Mechanical and Industrial Engineering
Köthen, Germany

Abstract – *Since the introduction of file-based workflows in broadcast and video production, a variety of formats and file standards has been emerged to provide exchange, research and archiving media assets in multiple system environments. During the working process, metadata is added to the assets. Metadata, which includes technical and descriptive information, enriches the asset because it becomes more accessible. But handling specific metadata across different formats and systems is a challenge due to the lack of compatibility and interoperability, which can lead to the loss of information. In relation to this issue, a prototype was developed to provide solutions for managing and mapping interoperable metadata. The prototype offers several web services to provide a better efficiency of working with metadata within media production, distribution and archiving. As a result, an insight look at the main feature components of the tool is given. With the underlying system architecture new ways of interoperability and advantages in carrying and transformation of metadata can be achieved.*

The Relevance of Metadata in Broadcasting

As the importance of tapeless workflows has increased in the production area of broadcasting, multiple formats and standards were introduced that shall enable the exchange, playback and archiving in various media system environments. Apart from the audio-visual content, which belongs to the data essence, metadata and especially descriptive information is generated within the value chain. There is a multitude of systems on the market that perform their file-based tasks innovatively and without an error. However, problems can arise when these individual systems have to work together. If several processes are summarised in a workflow, the systems usually lack of compatibility between exchange formats or communication interfaces. In relation to these problems, various exchange formats have been designed in the past, e.g., the Material Exchange Format (MXF) [1]. With this approach, the required metadata can be provided within a container. Because of the huge number of different devices in today´s media landscape, content is formatted several times to deliver it in a specific format. Changing metadata can involve complex processes, in which the information has to be unpacked from the container, changed and wrapped again.

To avoid this, a parallel carrying of metadata in form of a XML file [2] has established as a workaround within file-based workflows. Efforts within the public broadcasting services in Germany, Austria and Switzerland led to design and develop the *Medienbegleitkarte*[1], which will fully replace the presumably used video tape recorder (VTR) record card. Its usage in file-based productions will be introduced in 2015 to ensure a regulated handling of production-related metadata [3]. The Medienbegleitkarte is intended to be used primarily in the delivery of internal and external program contributions and can be carried separately from an MXF file. Depending on the requirements, an individual instance of the Medienbegleitkarte can be composed from different metadata sets and terms. The technical foundation of these sets is the Broadcast Metadata Exchange Format (BMF) [4], which was developed by the *Institut für Rundfunktechnik (IRT)*[2].

New Requirements and Approaches

Through the increasing usage of tri-medial productions, the coherent generation and presentation of print, audio and video content, arises new demands in broadcast environments. Regarding this development, metadata has gained an even higher priority. Metadata is no longer used exclusively for internal data exchange or pure documentation. Metadata can provide a meaningful basis for external services. The delivery and addition of content over the World Wide Web, e.g., HbbTV [5], second screen applications and online media library clarify the importance of metadata. This results in an increased distribution effort, which requires new solutions.

It is important to keep the outlay of generating and managing metadata as low as possible. In relation to targeted storage and generation of metadata, a large amount of information probably follows. The processing of this resulting amount of information is expensive and time consuming. For this reason, there are already software solutions available in the market that can extract the information from audio-visual content systematically and fully automatically. The software 'Semantic Video Annotation Suite' by Joanneum Research is an example for the automatic extraction of metadata [6]. Such analytical tools have the possibility to recognise movie cuts, fades and camera movements. The results can be represented in MPEG-7 [7]. The acquired metadata can then be passed on to subsequent systems. Finally, existing IT-based infrastructures of audio-visual media production can benefit from the introduction of a technology of metadata distribution.

---
[1] German term for media sidecar file
[2] Broadcast Technology Institute; primary research institute for public-broadcasting organisations in Germany, Austria and Switzerland

The lack of coherent, interoperable relationships of metadata complicates the work in a broadcast environment and limits the field of application. Interoperable metadata is primary designed to be used by machines to share functional units and to provide data exchange [8].

In many practical cases, metadata sets from various hardware and software systems have to be merged together. Importing inconsistent datasets into existing production environments require interfaces for proper integration and lossless mapping between different data models. Such interoperations can already be found in workflow management systems, which are able to make decisions according to the content of exchanged messages. For instance, the quality report of a video analysis can retrigger transcoding processes in case of quality deficiencies. Such interoperable solutions are often provided by specific Application Programming Interfaces (APIs). However, the amount of functionality of such interfaces can dither between products, or only be partially opened for developers. Media industry's efforts of establishing a more consistent approach resulted in the implementation of Framework for Interoperability of Media Services (FIMS), which delivers a system architecture for sharing commonly used media processes, such as ingest and transcode [9].

An alternative way of administration could be achieved by directly utilising the existing metadata output of the respective workstation. The management of metadata shares the same previously mentioned objectives as of managing essences or processes. But handling every single metadata output can be challenging, due to the fact of its occurrence, which can be at any point and time. Speaking of media processing, each ingest, transcode or file transfer generates new metadata within the workflow, which results in large amounts of additional data.

Challenges of Format Diversity

Metadata can be found in a vast amount of different formats, which complicates data exchange. To ensure trouble-free exchange of metadata, standardised data structures are needed. Most metadata is marked in the XML format, which can be natively transformed by the programming language XSLT [10].

Due to the extensible multipurpose architecture of the XML schema, countless possibilities of implementing data structures and terms are given. Relying on standards can be a good decision, because of the determined set of parameters and terms. This can maintain interoperability in typical use cases by the cost of extensibility and customisation. But as soon as the standardised structure is limiting the application, complicated workarounds are needed. In unpredictable scenarios, this can lead to incomplete data transformations between individual systems. In the worst case, information gets lost, which aggravates subsequent interpretation of the records.

Application for Managing Interoperable Metadata

The metadata research project, which has been developed in cooperation with *Anhalt University of Applied Sciences* [11] and *Dimetis GmbH*[3] [12] since May 2013, deals with the conception and development of an application for managing interoperable and semantic metadata. One of the goals of this network-based tool is to increase the efficiency of processes in media production and media archiving. Furthermore, the tool provides interoperable distribution options and multi-layered administration of metadata. The application shall allow an automated exchange of generated metadata and is designed for the use within the production, distribution and the archiving of media content.

Functionality and Architecture

The aim of the application is to automate metadata enrichment in file-based workflows and to support the observance and usage of a single metadata format within a company. The prototype is designed to interact with other media tools as a key component on the network. Therefore, the prototype is a client-server application, which is written in Java. An illustration of the systems architecture is shown in Fig 1. Functions for distributing, mapping and transforming metadata are implemented and also provided as SOAP web services [13] for external software components. Additionally, it is possible to control all operational processes of the application by the use of an external system, e.g., the *Dimetis Boss File Transfer Manager*. Additionally, the application includes a graphical user interface that can be accessed through a web browser, which acts as the client. The representation within the browser works with HTML5, CSS3 and JavaScript [14]-[16]. Thus, no additional plugins for commissioning are required.

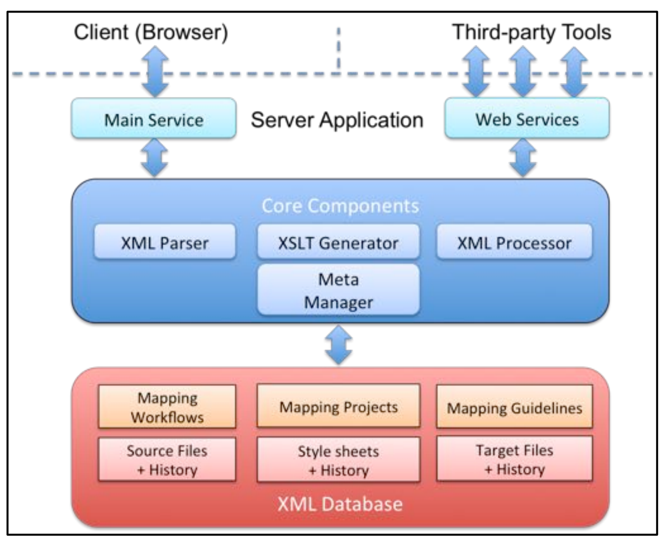

Fig 1 Architecture of the metadata management system

[3] Broadcast company located in Germany; provides software solutions for network management systems

The *Main Service* handles the communication between client and server and combines the underlying technical components for metadata processing. It is used to receive user input and external XML documents from the client. The system includes an XML parser for the modelling of metadata, an XML processor for validation and transformation and an XSLT generator for automated generation of XSL style sheets. Furthermore, the prototype implements an XML database for archiving, which is based on the *BaseX* database system [17].

In this compilation, native operations with XML can be implemented directly within the database. Inside a background process, queries are formulated by using the XML Query Language (XQuery) [18], so that an extensive manipulation and output of metadata is possible.

Within the database, several documents are stored for metadata management and held together in relations. Each supplied document is assigned to a unique identification number (UID), which represents a fixed relationship with certain essences or devices. As a result, a network of metadata evolves as a foundation for flexible merging and distribution between different formats. In addition, the working history of all metadata is documented by the system inside a versioning process. Based on this data collection, every manipulation of metadata can be tracked within a workflow. This applies to source and target documents as well as for the style sheets used for transformation. Furthermore, every project relevant data, which describes the respective metadata mapping and configuration between source and target documents, are kept in the database too.

Graphical User Interface

The integrated graphical user interface (GUI) is designed to prepare and configure metadata mappings. A screenshot of the GUI is shown in Fig 2.

A mapping describes the transport of data between a source and a target. XML documents, which are needed for a mapping, can be uploaded directly from the browser.

The mapping is implemented by a drawing surface between two modelled XML documents. Each metadata entry has its visual indicator for mapping operations, which is presented as a circle right next to it. The circle serves as an anchor point for the mouse cursor for comfortable interactions. With the help of the cursor, lines can be drawn between the entries of the source document on the left side and those of the target document on the right side. A drawn line represents the basis for a mapping, which can be coupled with additional conditions and parameters insight the *Mapping Editor* in a pop up window.

In most cases, multiple transformations of nodes and data types are necessary as well as the merging and splitting of the entries. Depending on the node type, more options could be available. In case of an element node inheritances to child entities can be chosen.

As further assistance for working with large documents a toolbar is available. For a specific mapping purpose, tree structures can be filtered by node type. Likewise, a search option offers results in a dropdown menu while typing. By clicking on a search result or mapping line, the navigation pushes the related node into the users view.

Moreover, the user interface provides options for viewing and editing source, target and transformation documents as well as the generated style sheet. To keep track of all documents, a built-in code editor with multiple tabs can be toggled to the side of the main view.

Metadata Modelling

To transfer content from one XML document to another, it has to be technically read and parsed, which is done by accessing the entities, the smallest information unit of an XML document. There are two common parser systems: Document Object Model (DOM) [19] and Simple API for XML (SAX) [20].

Both are used for different purposes within the metadata management system and work as background processes, which handle incoming metadata at any time. A DOM parser works by the principles of modelling a DOM tree (Fig 3).

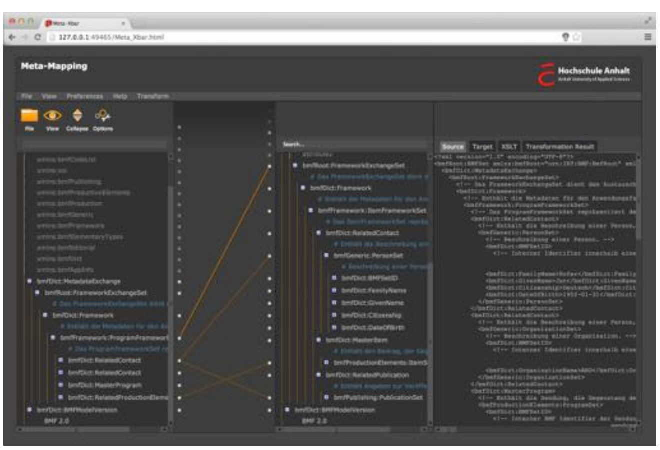

Fig 2 Graphical user interface

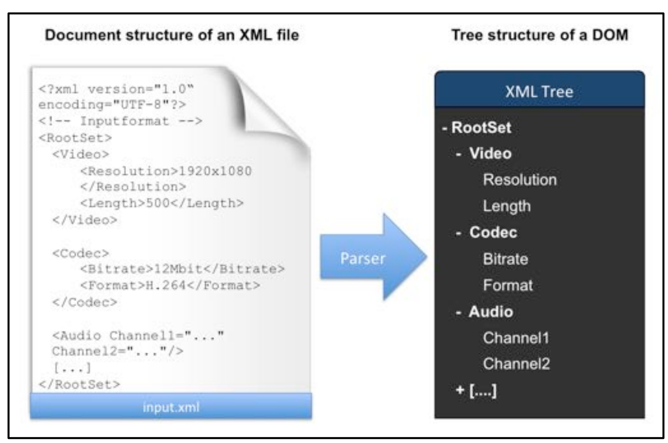

Fig 3 Modelling of an XML document

The DOM tree reflects the XML document by holding its entities as nodes in organised hierarchies and coherencies. Each node can be selected and manipulated, as well as the whole document structure. Furthermore, relations between entities can be found by analysing the structure of the model, which is useful for generating mapping instructions. Additionally, the DOM tree is used for displaying the metadata document in the graphical user interface.

In the case of processing only specific entities, a SAX parser is used. The SAX parser works event-based. While sequentially reading the document, it calls an event when the processed entity matches a given parameter. This approach takes less time than building a model of the whole document and is useful for filtering and searching for entities - one of the main operations in the prototypes database system.

Transformation and Translation of the Mapping Instructions

The setup of the mapping has to be done by the user through the integrated graphical user interface as described before. Each drawn mapping line contains additional metadata, which describes the mapping operations for the transformation process. Therefore, the mapping operations will be translated into XSLT 1.0 syntax by the XSLT generator and stored in a separate style sheet (Fig 4).

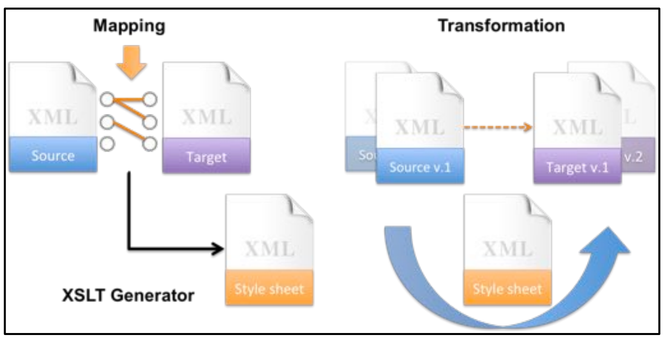

Fig 4 Translation (left) and transformation (right) of the mapping

A style sheet has the advantage to be highly reusable and portable in similar mapping situations, which resemble the same data structure and format. Additionally, the XSLT syntax enables enhanced cooperation with external third party tools. By defining a selection of entities of the source as references and a body, which represents the structure of the target document, the data flow can be controlled through the style sheet.

After dynamically generating a style sheet, it will be used in the transformation process together with the source document to finally deploy the mapping. This will be done by the XML Processor, which is based on the *Apache Xalan Processor* [21]. While processing the style sheet a new XML instance will be build that resembles the original target document with updated entities from the source. At the same time, the documents will be checked on validity and well-formedness. Finally, the resulting documents will be stored in the database and are ready for distribution to enquiring third party tools on the network.

Further Development

The metadata management system is a flexible tool for maintaining and mapping several metadata sources and systems. The prototype provides solutions for a better interoperability across workflows with high amounts of different formats, metadata sets and terms. In the current state of the prototype, mappings can be done across XML-based formats. The support of text and CSV files, as well as KLV-coded data will be implemented in further states of development.

The underlying XSLT Generator is built up from fragments of the XSLT language. The prototype does not support the whole XSLT vocabulary but can be easily expanded by operations to make the underlying mapping processes more efficient in future. All metadata documents will be stored inside an XML database, as well as related tracking information, which describes the documents state and binding in the ecosystem. The possibility to visualise the resulting metadata network could provide intuitive ways of interaction and managing features.

The prototype is capable of mapping multiple source and target documents but the graphical user interface is yet limited to display only one source and target at a time. At present, the processing of multiple files will be handled by sequencing the mapping projects via a batch processing approach. Implementing semiautomatic functions can increase the quality and speed of the mapping process. Therefore, a structure analyser could provide mapping suggestions in form of context sensitive parameters and operations.

The automation can be achieved with three different types of mapping suggestions: logical, application-specific and user-generated mapping rules. Logical mapping operations describe bindings by observing structural information of the document. The data structure of an XML instance can be fully described via schema documents, which are mainly used to validate XML instances. A schema defines all possible types and occurrences of entities inside a specific construct. The analyser searches for corresponding metadata entries by comparing the properties of each term of source and target schemes.

Additionally, application-specific mapping rules can inherit technical behaviour of particular relations in metadata standards, which are necessary to map through different standards with no restrictions. For example, BMF avoids redundant data by creating references between entities with the help of the Unique Identifier (UID). A specific mapping rule could imitate this behaviour by swapping already existing content with references according to the desired format. To operate correctly, the rule must be known by the parser as well as by the XSLT generator.

Another addition to the data analysis is the implementation of user-generated rules. The user can modify or accept the mapping suggestions. With the help of the user input, a learnable system can involve the gathered information for advanced semantic mappings in upcoming projects.

Conclusion

Metadata and its related files have a great impact on the multimedia workplace, and its involvement sets new requirements for the broadcast industry.

There are currently several approaches given from companies and working groups on how to deal with metadata in general and how to unify its information. In most cases, the usage of an uniform format will be recommended. Due to the complexity and limits of those formats, its implementation can be challenging and lead to insufficient results. The lack of tools with compliant binding and conjunction features is one reason for that. The aim of the metadata management system is to simplify the workflow with metadata and automate related tasks to achieve interoperability across file based environments. The functionality of the described prototype will be implemented in further software products of the Dimetis GmbH.

References

[1] Devlin, B., Wilkinson, J., Beard, M., Tudor, P., Wells N. (Editor); "The MXF Book: An introduction to the Material eXchange Format", *book,* Edition: 1st, ISBN 13: 9780240806938, 2006-03-28, 402.

[2] W3C; XML Core Working Group; Extensible Markup Language (XML); http://www.w3.org/XML/; *webpage*; last modified date:2014-10-03

[3] Institut für Rundfunktechnik GmbH, München: Technische Richtlinien-HDTV zur Herstellung von Fernsehproduktionen für ARD, ZDF und ORF; *program guideline*; May 2014

[4] Institut für Rundfunktechnik GmbH, München: Broadcast Metadata Exchange Format; http://www.irt.de/de/themengebiete/produktion/bmf.html; *webpage*; last modified: 2014

[5] Institut für Rundfunktechnik GmbH, München: Hybrid broadcast broadband TV; https://www.hbbtv.org/; *webpage;* last modified 2014

[6] Joanneum Research Forschungsgesellschaft mbH, Graz; www.joanneum.at, Semantic Video Annotation Suite; http://www.joanneum.at/digital/produkteloesungen/semantic-video-annotation.html; *webpage*

[7] Martinez, J., M.; "MPEG-7 Overview (version 10)"; *paper;* ISO/IEC JTC1/SC29/WG11, N6828; Palma de Mallorca; October 2004

[8] Rühle, S.(SUB Göttingen); Kompetenzzentrum für Interoperable Medien (KIM); Richtlinie für die interoperable Gestaltung von Metadatenprofilen; *guideline*; March 2012

[9] AMWA (Advanced Media Workflow Association); http://wiki.amwa.tv/ebu/index.php/Main_Page; *webpage*; last modified Date: 2014-07-07

[10] Tidwell, D.; XSLT; O'Reilly Media, Inc; USA; ISBN: 978-0-596-52721-1; Second Edition; *book*, June 2008

[11] Anhalt University for Applied Sciences; http://www.emw.hs-anhalt.de/www/; *webpage*

[12] Dimetis GmbH; http://www.dimetis.de/; *webpage*

[13] W3C; Gudgin, M., Hadley, M. Mendelssohn, N., Moreau, J.-J., Nielsen, F., H., et al.; SOAP Version 1.2 Part 1: Messaging Framework (Second Edition); http://www.w3.org/TR/2007/REC-soap12-part1-20070427/, *webpage, recommendation,* 2007-04-27

[14] W3C; Hickson, I., Berjon, R., Faulkner, S., Leihead T., Doyle Navara, E., et al.; HTML 5 A vocabulary and associated APIs for HTML and XHTML; http://www.w3.org/TR/html5/; *webpage, recommendation;* 2014-10-28

[15] W3C; Eternad, J., E.; Cascading Style Sheets (CSS) Snapshot 2010; http://www.w3.org/TR/css-2010/; *webpage;* 2011-05-12

[16] W3C schools; JavaScript Tutorial; http://www.w3schools.com/js/; *webpage*

[17] BaseX. The XML Database; http://basex.org/; *webpage;* last modified: 2014

[18] W3C; Boag, S., Chamberlin, D., Fernández, F., M., Florescu, D., Robie, J., et al.; XQuery 1.0: An XML Query Language (Second Edition); http://www.w3.org/TR/xquery/; *webpage, recommendation;* 2010-12-14

[19] W3C; Leithead, T.; DOM Parsing an Serialization; http://www.w3.org/TR/2014/CR-DOM-Parsing-20140617/; *webpage, candidate recommendation;* 2014-06-17

[20] About SAX (Simple API for XML); Megginson, D.; http://www.saxproject.org/; 2004-04-27; *webpage*

[21] The Apache XML project; Xalan-Java Version 2.7.1; http://xml.apache.org/xalan-j/index.html; *webpage*